Learning and Planning in Structured Latent Spaces for Legged Robot Locomotion



Alexander Luis Mitchell

Supervised by Prof. Ingmar Posner
and Prof. Ioannis Havoutis

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Hilary 2024

Acknowledgements

I would like to thank my supervisors, Professors Ingmar Posner and Ioannis Havoutis. You have both been excellent mentors throughout the last four years. I have thoroughly enjoyed being part of both the Applied AI lab and the Dynamic Robot Systems group.

I am very grateful to all my collaborators. I have learnt so much from you all and I am very excited to see what we will work on in the future. I would like to thank in no particular order Wolfgang Merkt, Martin Engelcke, Oiwi Parker Jones, Siddhant Ganapurwala, Mathieu Giesert, Jun Yamada, Aristotelis Papatheodorou, Jack Collins and Daniele De Martini. All of you helped me grow and develop as a researcher and you deserve a paragraph of your own.

I am grateful for my thesis examiners Professor Patrick van der Smagt and Doctor Jack Umenberger. I want to thank you for your time in reviewing this thesis and your useful suggestions.

I send a special thanks to all my friends and family who have kept me sane. Above all, I would like to thank my parents Anthony and Marikin for their loving support throughout the stressful and joyful moments this process has delivered. I would also like to thank the EPSRC Centre for Doctoral Training in Autonomous Intelligent Machines and Systems for their support and Wendy Poole for guiding our cohort through the D. Phil. process.

Abstract

Quadruped robots are capable of operating in human-oriented environments littered with stairs, no-step areas and cabling, whilst carrying multiple sensor payloads. Therefore, these robots are popular in the industrial, maintenance and energy sectors. However, the versatility of these robots comes with the cost of complex nonlinear dynamics. These dynamics have both continuous and discrete elements, for example, joint-space motion such as joint angles and torques vary smoothly, but contact states are discontinuous and non-differentiable. Optimising trajectories constrained by these dynamics can be complex and slow resulting in narrow basins of convergence and limited windows of operation. Indeed, these methods often require precomputed footstep locations and schedules to be real-time capable meaning that these gait characteristics cannot be varied on demand. Model-free methods tend to be black-box and it is not possible to vary the resulting policies or inspect the properties of the resulting gaits without some sort of prior inductive bias. An example of such an inductive bias is the relative phase of each leg normally referred to as the gait phase. Locomotion is periodic since each leg continuously makes and breaks contact and the gait phase describes the contact and swing relationships between the feet.

An ideal planner would have the following properties: capable of a diverse range of motions or gaits, and integrate proprioceptive and perceptive information such as vision for terrain traversal all while being robust and interpretable. Indeed in nature, we see a diverse range of robust skills such as continuous gait switching and modulation to external stimuli for example speed demands or changes in terrains. This thesis investigates whether such a controller can be created by learning a unified representation for quadruped locomotion across skills. We train a generative model namely a variational autoencoder (VAE) on examples of locomotion trajectories. In doing so, we discover that the robot's gait characteristics (footswing length and height) become disentangled into separate dimensions of the latent space. As a result, these parameters can be varied independently by altering the amplitudes of two oscillatory signals in latent space. Furthermore, training the model with examples of distinct gait types (trot, crawl and pace) reveals that the latent space is able to infer meaningful correlations across the gaits. As a result, the VAE is able to infer novel and unseen gait transitions between the motions. With the addition of the rudimentary terrain encoding, the latent space is able to adjust its structure to adapt to changes in terrain. A learnt planner in the latent space adjusts the gait characteristics

in response to the terrain and improves the robustness of foothold location during the climbing phase. At this point, we show that a latent space forms a suitable planner capable of delivering a diverse range of locomotion skills all while being robust and interpretable. Since our methods are inspired by observations of nature, we posit if there exists similar representations in the natural world. In prior work by Churchland et al, it was discovered that both locomotion and reaching skills are encoded by the circular firing of neurons in the motor cortex of primates. These circular neural dynamics are strikingly similar to those in the locomotion latent space so far as the time periods and phase lag between the cycles in the motor cortex and latent space are of the same magnitude. Inspired by the similarities between the locomotion and manipulation oscillations in the motor cortex, we train a manipulation-specific latent space and discover that the dimensions of this space are also meaningfully disentangled. Longitudinal and lateral movements of the robot's end effector map to separate latent-space dimensions. Therefore, rotational dynamics similar to those in the motor cortex can be used to solve reaching tasks deployed on the real robot manipulator.

This thesis explores the opportunities that a unified latent representation of locomotion can provide and shows how it is used to create a versatile planner. In doing so, a diverse range of motions with continuously adjustable gait characteristics is achievable. Furthermore, a latent space trained with examples of multiple distinct gaits is able to infer the gait phase across gaits. Exploration of this space reveals continuous gait transitions via novel contact schedules. The addition of a rudimentary terrain encoding and a learnt planner in latent space permits the robust traversal of uneven terrains. The planner is deployed on the real robot and all experiments are analysed using data from runs using the real platform.

Contents

1 Introduction **1**

1.1 Outline And Contributions 5

1.1.1 First Steps: latent space Control with Semantic Constraints for Quadruped Locomotion 6

1.1.2 VAE-Loco: Versatile Quadruped Locomotion by Learning a Disentangled Gait Representation 7

1.1.3 Gaitor: Learning a Unified Representation Across Gaits for Real-World Quadruped Locomotion 8

1.1.4 From Primates to Robots: Brain-like Latent Dynamics Emerge in Robot Systems during Walking and Reaching 10

1.2 Publication List 11

2 Background and Related Work **13**

2.1 Background 13

2.1.1 Rigid-Body Dynamics 14

2.2 Related Work 16

2.2.1 Flat Ground Operation 16

2.2.2 Terrain Traversal 21

2.2.3 Singular Policies For Multiple Skills 25

3 First Steps: Latent-Space Control with Semantic Constraints for Quadruped Locomotion **28**

4 VAE-Loco: Versatile Quadruped Locomotion by Learning a Disentangled Gait Representation **40**

5 Gaitor: Learning a Unified Representation Across Gaits for Real-World Quadruped Locomotion **60**

6 From Primates to Robots: Brain-like Latent Dynamics Emerge in Robot Systems during Walking and Reaching **74**

7 Discussion and Limitations **110**
7.1 Limitations 112
7.2 Future Work 113

8 Conclusions **116**

References **118**

Appendices

A Next Steps: Learning a Disentangled Gait Representation for Versatile Quadruped Locomotion **122**

1 Introduction

Quadruped platforms are a popular choice in both research and industry as they are capable of traversing complex terrains whilst carrying multiple sensor payloads [1]. Particular applications include continuous monitoring tasks in offshore energy facilities, inspection to track progress at construction sites and search & rescue after natural disasters. However, designing controllers for quadrupeds operating in such environments is challenging. Locomotion often requires solving a mixture of programs, which are constrained by both continuous and discrete dynamics. For example, joint position and torque trajectories vary smoothly, whilst the robot's contact dynamics have discontinuous elements, which arise from the continuous making and breaking of contact as the robot takes steps. The combinatorial nature of planning these contact transitions poses challenges to online performance, thus it is either disregarded and given to the solver as apriori contact schedules [2], or incorporated offline [3]. Methods which frame locomotion as an optimisation program and are constrained using the complete joint-space dynamics suffer from lengthy convergence times which can limit their real-time performance. In practice, domain-specific knowledge tends to alleviate the challenges associated with nonlinear locomotion dynamics.

Despite the complexities of quadruped locomotion, there is a broad range of approaches to these problems. A recent example of a method which uses an analytical model of the robot's dynamics is found in [4]. This method uses Differential Dynamic

Programming (DDP) [5] to refine the trajectory to a locally optimal one iteratively. This is achieved by rolling out the dynamics and back-propagating the gradients of the value function in order to find the locally optimal policy. This value function encodes the dynamics implicitly. Using a Model-Predictive Control (MPC) online closed-loop control can be achieved by solving the problem in a receding horizon manner. Recent works have shown that these methods can produce dynamic motions including trot, climbing of stairs and jumping. However, domain knowledge is required to use DDP-based methods for real-time deployment on quadrupeds at $50\,\mathrm{Hz}$. The contact schedule and timings are determined in advance using simplified models [6–8] meaning that once these parameters are set they are locked in and cannot be varied on-the-fly. As mentioned, contact dynamics are complex and nonlinear meaning that they are difficult of model and differentiate. As a result, there is a large body of research, which solves for a locomotion policy using learnt policies and value functions. The first examples of such methods deployed on the ANYmal series of robots include [9] and [10]. These methods are both model-free on-policy methods and use carefully designed reward functions to learn robust locomotion controllers on flat ground for blind operation. To achieve such impressive results, much reward tuning is required and it is not possible to vary the resulting locomotion. For example, parameters such as footswing heights and lengths are not adjustable. Subsequent methods such as [11] and [12] built upon prior art and incorporated the robot's gait phase as an explicit inductive bias. The gait phase governs the motion, timing and contact state of each foot, while the relative phase of all the other feet determines the gait. Using the gait phase as an inductive bias provides a great deal of structure to guide the value and policy learning process for locomotion. As a result, these methods required less tuning and were able to deliver trot locomotion in a broader range of environments including even terrains with and without perception. Despite these impressive capabilities, these methods were unable to produce or operate over multiple skills or gait types meaning that alternative controllers are required to switch to a different type of motion. For example, an operator is required to annotate the locations of staircases in the environment, and a separate controller is used to climb these. In recent years, the locomotion community has turned their attention to designing policies capable of generating multiple locomotion

skills given changes in terrain conditions. An example of this is in [13] which uses a planning policy to choose an individual sub-policy which is designed to generate a specific locomotion skill to traverse parkour-style obstacles. A limitation of this work is each sub-policy is independently trained meaning that smooth transitions between them are not possible. To address this limitation, [14] uses a transformer [15] to learn a single policy which can produce a range of skills. However, this is a black-box method meaning that it is not amenable to inspection and it is not possible to adjust the behaviour of each skill once trained. Irrespective of how locomotion controllers are designed, we would ideally choose a single locomotion planner capable of generating a broad range of robust locomotion skills with tunable gait characteristics. Additionally, the locomotion trajectories can be conditioned with perception to traverse uneven terrain all while being safe with an interpretable structure. Indeed, we see in nature that animals are able to seamlessly switch between different gait types on-demand and in response to external stimuli such as terrain types and increasing speed demands. Please note that this thesis focuses on the gaits used for traversing terrains commonly found in industrial and outdoor environments such as such as trot, crawl and pace. Highly agile motions such as jumps are out of the scope of this thesis.

In contrast to prior art, we take a data-driven approach to locomotion planning and learn a controller capable of generating a broad range of motions in an interpretable and robust fashion. In particular, we turn our attention to generative modelling and representation learning. Two seminal methods from this community are Deep Variational Bayes Filter (DVBF) [16] and Embed to Control (E2C) [17]. These methods are similar in as far as they propose controllers for systems with nonlinear dynamics by solving path planning problems in structured and learnt representations. DVBF learns a manifold from images of the motion of an inverted pendulum and individual dimensions of this representation encode semantically meaningful states such as the pendulum angle and angular velocity. Therefore, dynamically consistent trajectories are generated by moving through this space. E2C learns a representation of the dynamics similarly from images of an inverted pendulum model as well as a cart-pole and a two-link manipulator. During training, E2C learns an optimal control formulation in the latent space resulting in a

locally linear representation of the observed dynamics. Similarly to DVBF, the learnt representation encodes the underlying states of the systems into the structure of the space. These approaches show that it is possible to learn a meaningful representation in which complex control problems become path-planning ones in a structured latent representation. Moreover, both these methods show that latent space planning formulations are able to infer semantically meaningful information about the underlying states of the system. Therefore, in this thesis, we investigate if the application of similar representation learning methods applied to quadruped locomotion can infer fundamental states in order to express diverse locomotion skills in a single structured latent space and if this space can be used to solve a variety of complex locomotion tasks.

Inspired by latent space planning methods [16, 17], we take a data-driven approach to learn a unified representation of locomotion across multiple gaits. This representation sufficiently captures correlations across different gait types (e.g. trot, crawl and pace) as well as across gait characteristics (e.g. swing height and lengths) from a small set of expert demonstrations. Indeed, the structure of this representation encodes the inferred gait phase relations between each leg and traversing this space solves a variety of locomotion problems including perceptively traversing terrain and continuous gait transitions via automatically discovered intermediary gaits. Inspection of the latent space reveals that the contact schedule of each gait is expressed in the representation's structure and that this space is disentangled with respect to the gait characteristics. As a result the robot's cadence, footswing lengths and heights can be controlled independently of one another using two oscillatory signals injected into two different dimensions of the latent space. By the incorporation of a rudimentary terrain encoding, we show that the structure of the space can adapt in response to the ground around the robot to adjust the robot's gait characteristics. We show that this improves the robustness of the foothold selection as the robot climbs terrain. Since we use a generative model to learn the latent representation, it is possible to estimate the likelihood of the encoded robot states. This is used to measure if a disturbance is significant enough to produce out-of-distribution behaviour e.g. when the robot is pushed aggressively by an adversary. We then leverage the unique ability of our method to vary the gait characteristics to respond to the disturbance and rapidly

increase the robot's cadence. This response is inspired by research looking at how humans respond to similar disturbances and we show that this response increase the range of disturbances our controller can reject making it more robust. As a result, we demonstrate that a learnt representation for locomotion forms an interpretable planning space with which a broad range of robust and tunable locomotion gaits are generated.

1.1 Outline And Contributions

The investigation into the challenges and opportunities of learning a unified representation for locomotion commences in Chapter 2 with a discussion of key concepts for locomotion planning and a review of prior literature in the field of quadruped locomotion. The key concepts covered in Chapter 2 focus on the equations of rigid-body dynamics. Any locomotion trajectory must satisfy the rigid-body dynamics equation in order for the trajectory to be physically feasible and work on the real robot. The related work section discusses different methods of producing solutions for locomotion which satisfy the dynamics equation. This section is structured to group methods by capability. This begins with literature designed for blind locomotion on flat ground, followed by terrain-aware locomotion and the last subsection focuses on approaches designed to combine multiple modes of locomotion. Distinguishing features of a particular method are discussed relative to similar methods designed either with the same assumptions or to tackle similar locomotion problems.

In the remainder of this section, we present an outline of the development process which resulted in the construction of a unified representation for locomotion across gaits as well as the contributions of each chapter. As mentioned, the dynamics of locomotion are complex and non-linear in places, therefore in each chapter, we make assumptions to simplify the dynamics of locomotion and challenge these assumptions in subsequent chapters.

1.1.1 First Steps: latent space Control with Semantic Constraints for Quadruped Locomotion

To begin, we apply latent space planning methods to a blind *static* crawl gait in Chapter 3 which presents *First Steps* [18]. A gait is defined as static if at every timestep the robot is statically stable meaning that if the robot comes to rest at any point during the gait cycle the robot would not fall. In particular, we use a variational auto-encoder (VAE) [19, 20] augmented with performance predictors which are multi-layer perceptrons (MLPs) attached to the latent space. The VAE's encoder infers the latent representation, whilst the decoder reconstructs the input and the performance predictors estimate which of the robot's feet are in contact and if the robot is standing in a stable pose respectively. This architecture is exposed to static snapshots of the robot in both statically stable and unstable poses to create the latent space. The total training loss incorporates the beta-VAE loss [21] summed with the binary cross-entropy (BCE) loss from the performance predictor. The BCE loss updates the VAE's encoder's weights and structures the latent representation to encode the crawl gait's contact schedule into the latent space. Indeed, the performance predictors segment the latent space such that the contact sequence of the crawl gait is embedded into the latent space. Since the performance predictors are differentiable, we have a differentiable approximation for the robot's contact dynamics and solve for locomotion trajectories using gradient-based methods applied directly to the latent space.

The primary contribution of Chapter 3 is the demonstration that a structured latent space can be used to control a complex dynamic system in both simulation and the real ANYmal B platform. Secondly, the methods in this Chapter have inspired follow-up work in the domain of manipulation. This includes planning in latent space for manipulation in constrained environments. This is demonstrated in [22] as well as [23], which shows a similar methodology applied in a closed-loop planning framework and is able to adjust its trajectories in response to moving obstacles.

1.1.2 VAE-Loco: Versatile Quadruped Locomotion by Learning a Disentangled Gait Representation

The following Chapter 4 presents *VAE-Loco* [24] which addresses blind locomotion using the dynamic trot gait. A trot gait is dynamically rather than statically stable since there are phases where the robot has only two of its point feet in contact with the ground and during this phase, the base is under-actuated and rolls about the line between the two feet. Hence, the robot uses its momentum to carry it from one full-support phase (four feet in contact) to another during the uncontrollable two feet in contact phase. The first modification to the VAE used in Chapter 3 is to encode a history of robot states in order to infer the robot's momentum in the latent space. As in *First Steps*, performance predictors attached to the latent space estimate the contact states of the feet and structure the latent space. Inspection of the space reveals that it is disentangled meaning that key gait characteristics such as footswing length and height can be varied independently by varying the amplitudes of signals injected into the latent representation. Further analysis shows that the disentanglement of the latent space with respect to the gait characteristics is a result of backpropagating the BCE loss through the VAE's encoder. This is shown by comparing the latent spaces resulting from training a vanilla beta-VAE with that of *VAE-Loco*. This reveals that footswing height and length remain correlated in the vanilla beta-VAE's case, and a full discussion of this is found in the introduction to Chapter 4 and the body of this chapter.

VAE-Loco's latent representation forms a planning space in which oscillatory drive signals are applied to produce a range of blind-trot gaits on flat ground with gait characteristics (footswing cadence and stance duration) far broader than seen during training. The parameters of the oscillatory drive signals are adjustable and map directly to the footswing characteristics. The amplitude of the primary drive signal is proportional to the footswing height, whilst the signal's time period is equal to the swing duration. The secondary trot signal's amplitude determines the footswing length, but in practice, this signal is inferred during operation. As a result *VAE-Loco* showcases the unique capability of varying the in-flight gait characteristics (footswing timings, lengths and heights) on-demand. Alternative gradient-based methods such as [3] or [4] are unable to

achieve this behaviour as optimising the contact timings makes these methods restrictively slow and hampers the ability to use these methods in real-time. Similarly, learning-based methods such as [9, 10] tend to be black-box and it is not possible to adjust the resulting gaits without retraining the models. Continuous variation of the robot's trot gait on the fly is demonstrated on the real ANYmal B and C quadruped without retraining the model and the latent space forms a planner in a closed-loop control framework deployed directly on the robot. Since the VAE is a generative model, it can be used to determine out-of-distribution inputs resulting from disturbances or pushes. During locomotion, the evidence lower-bound is monitored and large disturbances produce large spikes in this value. On detection of a disturbance, the cadence of the robot is rapidly increased and it is shown that this permits the rejection of larger perturbations.

In summary, the contributions of Chapter 4 are firstly showing that using oscillatory drive signals in a learnt latent space uniquely permits the continuous variation of the in-swing gait characteristics on-demand for trot locomotion. Secondly, the capability of rapidly increasing the footswing speed is used to help reject disturbances which are detected by monitoring the ELBO during operation. This reaction mechanism helps to show that *VAE-Loco* is a robust planner for locomotion. This chapter investigates the latent space structure and reveals that backpropagating the BCE loss through the VAE's encoder is responsible for disentangling the space with respect to the gait characteristics. At this point, *VAE-Loco* demonstrates that a latent space representation is a suitable planning space for generating and interpolating between the characteristics of a variety of different trot gaits. It is also shown that this forms a robust controller when *VAE-Loco* is a planner in a closed-loop control framework. Finally, the structure of the learnt latent space is also interpretable and the resulting gait characteristics are predictable.

1.1.3 Gaitor: Learning a Unified Representation Across Gaits for Real-World Quadruped Locomotion

Inspired by *VAE-Loco's* ability to represent the robot's contact schedule in the structure of the latent space, we posit that the VAE should be able to produce a unified representation for locomotion across *multiple* gaits. Indeed, Chapter 5 presents *Gaitor* which creates

a single latent representation for three gaits (trot, crawl, and pace). Investigation of the latent structure reveals that continuous transitions between gaits emerge on-demand via automatically discovered intermediary contact patterns. In particular, the VAE infers phase-based correlations between all three gaits and uses this to learn a two dimensional and disentangled relationship between the gaits. A result of the VAE exploiting the phase relation between gaits to create a compact encoding is that intermediate contact schedules between the gaits automatically arise as a different desired gait is demanded. The methods in *VAE-Loco* were reserved for blind locomotion on flat ground. To address this limitation, *Gaitor* is conditioned using a rudimentary encoding of the terrain around the robot. Further analysis of the VAE's latent representation reveals that as the terrain around the robot varies, the structure of the space adapts meaning that the robot's gait characteristics adjust. It is observed that as the robot is about to climb onto an obstacle, the latent space will change to promote footswing lengths and the robot will place its feet in a more robust location further away from the edge of the obstacle. This means that the initial step onto the terrain will be further away from the edge of the step. In order to maximise the utility of this behaviour, a learnt planner operates in the latent space. This adjusts the latent space trajectory which alters the robot's gait characteristics in response to changes in terrain. The VAE and the learnt planner are deployed as a real-time MPC-style planner in a closed-loop control loop. We show that this controller is able to traverse significant obstacles.

The contributions of Chapter 5 are first to show that a unified latent representation across gaits can be used to generate continuously variable gait transitions from trot to crawl to a pace/crawl hybrid via automatically discovered intermediary gaits. The resulting latent space is disentangled for all learnt gaits meaning that an elliptical trajectory in a 2D slice of latent space is sufficient to produce the different gait types and generate perceptive locomotion to traverse significant obstacles in front of the robot. The work presented at this point in the thesis shows that a latent representation can form a planner suitable for generating a broad range of gait types (trot, crawl and a crawl/pace hybrid). Furthermore, the gait characteristics (swing lengths and heights) can be varied in response to the terrain ahead of the robot facilitating the climbing of obstacles. The learnt latent

space is readily interpretable also. Therefore, *Gaitor* shows that a unified representation for locomotion permits the creation of a single controller which achieves our design specifications: a method which can generate and interpolate between a broad range of motions, whilst being robust and interpretable.

1.1.4 From Primates to Robots: Brain-like Latent Dynamics Emerge in Robot Systems during Walking and Reaching

The methods in *Gaitor* show that a unified representation of locomotion across gaits naturally leads to continuous transitions between different gait types and when conditioned using a rudimentary terrain encoding can perceptively traverse significant obstacles. In Chapter 6, we investigate if there exist similar representations for locomotion in nature. In doing so, we discovered work by Churchland et al. [25], which finds that both locomotion and reaching tasks are encoded as oscillations in the motor cortices of primates. Inspired by this, we compare the similarities between the signals used to control the real quadruped and those found in the motor cortex. Churchland et al. focus their research on the analysis of primate reaching which is similarly instigated by rotational dynamics in the brain. Therefore, we apply rotational signals similar to those used to control locomotion in a manipulation-specific latent representation and show that we can control the robot manipulator to solve reaching tasks in the real world. Inspection of the manipulation latent space reveals that two independent dimensions of this space encode longitudinal and lateral motions of the end-effector. This result mirrors the disentanglement of the locomotion latent space. In essence, the learnt representation is suitable not only for estimating the robot's state but also for planning complex motions. Having shown the similarities between the oscillatory signals found in the brains of primates and in the latent space, lesioning pathologies are explored in the context of the in-silico system. This investigation reveals how latent space correlations propagate through the decoder and shows that there is a high degree of locality in that network.

The primary contribution of Chapter 6 is to compare the similarities between the signals in the brain during walking and reaching to those in the locomotion and manipulation latent spaces. Secondly, it is shown that the manipulation latent space is

similarly structured to the locomotion-specific space meaning that longitudinal and lateral movements of the robot's end-effector are encoded into two independent dimensions of the latent space. As a result, planning using two oscillatory signals in this space solves reaching tasks on the real robot.

1.2 Publication List

The list of publications and submitted work carried out during the thesis is included in this section. There are a few papers which do not fit in the scope of this thesis and these are included in the second list below. The thesis includes the following publications:

1. *A. Mitchell, M. Engelcke, O. Parker Jones, S. Ganagapurwala, O. Melon, D. Surovik, I. Havoutis, I. Posner.* "First Steps: Latent Space Control with Semantic Constraints for Quadruped Locomotion". Published at IROS 2020

2. *A. Mitchell, W. Merkt, M. Geisert, S. Ganagapurwala, M. Engelcke, O. Parker Jones, I. Havoutis, I. Posner.* "VAE-Loco: Versatile Quadruped Locomotion by Learning a Disentangled Gait Representation". Published in IEEE Transaction on Robotics (T-RO) 2023.

3. *A. Mitchell, O. Parker Jones, J. Yamada, W. Merkt, I. Havoutis, I. Posner.* "From Primates to Robots: Emerging Oscillatory Latent Space Dynamics for Sensorimotor Control". Published in Conference on Cognitive Neuroscience (CCN).

4. *O. Parker Jones*, A. Mitchell*, J. Yamada*, W. Merkt, M. Geisert, I. Havoutis, I. Posner.* "Oscillating latent dynamics in robot systems during walking and reaching". Published in Nature Science Reports

5. *A. Mitchell, W. Merkt, A. Papatheodorou, I. Havoutis, I. Posner.* "Gaitor: Learning a Unified Representation Across Gaits for Real-World Quadruped Locomotion".

We also include the publication *Next Steps* [26] in the Appendix as this is the conference version of *VAE-Loco*. All the information and results in *Next Steps* are reproduced and analysed in greater depth in the journal version *VAE-Loco*.

1. *A. Mitchell, W. Merkt, M. Geisert, S. Ganagapurwala, M. Engelcke, O. Parker Jones, I. Havoutis, I. Posner*. "Next Steps: Learning a Disentangled Gait Representation for Versatile Quadruped Locomotion". Published in ICRA 2022

I have also been second/third author of two other publications. For these works, I assisted in the experimental work only so do not include these as chapters in my thesis.

1. *S. Gangapurwala, A. Mitchell, I. Havoutis*. "Guided constrained policy optimization for dynamic quadrupedal robot locomotion." Published in Robotics and Automation Letters 2020

2. *A. Papatheodorou, W. Merkt, A. Mitchell, I. Havoutis*. "Momentum-Aware Trajectory Optimisation using Full-Centroidal Dynamics and Implicit Inverse Kinematics". Submitted to IROS 2024

2
Background and Related Work

To solve real-world locomotion problems, it is necessary to generate trajectories which satisfy the underlying dynamics equations of a quadruped system. In this section, we introduce the underlying dynamics of the quadruped robot, and some key definitions of locomotion followed by a discussion of prior work in the field of locomotion.

2.1 Background

The design of the quadrupeds used in this thesis has a significant impact on their dynamics. In this thesis, ANYmal B and ANYmal C are used. These have floating bases and four legs with three actuated joints. The joints are linked with rigid bars and the robot has point feet. As a result, the robot's locomotion dynamics can be explained using rigid-body dynamics. The robot's base has six degrees of freedom, three for position and three for its orientation. Each joint has one degree of freedom meaning that the total degrees of freedom are 18. Therefore, the base is underactuated meaning that the base degrees of freedom cannot be directly controlled but instead, the legs control the base pose. The robot has point feet only meaning that the feet cannot exert a moment on the ground when in contact. This also has the consequence that the robot can only fully control the base's angular momentum when three or more feet are in contact with the terrain. Contact with the terrain is defined when the velocity and acceleration of the robot's

feet are zero meaning that the force due to friction between the feet and the ground are equal to the end-effector forces in the plain of the terrain. Finally, the feet can only exert pushing forces and cannot pull the robot forward.

Locomotion is characterised by the continuous making and breaking of contact. This leads to contact schedules with swing characteristics and distinct gait types. Contact schedules govern the timings between the swing and stance phases. Swing phases occur when the feet are not in contact, whilst stance denotes that a foot is in contact. Gait characteristics are qualities of locomotion such as swing length, height and duration. Distinct gait schedules often seen in nature are named and the ones discussed in this thesis are trot, crawl and pace. Trot gaits are common in literature and are characterised by four feet in contact followed by two feet in contact. During the trot gait, the pairs of feet in contact are arranged diagonally so that if the left front is in contact, so is the right hind and vice versa. The contact schedules of these gaits affect the robot's dynamics. For example, during trot and pace, the robot has no control over the robot's roll about the support line.

2.1.1 Rigid-Body Dynamics

The equation of dynamics for robots with rigid bodies balances all the forces and torques applied to the robot. As mentioned, the quadruped is a floating-base robot since the base is not directly actuated. To explain the floating-base dynamics of the robot, the complete joint state $\mathbf{q}$ is a concatenation of the centre of mass (CoM) pose $\mathbf{c}$ relative to an inertia frame, which has six degrees of freedom and the angles of the joints $\{q_0, \ldots, q_n\}$, where n is the number of actuated joints and this makes $\mathbf{q} = [\mathbf{c}^T, q_0, \ldots, q_n]^T$. As a result, the dynamics equations describe the linear and angular forces applied to the robot's centre of mass and the torques applied to the joints. The total torques and forces on the robot can be described using the following equation:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{S}^T\boldsymbol{\tau} + \mathbf{J}_c^T(\mathbf{q})\boldsymbol{\lambda}. \tag{2.1}$$

This Equation (2.1) comprises of the following variables:

- $\mathbf{q} \in \mathbb{SE}(3) \times \mathbb{R}^n$: is the complete joint configuration including the CoM position $\mathbf{c} \in \mathbb{SE}(3)$ and the n joint angles $\{q_0, \ldots, q_n\}$. Hence, $\mathbf{q} \in \mathbb{R}^n$.

- $\dot{\mathbf{q}}$, and $\ddot{\mathbf{q}}$: are the full-state joint velocities and accelerations respectively.
- $\mathbf{M}(\mathbf{q}) \in \mathbb{R}^{(6+n)\times(6+n)}$: the inertia matrix.
- $\mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^{6+n}$: the Coriolis and body force vector, often referred to as second-order or nonlinear effects vector.
- $\mathbf{S} \in \mathbb{R}^{n\times(6+n)}$: actuator selector matrix.
- $\boldsymbol{\tau} \in \mathbb{R}^{n}$: are the torques applied at the joints.
- $\mathbf{J}_c(\mathbf{q}) \in \mathbb{R}^{(n_c)\times(6+n)}$: is the contact force Jacobian, where n_c is the number of contact constraints.
- $\boldsymbol{\lambda} \in \mathbb{R}^{n_c}$: external contact force vector.

The terms on the left-hand side of Equation (2.1) include the mass matrix $\mathbf{M}(\mathbf{q})$ multiplied by the full-state joint acceleration $\ddot{\mathbf{q}}$ summed with the second-order effects term $\mathbf{h}(\mathbf{q}, \dot{\mathbf{q}})$. The inertia matrix is a diagonally dominant matrix representing the masses of each of the robot's bodies such as the mass of the robot's base or links along the diagonal. The off-diagonal terms of the mass matrix relate the acceleration of a particular body to another distinct body. The mass matrix is a function of the joint configuration $\mathbf{q}$ and varies with $\mathbf{q}$. The nonlinear effects matrix $\mathbf{h}(\mathbf{q}, \dot{\mathbf{q}})$ accounts for a few physical quantities such as the Coriolis effects due to a moving body with inertia and the force from gravity. The right-hand side of Equation (2.1) includes the applied joint torques $\boldsymbol{\tau}$ and the contact forces. Since the robot's base is underactuated, there are no torques applied to the CoM, hence $\boldsymbol{\tau} \in \mathbb{R}^n$. The selector matrix $\mathbf{S}$ projects the applied joint torques into the joint-state frame and as such comprises a diagonal matrix with ones for the actuated elements and zeros everywhere else. The contact forces $\boldsymbol{\lambda}$ are also translated to joint torques by using the contact Jacobian $\mathbf{J}_c(\mathbf{q})$. For example, for a point foot, which can only exert linear forces, the contact force $\boldsymbol{\lambda} \in \mathbb{R}^3$ is comprised of $\boldsymbol{\lambda} = [\lambda_x, \lambda_y, \lambda_z]^T$, where λ_z is a force normal to the contact plane and the other forces λ_x, λ_y are tangential to the contact plane.

The quadrupeds used during the development of the methods in this thesis have series elastic actuators (SEA) [27] at each joint. These actuators produce torque using an electric motor with a reduction drive. A spring clutch provides torque sensing and dampens large impacts due to contact changing. Since the actuators are mounted at the joints, the legs

constitute a significant proportion of the mass. Therefore, more dynamic manoeuvres with faster leg movements induce high Coriolis effects. Other robots such as *Spot* or *MIT Cheetah* [28] focus most of the mass near the robot's base. These robots have four-bar linkages to transfer torque from the hips to the knees.

The SEA actuators have complex dynamics. The SEA dynamics can be modelled as a fourth-order resonant system with close to critical damping [27]. These dynamics must be accounted for when solving locomotion planning problems. Similarly, the robot's feet are rubber spheres modelled as points and without contact sensing. The contact forces are instead inferred from the joint torque residual. In practice, these feet do deform during operation to distribute the robot's load across the ground, but this tends not to be included in the equation of dynamics or modelled in simulation. The dynamics of the SEA actuators distinguish the ANYmal series of robots from alternatives such as Spot or MIT Cheetah. The latter two robots use reduction gearboxes and electric drive without the spring actuator. Due to differences between the actuation dynamics of ANYmal and other similar platforms, this thesis focuses on methods in literature which deploy on the ANYmal series of robots as transferring methods deployed on alternative platforms can be non-trivial or the resulting performance can have different operating windows when deployed on the ANYmal robot.

2.2 Related Work

In the next sections, related work in the field of locomotion is discussed and grouped by capability. This begins with methods for blind locomotion over flat ground, followed by approaches for terrain traversal, and concludes with a discussion of methods which combine various locomotion skills together.

2.2.1 Flat Ground Operation

The methods discussed in this section are designed for blind quadruped locomotion on flat ground. To begin, reduced-order models of the robot's dynamics are discussed. These models are used to solve for trajectories in task space, and these trajectories are tracked using a gradient-based whole-body controller. The formulation of this tracking

controller is also discussed. Reduced-order models tend to have fast convergence times meaning that they are suitable for real-time deployment. Nevertheless, there are some limitations to these methods which are discussed also. During the development of these algorithms, the first RL methods deployed on the real ANYmal B appeared [9]. Prior work from this community was confined to simulation only for ANYmal B [29]. The RL community favoured the development of on-policy and model-free algorithms for locomotion control. The majority of the methods discussed here take this form due to their prevalence in literature. These methods delivered robust and performant locomotion, but there are some limitations to these methods.

Reduced-Order Dynamics Models

Reduced-order models simplify the rigid-body dynamics in Equation (2.1) with the aim of speeding up the convergence rates of a trajectory optimisation framework. The first assumption of these types of models is massless limbs. The robot's mass is assumed to be concentrated at the centre of the mass of the base. As a result, the Coriolis effects on the limbs are ignored.

The dynamics of the base can be modelled in different ways. A conceptually simple example of this is to model the CoM of the floating base as the top of an inverted pendulum [30] with a constant height, where the bottom of the pendulum is located at the centre of pressure (CoP). This is used in [31] to create static crawl gait trajectories. The contact dynamics are enforced using friction cone constraints and using a zero-moment point (ZMP) model [6]. A ZMP constraint is designed to model the dynamics of the feet in contact and enforces that a foot in contact exerts no moment about the terrain. For any type of foot, the foot rolls only and applies zero torque to the ground. For an inverted pendulum model, the CoP is constrained to be the location of the ZMP. The base and feet trajectories are constrained using splines, while foothold locations and timings are computed in advance. The trajectories are converted to joint positions using inverse kinematics and joint torques by solving an inverse dynamics program. An example inverse dynamics formulation can be found in [32]. Inverted pendulum and ZMP constraints provide analytical gradients and can be implemented so that optimisation programs which

use these models converge fast enough to be used as real-time planners in a closed-loop control framework. Planners usually operate at a frequency of $50\,\mathrm{Hz}$ and the control loop tends to run a target frequency of $400\,\mathrm{Hz}$.

During more dynamic motions, the base will have significant inertia which is not modelled using an inverted pendulum alone. Therefore, subsequent publications after [31] tend to model the base and its inertia using centroidal dynamics [7]. The linear dynamics of the CoM are a sum of the forces applied at the base e.g. the force from gravity as well as those from the feet. The angular dynamics are the sum of the torques applied to the CoM. This includes the moment resulting from the contact forces, which are equal to the angular acceleration multiplied by the rotation inertia matrix summed with the torques resulting from the angular velocity of the base. The base is modelled to have constant rotational inertia and the massless limbs assumption persists.

Modelling the base angular inertia using centroidal dynamics introduces nonlinear and non-convex constraints. Therefore, a nonlinear dynamics solver is required to solve a gradient-based program. An example of a trajectory optimisation framework using centroidal dynamics is [8]. The trajectories generated using these types of constraints are tracked using a whole-body controller (WBC) [33] which solves an optimisation to find the joint torques which are sent to the actuators. This is a hierarchical formulation where constraints can be prioritised. The WBC constraints enforce the centroidal dynamics constraints, the inverse kinematics and the contact dynamics. As mentioned, optimisation methods which use these types of constraints can converge fast enough to be deployed at a control frequency of $400\,\mathrm{Hz}$. A limitation of this class of methods is that they reduce the operating window of the robot's performance. For example, these methods are unable to adjust *in-flight* gait characteristics such as footswing speeds, heights and lengths. This limitation is addressed in Chapter 4.

Reinforcement Learning and Quadruped Locomotion

The reinforcement learning (RL) community in the field of locomotion is a large and growing one. In this subsection, a short overview of RL components common to all methods is discussed prior to introducing specific approaches from the literature. RL

involves solving the Bellman equation to maximise an optimal return function over the set of possible states using dynamic programming. The Bellman equation is a goal-driven value function that relates the state of a system to the expected cumulative reward of being in that state. Dynamic programming refers to finding an optimal policy or trajectory to maximise the value function. RL maximises the Bellman equation using exploration to reach a particular goal and receives delayed rewards to learn an optimal policy. This framework models state evolution using Markov decision process (MDP) [34] meaning that each state is a function of the previous action and state only. An action is an input to the system and steers the state trajectory. During the exploration phase, an RL agent receives a reward from the environment either every timestep in the dense case or once the agent reaches a particular goal state in the sparse case. During this process, the agent learns a value function which as mentioned relates the current state to the expected cumulative reward for being in that state. Therefore, the optimal policy leads to an optimal state-space trajectory which maximises the value function. Since the RL methods employ an exploration phase, RL is globally asymptotically optimal across the state space [35]. The last component of RL is an optional model of the system's dynamics. Methods which use an explicit model are called model-based and use this model is used to predict the next state in order to learn the value function and policy. Alternatively, model-free methods learn a value function and find a policy using an implicitly learnt model of the dynamics from observing the evolution of the system's dynamics during exploration. The vast majority of RL methods applied to locomotion and deployed on the ANYmal robot are model-free and on-policy. These are discussed in the following sections.

On-Policy Model-Free Reinforcement Learning

Despite the field of RL producing a broad range of different methodologies, locomotion RL practitioners tend to favour model-free on-policy methods. Some of the early RL works deployed on a real ANYmal robot for dynamic locomotion are found in [9] and [10]. These methods use an on-policy learning approach, namely proximal policy learning [36] to create a policy for locomotion. The latter work [10] also introduces constraints during training to improve and stabilise the policy. Before the works in [9] and [10], most RL

methodologies were limited to simulation only [29, 37, 38] with significant performance deterioration when applied to the real system. This phenomenon was a result of dynamics, which were not modelled adequately in simulation. In general, RL policies exploited simulated contact dynamics. Prior to [9], these dynamics were almost binary on/off in nature, but with the advent of a novel simulator *Raisim* [39] the contacts were modelled differently. Indeed, the contacts are modelled as slightly deformable meaning that contact is only broken once the feet are lifted above the terrain. In addition, ANYmal has series elastic actuators (SEAs) [27] which themselves are complex systems with very specific responses. In general, these can be modelled as fourth-order resonant systems with close to critical damping. The work in [9] modelled these actuators using neural networks in Raisim [39]. The contact and actuator dynamics provided greater fidelity to the simulator and facilitated simulation to real robot transfer. These RL methods produce robust motion with impressive capabilities. However, these model-free methods are black-box methods and it is not possible to adjust the learnt behaviour. For example, gait parameters such as cadence and footswing heights and lengths are baked in and cannot be adjusted without retraining the policy.

Latent Space Planning for Locomotion

Latent space planning methods learn a representation of a system's dynamics meaning that control problems can be solved using trajectories in this space. To achieve this, latent space planning methods learn a statistical model of the training data using a generative model. One of the key advantages of latent space planning methods is that these approaches are effective at inferring key insights about an observed system. A locomotion-specific example of this can be found in [40] which tackles sim-to-real transfer for both the Daisy hexapod and Yumi dual-arm robots using extremely few trials to transfer to the real robots. This method learns a latent representation of simulated locomotion trajectories using a sequential VAE. The training data includes both desirable and undesirable examples from a simulated central pattern generator (CPG) based controller for the locomotion tasks. A Bayesian optimisation (BO) framework is then employed to optimise a trajectory in the space and this is deployed on the robot. The authors show that this approach is able

to generate feasible trajectories and the real Daisy robot is able to walk along a narrow walkway successfully with fewer than ten trials. Subsequent ablation studies find that BO in the latent space finds optimal controllers with fewer trials than operating in the raw state space. One of the crucial conclusions of this work is that the latent representation infers meaningful features for locomotion and this accounts for the improvement in planning performance when using the latent space rather than the representation being simply a compression of the robot's state space.

2.2.2 Terrain Traversal

One of the design decisions common with previous works discussed so far is that they are limited to blind operation over flat ground only. In this section, a range of methods to integrate vision and produce terrain-aware locomotion policies are discussed.

Model-Based In Combination With Model-Free Methods

The first group of methods for perceptive terrain traversal uses learning-based methods to solve foothold selection problems using a model-free RL framework and a model-based optimisation method to find task-space trajectories. An example of this is RLOC [41] which uses a height map of the terrain around the robot and a desired base heading to find foothold locations given a particular contact schedule. The foothold locations form constraints for *Dynamic Gaits* [8] which finds trajectories of the robot's base and feet. As discussed in the previous sections, modelling the quadruped dynamics is feasible and reduced-order models are effective and fast for a broad range of operations. However, modelling the variety of different terrains by hand can be complex and a model-free RL framework can explore and infer the value of foothold locations in various different environments.

Terrain-Aware Task-Space Optimisation

A general-use task-space optimisation is proposed in *TOWR* [3]. This framework is designed to solve locomotion trajectories for any legged system. *TOWR* generates centre of mass (CoM) and end-effector trajectories using gradient-based optimisation.

A centroidal dynamics model is used to approximate the dynamics of the robot's torso, whilst the legs are assumed to be massless. The ZMP criterion and an inverse pendulum model are both used as constraints for the simplified dynamics. The methods in Chapter 3 and Chapter 4 are compared to *TOWR*'s formulation. As a result, this formulation is fleshed out in detail in this section. Additionally, *TOWR* presents a clear formulation for a general task-space optimisation and can be used to highlight some specific limitations common to similar methods. A general form formulation for quadruped locomotion similar to *TOWR* is included below:

$$\min_{\mathbf{u}\in\mathcal{U}} \sum_{k=0}^{N-1} \|\mathbf{x}_k^{\text{ref}} - \mathbf{x}_k\|_{\mathbf{Q}_k}^2 + \|\mathbf{u}_k^{\text{ref}} - \mathbf{u}_k\|_{\mathbf{R}_k}^2 + \|\mathbf{x}_N^{\text{ref}} - \mathbf{x}_N\|_{\mathbf{Q}_N}^2 \tag{2.2}$$

$$\text{s.t.} \quad \mathbf{x}_{k+1} = \mathbf{x}_k + \int_{t_k}^{t_k+dt} \mathbf{f}\left(\mathbf{x_k}, \mathbf{u_k}\right) \mathbf{dt} \qquad \text{(Dynamics)} \tag{2.3}$$

$$\underline{\mathbf{u}} \leq \mathbf{u} \leq \bar{\mathbf{u}} \qquad \text{(Control-Input Bounds)} \tag{2.4}$$

$$\text{if foot } i \text{ in contact:} \tag{2.5}$$

$$\mathbf{r}_{\perp \text{z}}(k) \geq \mu \mathbf{r}_{\text{z}}(k) \qquad \text{(Friction Cone)} \tag{2.6}$$

$$\dot{\mathbf{p}}_i(k) = \mathbf{0} \qquad \text{(No Slip Constraint)} \tag{2.7}$$

$$\boldsymbol{\lambda}_i(k) \cdot \mathbf{n}(\mathbf{p}_i^{xy}) \geq 0. \qquad \text{(Unilateral Normal Force)} \tag{2.8}$$

The cost function above minimises the two-norm between a reference trajectory $\mathbf{x}_k^{\text{ref}}$ and the optimised version $\mathbf{x}_k$ and is regularised using the diagonal matrix $\mathbf{Q}$ summed with a cost between the input and a reference $\mathbf{u}_k$ and $\mathbf{u}^{\text{ref}}$ regularised using another matrix $\mathbf{R}_k$ and finally a terminal cost minimising the two-norm between terminal state $\mathbf{x}_N$ and desired terminal state $\mathbf{x}_N^{\text{ref}}$. The cost is summed over discrete states over a horizon of N future times steps sampled from the current time step k. The state $\mathbf{x}_k$ for a task-space formulation includes the base and feet positions and velocities. For example, the reference trajectory can be a CoM trajectory from the initial state to the final. The robot's centre of pressure (CoP) is modelled as in the input of the system $\mathbf{u}_k$, and is commonly used in inverted pendulum formulations [31]. The first constraint in Eq. (2.3) represents the robot's dynamics, which in this case is a linearised function with dynamics $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u})$. The dynamics function is discretized by integrating $\mathbf{f}$ over the duration dt which is equal to the time between timesteps, i.e. the duration $t_{k+1} - t_k$.

The next constraint is optional and applies upper and lower bounds on the input $\mathbf{u}_k$, which are represented by $\underline{\mathbf{u}}$ and $\bar{\mathbf{u}}$ respectively.

As mentioned already, there are contact constraints which are only active when the feet are in contact. The contact constraints are represented here using three equations from Eq. (2.6) to Eq. (2.8) and include a friction constraint, a no-slip condition for the velocity of the foot, and a push force only constraint. The friction constraint is modelled using a friction cone formulation. Therefore, the reaction forces in the plane of the terrain $\mathbf{r}_{\perp \mathrm{z}}(k)$, which are perpendicular to the normal force $\mathbf{r}_{\mathrm{z}}(k)$ must be less than the static friction coefficient μ times the normal reaction force from the terrain $\mu \mathbf{r}_{\mathrm{z}}(k)$. Since the foot is in contact, the velocity $\dot{\mathbf{p}}_i(k)$ of foot i is zero during the stance duration. Finally, the last contact constraint in this formulation is the unilateral force constraint. The normal reaction force at the location of foot i is $\mathbf{n}(\mathbf{p}_i^{xy})$, where $\mathbf{p}_i^{xy}$ is the location of the foot in the plane of the terrain and the positive x-direction is inline with the direction of the robot's motion. The normal reaction force dot-product with the contact force $\boldsymbol{\lambda}_i(k)$ is therefore greater than zero for push-only forces. The contact constraints are nonlinear since the robot is only able to apply push forces and cannot pull at the terrain. This is a result of the feet geometry which can generate a high friction force when pushing rather than pulling. The contact forces also rapidly drop to zero during the swing phase meaning that these dynamics have non-smooth and undefined gradients without a differentiable approximation.

TOWR uses the formulation above in Eq. 2.2 without the cost function. Despite the *TOWR* formulation being generalist and intuitive it does have some limitations. *TOWR* can be slow to converge meaning that it is not a feasible solution for real-time applications and replanning is not possible. This is accentuated when the foothold timings and locations are added to the optimisation. Therefore, these methods tend to be used with pre-determined foothold locations and timings. The formulation requires switching between multiple constraint sets as the robot takes steps, which is conceptually complex. Instead, this thesis investigates learning a representation for locomotion where contact constraints are encoded in sequence in the latent space. Therefore, locomotion trajectories are generated by traversing the learnt representation. Encoding contact

schedules into latent space commences in Chapter 3. Exploration of varying the in-flight contact timings is contained in Chapter 4.

Terrain-Aware Joint-Space Optimisation

In recent years, optimal control methodologies which model the entirety of the robot's dynamics have emerged. Examples of these methods include [2, 4]. As is mentioned briefly in the introduction, the methods proposed by Mastalli et al. use an analytically derived value function which encompasses the quadruped's rigid-body dynamics. The desired locomotion trajectory optimises this value function using an optimisation framework. The value function is linearised about the current robot state and differential dynamic programming (DDP) [5] is used to find a solution. DDP uses a second-order method and gradient descent to optimise the value function and hence solve the Bellman equation. The terrain heights around the robot influence the contact constraints permitting terrain-aware locomotion. The Jacobians and Hessians of the value function are derived analytically. Since a gradient descent method is employed, the solution found using DDP is locally optimal. However, this method is fast enough to find trajectories for agile manoeuvres such as jumps. Despite using the entirety of the dynamics to optimise the trajectory, these methods still rely on pre-set, fixed contact schedules. Hence, varying the contact timings of the in-flight feet is beyond their capabilities. This limitation is addressed in Chapter 4 and continuous variation of the robot's gait parameters is leveraged in Chapter 5 for perceptive terrain traversal.

Gait Phase As An Inductive Bias

Inductive biases are popular for controlling the locomotion contact sequence. This can be seen in [11] which is an RL framework utilising the phases of each leg to generate control policies. The swing trajectories of the feet are predicted as the action from the policy. The feet trajectories are converted to joint space using an analytical inverse kinematics formulation. The joint-space trajectories are sent to a learnt position and derivative (PD) controller. The training regime of this work is also notable as it uses a teacher policy with access to privileged information to train a student with access to

only proprioceptive data. The student is highly effective at terrain traversal and can operate in challenging outdoor environments.

The methods in [11] are highly capable on uneven terrain despite being a proprioceptive policy only. Therefore, the next logical innovation is the integration of perceptive data. This extension occurs in [12] and makes use of the same phase-based formulation as in [11]. Vision information is incorporated using a terrain encoder and a similar teacher-student training framework is employed to train the final policy. The teacher policy has access to privileged information such as the actual terrain heights of the simulated terrains, whilst the student policy infers this information from the terrain encoding. The terrain encoder incorporates a belief encoder, which privileges proprioceptive information over perceptive data and is trained to categorise how useful the vision data is for locomotion. The trained student policy is deployed on the real ANYmal C robot zero-shot and the authors demonstrate robust locomotion in real-world environments.

The methods developed in [11] and [12] show that the phase and relative phase of the legs form a useful inductive bias for learning-based controllers. In this thesis, we posit that the phase of each leg can be inferred and used to create a unified representation of locomotion across gaits. This is shown in Chapter 4 and Chapter 5. The former chapter creates a latent space for the trot gait, which can generate continuous variation in the in-flight gait characteristics. The latter chapter extends this and shows that a similar representation across gaits can find novel transitions between distinct skills. The addition of a rudimentary terrain encoding permits traversal of uneven terrains.

2.2.3 Singular Policies For Multiple Skills

In previous sections, the methods presented produced locomotion trajectories optimised for specific scenarios or gaits. As a result, multiple specialised controllers are required to tackle specific tasks. This section highlights methods which generalise to a broad range of different situations. For example, this includes transitions between discrete skills such as gait types.

Top-Down Methodologies

Top-down methods use an inductive bias or a hierarchical formulation to solve complex problems such as gait switching. The methods in [42] tackle gait switching and use a phase-based inductive bias for parameterising the contact schedule. The policy predicts the relative phases between legs. An MPC formulation solves separate optimisations to predict the swing trajectories of the legs. This work is relevant to this thesis as it produces gait switching. However, the relative phases of each gait are used explicitly as an inductive bias. In contrast, we infer the gait phase from examples of individual gaits resulting in smooth and natural-looking gait switching. Complete details are found in Chapter 5.

An example of a hierarchical approach to gait switching is proposed in [13] and trains five discrete sub-policies designed for specific manoeuvres. These sub-policies include walking, climbing and jumping onto obstacles and are selected using a separate terrain-aware policy. This super policy selects the desired policy given the terrain and a desired location for the robot. The authors use this method to produce highly agile manoeuvres to traverse a real-world obstacle course. A limitation of this impressive work is that no information is shared across the gait sub-policies meaning that continuous switching between skills is infeasible. Locomotion gaits are highly correlated meaning that a shared gait representation could be leveraged to blend between the different skills. Learning a shared representation across gaits is explored in Chapter 5 of this thesis.

Bottom-Up Methods

An alternative to the top-down methods is *Locomotion-Transformer* [14]. This is a black-box method and uses a transformer model [15] to create a multi-modal policy across multiple locomotion manoeuvres. This yields similarly impressive behaviour to [13] using a single terrain-aware policy. However, *Locomotion-Transformer* is not easy to interpret. In this thesis, the opportunities and advantages of an interpretable representation across gaits are explored. This reveals a readily interpretable representation which is meaningfully disentangled and encodes information shared across gaits. In doing so, the representation learning method is able to generate continuous transitions across discrete gaits via novel and automatically discovered intermediary gaits.

Latent Space Planning

As mentioned, latent space planning methods are effective at inferring the underlying states of the system [16, 17]. An approach in this space for locomotion planning across multiple gaits is presented in [43] and was conducted contemporaneously with the development of methods in Chapter 5. The methods in [43] learn a space across multiple gaits such as trot, crawl and pace and samples from this representation condition downstream tasks. This is used for gait switching and traversal of a variety of terrains. The gaits used to learn the latent space are generated using a phase-base contact scheduler similar to [42]. The gait encoder takes the phase of each leg as input to predict an encoding. This encoding is regularised and constrained to fit onto the surface of a unit hypersphere. During deployment, the gait encoding, base velocity and encoding of the previous robot states form input to a low-level policy. This policy predicts the next joint state which is sent to the actuators. Additionally, a learnt discriminator ensures that the resulting joint-state motion resembles a gait seen during training.

Though the methods in [43] are designed to tackle similar problems to Chapter 5, the resulting latent representations are different. Chapter 5 presents *Gaitor*, which learns a representation for quadruped locomotion across multiple gaits. The latent representation in *Gaitor* explicitly infers the gait-phase relationships between the legs across gaits. This gait phase is inferred from a history of proprioceptive robot states rather than being input to the encoder as is the case in [43]. In *Gaitor*, the latent space forms a planning manifold in which elliptical trajectories are injected. These are subsequently decoded to create locomotion trajectories. In contrast, the gait latent representation in [43] is used to condition a downstream policy which sends actions to the actuators.

3

First Steps: Latent-Space Control with Semantic Constraints for Quadruped Locomotion

The methods in this chapter produce a structured latent space specifically designed to capture the contact schedule and dynamics of the static crawl gait. As mentioned in the introduction, locomotion dynamics are non-linear and non-smooth during contact switching as the robot takes steps. The non-linearities arise from the rapid changes in contact forces as the robot steps as well as from the torque profiles required when the no-slip contact condition is enforced. In general, optimisation-based methods fit a smooth and differentiable function to approximate these dynamics and use predetermined contact schedules in order to make the optimisation solve fast enough to be deployed in a real-time control loop. For example, the formulation presented in *TOWR* [3, 44] is capable of optimising the contact timings, but this makes the convergence times prohibitively slow for real-time deployment. Hence, predetermined contact timings are used in practice. Using a predetermined contact schedule facilitates direct multiple shooting between contact changes, but these schedules tend to use reduced-order approximations of the dynamics [8, 31] and are often overly conservative to reduce the risk of creating infeasible foot timings.

In contrast to prior art, *First Steps* creates a latent representation for the crawl gait. The latent space is created such that the crawl gait contact schedule is embedded

into the structure of the space. To do this, static snapshots of the robot's state are exposed to a variational autoencoder (VAE) [19, 20] modified to use two additional multi-layer perceptrons called *performance predictors* attached to the latent space. The performance predictors estimate the probability that key dynamical constraints are active. In particular, the performance predictors use the latent state as input to predict if the robot is firstly statically stable and secondly the probabilities of which feet are in contact. The training loss is a combination of the beta-VAE loss [21] summed with the binary cross-entropy loss (BCE) for estimating the contact states and the robot's static stability using the performance predictors. During training the BCE gradients from optimising the performance predictors pass through the encoder to update its weights. This creates distinct clusters in the latent space. These clusters group latent samples by contact state and by the static stability criterion.

The performance predictors are differentiable approximations of the robot's static stability and contact dynamics. Therefore, gradient-based methods can be used to find latent space trajectories for static locomotion. For example, in *First Steps*, it is demonstrated that it is possible to find a feasible trajectory from an unstable robot pose to a stable one simply using gradients from the stability performance predictor in latent space. Next, we show how crawl locomotion trajectories can be created in latent space when both performance predictors are used in tandem. A small optimisation program is proposed in *First Steps*, which finds a latent space trajectory constrained to firstly be stable and secondly to increase the likelihood that the robot is taking steps in the sequence of the crawl gait. Once this optimisation is solved the resulting latent space trajectory is decoded and deployed onto the real ANYmal B robot. We also compare the complexity of the formulation in *First Steps* to that of *TOWR*, which can solve similar locomotion problems. This comparison is summarised in Fig. 2 in *First Steps*. The optimisation program in *First Steps* is conceptually simpler than that of *TOWR* as many of *TOWR's* constraints are grouped and approximated by the learnt performance predictors in *First Steps*. To the best of our knowledge, *First Steps* is the first work to show locomotion trajectories for real-world locomotion, which were solved in a suitably structured latent space and deployed on a real robot.

This chapter has been published as *First Steps: Latent-Space Control with Semantic Constraints for Quadruped Locomotion* at the IROS conference in 2020.

First Steps: Latent-Space Control with Semantic Constraints for Quadruped Locomotion

Alexander L. Mitchell[1,2], Martin Engelcke[1], Oiwi Parker Jones[1], David Surovik[2], Siddhant Gangapurwala[2], Oliwier Melon[2], Ioannis Havoutis[2], and Ingmar Posner[1]

***Abstract*— Traditional approaches to quadruped control frequently employ simplified, hand-derived models. This significantly reduces the capability of the robot since its effective kinematic range is curtailed. In addition, kinodynamic constraints are often non-differentiable and difficult to implement in an optimisation approach. In this work, these challenges are addressed by framing quadruped control as optimisation in a structured latent space. A deep generative model captures a statistical representation of feasible joint configurations, whilst complex dynamic and terminal constraints are expressed via high-level, semantic indicators and represented by learned classifiers operating upon the latent space. As a consequence, complex constraints are rendered differentiable and evaluated an order of magnitude faster than analytical approaches. We validate the feasibility of locomotion trajectories optimised using our approach both in simulation and on a real-world ANYmal quadruped. Our results demonstrate that this approach is capable of generating smooth and realisable trajectories. To the best of our knowledge, this is the first time latent space control has been successfully applied to a complex, real robot platform.**

I. Introduction

Four-legged robots (quadrupeds) are capable of traversing a broader range of terrains than wheeled robots and can carry larger payloads with longer battery lives than drones ([1], [2], [3], [4]). Unlike wheeled robots, legged robots can choose to place their feet on specific locations within the terrain which can support the mass of the robot. This makes them highly suitable for inspection and monitoring tasks in unstructured domains ([2], [4], [5]). However, this flexibility comes at the cost of platform complexity. Kinematics and dynamics are considerably more complicated for quadrupeds than for wheeled robots or drones. This makes trajectory optimisation, i.e. computing feasible paths for robot end-effectors and for the robot's centre of mass (CoM) given kinematic, dynamic, and environmental constraints, one of the central challenges in their deployment ([6], [7]).

Traditionally, trajectory optimisation for quadrupeds is solved using constrained optimisation. However, the robot's feasible joint space and dynamics such as stability, torque limits, and contact forces, require complex often non-differentiable constraints. This makes the optimisation intractable. A typical approach, therefore, is to use approximate, hand-derived dynamic models and arbitrarily reduce the kinematic range of the robot [6]. These linear approximations typically over-simplify the problem and limit platform capabilities, leading to overly narrow convergence basins given feasible initial states (e.g. [6], [8], [9]). As a result, trajectories solved using linear approximations are only suitable for specific use-cases and do not easily generalise to novel situations. In contrast, approaches which account for the full kinodynamic representation of the robot require iterative, sequential optimisations to check for constraint satisfaction [10]. This often makes them computationally infeasible for real-time deployment.

[1]Applied AI Lab (A2I), [2]Dynamic Robot Systems (DRS)
Oxford Robotics Institute (ORI), University of Oxford
Correspondence to: mitch@robots.ox.ac.uk

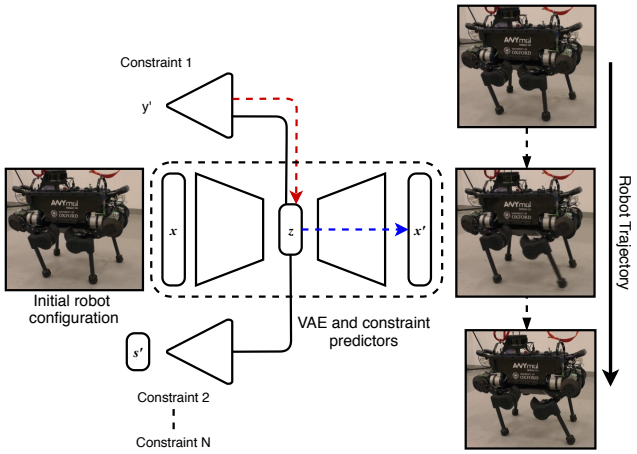


Fig. 1: A VAE encodes the robot state and captures correlations therein in a structured latent space. Once trained, performance predictors (triangles) attached to the latent space predict if constraints are satisfied and apply arbitrarily complex constraints such as robot stability. The flexibility of this approach allows for the application of any number of constraints by performing activation maximisation of a loss summed over all active constraints (see, for example, Eq. 8). Decoding positions in latent space (blue) along the optimisation trajectory translates to movement of the robot.

Here, we propose a radically different approach to quadruped control. Using a generative model of the robot state we perform trajectory optimisation by directly optimising the position in a structured latent space, which captures a statistical model of the robot's feasible joint-space (see, Fig. 1). In particular, we employ a *variational autoencoder (VAE)* ([11], [12]) to encode the robot state including joint positions. Inspired by [13], operational constraints are induced via *high-level*, semantic indicators and represented by learned performance predictors operating directly in the latent space. Arbitrarily complex constraints – and goals – can thus be enforced by performing gradient-based optimisation

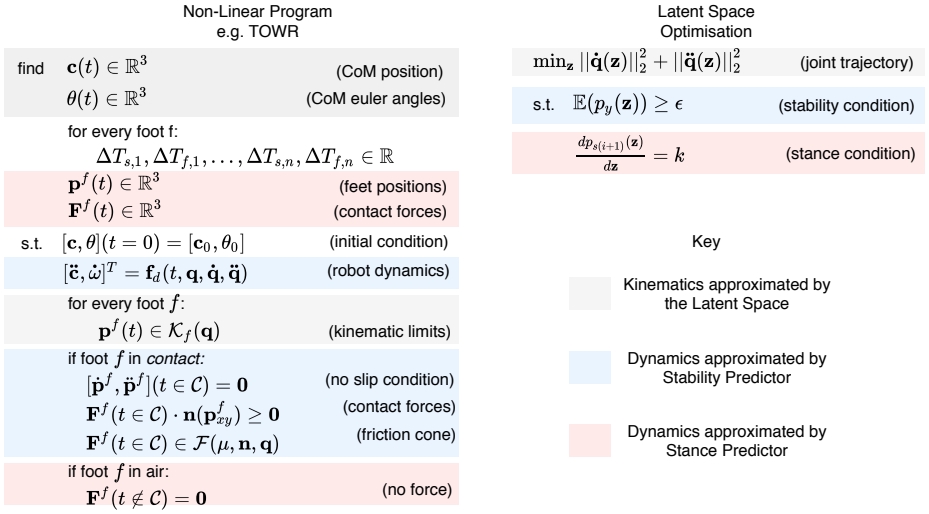


Fig. 2: Conceptual comparison of solving a non-linear program (NLP) such as TOWR with latent space optimisation. The NLP splits locomotion into an optimisation subject to a series of constraints, whilst trajectories in our approach are solved via gradient descent in a structured latent space. Complex constraints such as stability are enforced using learned classifiers (performance predictors) and gradients from differentiating a target loss are used to update a trajectory in latent space. Performance predictors replace multiple and often non-differentiable dynamics exhibited in the NLP.

in the latent space driven by the performance predictors via activation maximisation [14].

By implicitly capturing a full kinodynamic model, our approach enables *efficient* trajectory optimisation without relying on hand-specified dynamics models or linear approximations. In contrast to prior art, it predicts whether constraints are satisfied with a single pass through our neural network. More importantly, it enables direct optimisation of traditionally non-differentiable dynamics (see, Fig. 2) together with the correlations captured by the latent space, and this results in inherently smooth, feasible robot trajectories when decoding poses into state-space along the optimised path in latent space.

We validate our approach to latent-space control in the context of quadruped locomotion and constrain robot stability and end-effector contact dynamics to obtain a walking gait (Fig. 1). Using both simulated and real robot experiments, we demonstrate that our approach to latent-space control is able to find trajectories from unstable to stable robot configurations as well as to execute realisable gait cycles. This is achieved with execution times almost an order of magnitude faster than a comparable analytical approach [15].

To the best of our knowledge this is the first approach to latent-space control for a complex dynamic system which is validated both in simulation and on a real platform.

II. Related Work

As torque controlled robots have become more complex, the weaknesses of traditional trajectory optimisation approaches have become more pronounced. These mostly stem from large configuration-spaces and the complexity of model dynamics. Hand-derived dynamics models such as centroidal dynamics [16] can be used as dynamic constraints ([7], [6]). For example, *TOWR* [6] is a general approach for solving locomotion tasks for any legged robot over known terrain. Specifically, TOWR solves a non-linear program for the centre of mass and feet trajectories whilst optimising the contact forces. This is computationally slow and prohibits online updates to the current plan to react to environmental changes. Furthermore, kinematic constraints restrict the range of the robot's movement. While TOWR is suitable for highly-dynamic locomotion, our approach is conceived to be sufficiently general for low-velocity operation such as locomotion over rough terrain.

[15] estimates robot stability by finding a "margin of stability" while taking into account the robot's centre of mass, friction forces, and torque limits. We use this formulation to create training data for our stability predictor and as a performance baseline. While the original approach requires sampling and evaluating multiple points, our method directly optimises for stable trajectories using a learned stability predictor.

Reinforcement learning (RL) (e.g. [17], [18]) is a promising alternative to optimisation-based approaches. However, RL policies are difficult to train, requiring vast quantities of data, and suffer from unstable gradient estimates. Once trained, specific constraints such as foothold placements are impossible to enforce. In contrast, our approach is flexible: additional constraints are applied by constructing an appropriate loss function summed over all active constraints, which is differentiated to provide a gradient update.

Recent advances in latent-space optimisation also attempt to overcome the limitations of traditional approaches. Universal Planning Networks (UPN) [19] use a gradient-based method in a structured latent space to find trajectories between an initial and final image. Similarly, Embed to Control (E2C) [20] performs system identification and state

estimation from visual inputs for control problems. In addition, a forward dynamics model is learned to traverse the latent space given an input. Both UPN and E2C need to be provided with a goal condition. In contrast, performance predictors, presented here, are capable of predicting if arbitrarily complex constraints, such as robot stability, are satisfied. A target loss is evaluated using these performance predictors and differentiated until the latent variable is guided into the terminal set via gradient descent.

III. Gradient-Based Control in Latent Space with Performance Predictors

A. Learning a Latent Representation of Robot States

VAEs learn compact and smooth representations of high-dimensional data ([11], [12]). We utilise a VAE to encode the robot's joint angles $\mathbf{q}$, feet positions in the base frame $\mathbf{p}^f$ (where $f \in \mathbb{N}$ is the number of feet), joint torque τ, contact forces λ, and the gravity body force vector $\mathbf{g}$, forming the input to the VAE $\mathbf{x} = [\mathbf{q}, \mathbf{p}^f, \tau, \lambda, \mathbf{g}]^T$, $\mathbf{x} \in \mathbb{R}^{51}$. To find these quantities, *static*, snapshots of the robot in random, but feasible joint configurations are sampled under the condition that joint velocities and accelerations are set to zero. Finally, the VAE is trained to reconstruct the input $\mathbf{x}$ via a latent space denoted $\mathbf{z} \in \mathbb{R}^{N_z}$. Training is achieved by minimising the KL-regularised mean-squared error of the reconstruction:

$$\mathscr{L}_{VAE} = MSE(\mathbf{x}, \mathbf{x}') + \beta D_{KL}[q(\mathbf{z}|\mathbf{x})||p(\mathbf{z})] \tag{1}$$

Given the importance of robot stability in general and information about its current stance for the purpose of locomotion in particular, the VAE is trained jointly with two *performance predictors* [13] which estimate the probability of the robot being stable y' and the probability that it is currently in a specified stance s_i' using a binary cross-entropy (BCE) loss. Gradients from these two classifiers are backpropagated through the VAE encoder and aid in shaping the latent representation. Given two hyperparameters μ_1 and μ_2 for weighting the loss terms, the full training objective can thus be written as:

$$\mathscr{L} = \mathscr{L}_{VAE} + \mu_1 BCE(y, y') + \mu_2 BCE(s_i, s_i') \tag{2}$$

This provides an inductive bias to the model, which structures our latent space so that consecutive stance clusters are adjacent in the latent space (see, Fig. 3). This structure is achieved by the design of the stance labels. Stances in the walk gait where four feet are on the ground are one-hot encoded, whilst intermediary classes with a foot is in the air are two-hot encoded. Therefore, regions exist in latent space where two classifiers are active and, subsequently, consecutive stance clusters are encoded in sequence in the latent space.

B. Constrained Latent-Space Traversal

The performance predictors estimate whether non-linear and often non-differentiable kinodynamic constraints, as well as goal conditions are satisfied. Importantly, formulating these constraints as neural network classifiers renders them

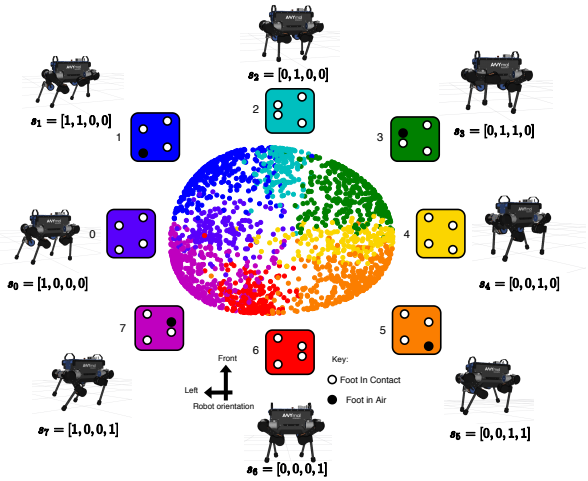


Fig. 3: The latent space (central panel) is structured and clearly shows the stance clusters (coloured). For visualisation, principle component analysis (PCA) [21] with a radial basis function (RBF) kernel reduces the latent space from 64 dimensions to two. Walking is achieved by cycling clockwise through the eight stances, which are represented by coloured diagrams orientated so that forward is up the page and left aligns with reader's left. Open white circles represent the target position of feet in contact with the ground; closed black circles represent feet in the air.

differentiable with respect to the latent variable $\mathbf{z}$. This allows us to update a latent variable sequentially via gradient descent until a constraint is satisfied. This is performed either to steer a trajectory to satisfy a terminal constraint or to enforce a condition upon the entire trajectory. Using gradient-descent to update a latent variable while a performance predictor enforces a particular constraint is a use of *activation maximisation* (AM) [14].

The general case for *activation maximisation* requires one or more differentiable loss functions $\mathscr{L}_k(y, y')$, where y is the target value and y' is the predicted value. Each loss function has an associated step size α_k, which scales the respective gradient step. Finally, an update to the latent variable $\mathbf{z}$ is made via gradient descent in the latent space:

$$\mathbf{z} \leftarrow \mathbf{z} - \nabla \sum_k \alpha_k \mathscr{L}_k(y, y') \tag{3}$$

This approach is inherently flexible as additional constraints are applied simply by adding additional loss functions prior to a gradient update.

C. Activation maximisation for stability

A specific use of constrained latent-space traversal in the context of quadruped control is disturbance rejection. Performance predictors are not only capable of predicting if the robot has become unstable, but also finding smooth and stabilising trajectories by continuously backpropagating the error from a performance predictor back into the model's latent representation $\mathbf{z}$ using activation maximisation (AM). Each latent update $\mathbf{z}$ is decoded to state space $\mathbf{x}'$ to create a trajectory. By choosing to minimise the BCE loss, the

predicted probability that the robot is stable y' is driven close to the desired probability y. Once a step size α_y is selected, AM for robot stability is defined:

$$\mathbf{z} \leftarrow \mathbf{z} - \nabla \alpha_y BCE(y, y') \tag{4}$$

The resulting trajectories are evaluated for their efficacy and feasibility in section IV.

D. Optimisation for Locomotion

Locomotion requires finding a trajectory which allows the quadruped to take a series of steps through a gait sequence while the robot remains stable. The robot is stable when the feet in contact support the CoM without slipping while respecting maximum torque limits. To solve for a walk trajectory in latent space, two performance predictors are required: one for stability and another to move the robot between gait stances (see Fig. 3).

To frame locomotion as an optimisation problem, we minimise the joint velocities $\dot{\mathbf{q}}$ and accelerations $\ddot{\mathbf{q}}$ over an entire trajectory. The joint velocities and accelerations are calculated from the joint angles using a forward-Euler derivative estimation. Intuitively, minimising these values will result in smoother, more readily executable trajectories as the actuators perform best when the input is varied smoothly [22]. Therefore, we solve for a trajectory that minimises the squared L_2-norm of the joint velocity and acceleration of the robot, whilst maximising the robot's stability y and linearly increasing the probability of being in the next stance s_{i+1}. Trajectories are of length N, which is N/f_s seconds long, where f_s is the sampling frequency. The initial robot configuration $\mathbf{x}_0$ is encoded into the latent space to find $\mathbf{z}_0$. N repeats of $\mathbf{z}_0$ are stacked to create $\mathbf{Z} \in \mathbb{R}^{N,N_z}$, and this forms the latent-space trajectory. Concretely, we perform

$$\mathbf{Z}^* = \arg\min_{\mathbf{Z}} \quad \left(||\dot{\mathbf{q}}(\mathbf{Z})||_2^2 + ||\ddot{\mathbf{q}}(\mathbf{Z})||_2^2 \right) \tag{5}$$

$$s.t. \quad \mathbb{E}(p_y(\mathbf{Z})) \geq \varepsilon \tag{6}$$

$$\frac{dp_{s(i+1)}(\mathbf{Z})}{d\mathbf{Z}} = k \tag{7}$$

The constraints in Equations 6 and 7 are converted to costs using Lagrange multipliers λ_i and added to the objective function in Equation 5 to form the locomotion loss

$$\mathscr{L}_{loco} = ||\dot{\mathbf{q}}||_2^2 + ||\ddot{\mathbf{q}}||_2^2 + \lambda_0 BCE(y, y') + \lambda_1 BCE(s, s') \tag{8}$$

$\mathbf{Z}$ is passed through the VAE's decoder to obtain the reconstructed $\mathbf{X}$, and also through the performance predictors to provide stability ($\mathbf{Y}$) and stance ($\mathbf{S}$) estimates. This output is concatenated into $\mathbf{W} = [\mathbf{X}, \mathbf{Y}, \mathbf{S}]$ and multiplied with a selector matrix to compute the quantities needed to evaluate the locomotion loss $\mathscr{L}_{loco}$. $\mathscr{L}_{loco}$ is summed over the entire trajectory and multiplied with a step size α_{loco} before being differentiated with respect to $\mathbf{Z}$ to provide a gradient update such that

$$\mathbf{Z} \leftarrow \mathbf{Z} - \nabla_{\mathbf{Z}} \alpha_{loco} \mathscr{L}_{loco}(\mathbf{Z}) \tag{9}$$

IV. Experiments

We apply constrained latent-space traversal to solve complex planning tasks on a quadruped robot. In particular, we focus on stability and locomotion. Our evaluation first investigates whether our approach to latent-space control is able to provide feasible trajectories when rapid robot stabilisation is required. This is compared to a baseline of performing activation maximisation by backpropagating gradients directly into the input of a feed-forward classifier rather than a structured latent encoding. Secondly, we add a constraint on the robot stance to enable it to take steps while remaining stable throughout the movement. We first describe the dataset of sampled robot configurations and associated ground-truth labels required for training the VAE and the performance predictors. We then provide the architectural details of our networks before presenting results of the individual experiments.

A. Dataset Generation

A dataset is sampled consisting of triplets $\{\mathbf{x}, y, s\}$, where $\mathbf{x}$ is a static robot configuration and $\{y,\ s\}$ are the associated ground truth labels for robot stability (binary) and stance (one out of eight), respectively. The robot's centre of mass location is sampled uniformly within $\pm$120mm in the longitudinal direction and $\pm$100mm in the lateral direction. Feet not in contact with the ground are sampled at heights of $\{0, 40, 80, 120\}$mm. We collect a total of 88,000 triplets whereby 70,400 are used for training (80%) and 17,600 for testing (20%).

Stability: To obtain the ground truth labels for robot stability, we follow the approach in [15], which considers friction limits of the ground and the robot as well as torque limits of the series elastic actuators [22] of the robot. The friction cone model estimates if the robot's feet in contact will slip on the surface. A foot in contact is defined such that the velocity or acceleration of the foot is zero. This means that there are different sets of contact forces and torques for a foot in contact with a surface and a foot which touches the surface without being in contact. The coefficient of friction is assumed to be constant and hence as long as the lateral forces are less than the force normal to the surface multiplied by the coefficient of friction, the robot will not slip.

Stances: Robot stances are defined by which feet are in contact with the ground and the relative ordering of the feet. Specifically, we break up a walking gait into eight phases where each phase is associated with a unique stance (see Fig. 3). The eight stances are made up of four stances with one foot in the air and four with all four feet in contact. Locomotion thus corresponds to cycling through these stances, see (Fig. 3).

B. Architecture Details

The VAE architecture comprises an encoder and decoder, which each have two fully-connected layers with 256 units and ELU non-linearities [23]. The latent space is of width 64 with a zero mean and identity variance Gaussian prior, and a diagonal Gaussian posterior distribution. The stability

classifier consists of three fully-connected layers with a width of 64 units and ELU non-linearities. It has a single output for the stability semantic label. The stance classifier consists of three fully-connected layers with a width of 64 units and ELU non-linearities. It predicts a four element output vector providing the stance encoding (see Section III-A). In the case of the locomotion experiments described below, our model takes the position of the robot's centre of gravity as an additional input. This extra input is reconstructed along the trajectory and was found useful for debugging purposes. We have verified empirically that it does not significantly impact model performance otherwise. As a baseline, a second feed-forward neural network is trained so that activation maximisation is performed directly into input space as opposed to the latent space. This classifier network takes as input the robot data $\mathbf{x}$ and predicts the probability that the robot is stable and which stance the robot is in. The architecture used consists of three fully-connected layers with a width of 64 units and ELU non-linearities.

C. Stability Experiments

The context for this experiment is disturbance rejection. If, for example, the robot is perturbed and no longer stable, we demonstrate how activation maximisation (AM) in a suitable latent space can find a smooth trajectory to a stable pose. At each time step, we backpropagate the stability error from a performance predictor back into the model's latent representation $\mathbf{z}$. This is decoded to $\mathbf{x}'$ and sent as a command to the robot's low-level controller until the robot is stable. We compare AM into a structured latent space (*latent-AM*) with applying AM directly to the input (*input-AM*). The latter scheme bypasses the latent space and, therefore, input-AM does not explicitly account for any correlations between the input state variables.

In each experiment (latent-AM and input-AM), we randomly initialise 1000 unstable configurations. AM is then driven by a binary cross entropy loss with step size $\alpha_y = 1 \times 10^{-3}$; and $\mathbf{z}$ is updated following Eq. 4 for 300 gradient steps.

Our measure of success is the proportion of episodes that feasibly transition from an unstable to a stable configuration. Here we define a stable configuration in terms of the robot's CoM together with the contact points where its feet meet the ground. Hence, it is necessary for both models to also predict which feet are in contact with the ground. Given that these contact points define the vertices of a polygon projected on the ground, the robot is stable when its CoM hovers within the support polygon. Otherwise, if the CoM projection is outside of this polygon, the robot is overextended and considered unstable. Since the robot's CoM can be inferred from its joint configurations and feet positions in the base coordinates, the CoM was not included as an input to the model.

Furthermore, we only consider an experiment a success if the robot configurations which constitute a trajectory are *kinematically* feasible. This means that there are no self-collisions or joint limit violations in all the poses which constitute a trajectory.

We report that the overall proportion of latent-AM experiments that successfully reach a stable pose is 94.6%. However, input-AM success is significantly lower: only 62.6% of input-AM experiments succeeded.

The performance of both latent-AM and input-AM is compared once 95% confidence intervals are inferred using the Wilson score interval [24]. Latent-AM confidence intervals range from 93.1% to 96.2%, whilst input-AM varies from 59.5% to 65.5%. There is no overlap between latent-AM and input-AM confidence bounds, and, hence, there is a significant and substantial difference in performance between the two schemes.

To probe the robustness of the final robot pose, we consider more restrictive definitions of stability which shrink the support polygon, to infer a *stability margin*. Conceptually, CoMs near the centre of the support polygon are more stable than CoMs near the perimeter. The stability margin is defined as the area with which the support polygon can be reduced while the robot remains stable. In practice, trajectory optimisers shrink the support polygon by ∼20% to mitigate any error in estimating the CoM position due to sensor and actuator noise [25]. Table I shows the proportion of experiments which find a stable final pose as the support polygon shrinks in increments ∼6% from its original periphery (no change in the margin) up to 31% smaller polygons. The latent-AM success-rates never drop below 92%, while input-AM performance deteriorates considerably from 63% to 52%. This clearly indicates that latent-AM is significantly more likely to find a robustly stable robot pose.

TABLE I: The percentage of experiments which terminate at a stable robot configuration for both latent-AM and input-AM. 95% confidence intervals are also presented in brackets and calculated using Wilson score interval [24]. To measure the robustness of this stable pose, the support polygon is shrunk by a margin from zero (no margin) to 31% in 6% intervals. With no margin, 95% of latent-AM trajectories reach a stable pose as opposed to only 63% of input-AM plans. The success of both schemes reduces as the margin increases, but input-AM performance degrades severely (∼ 10%). However, latent-AM success remains above 92%.

	Successful Trajectories/ %	
Margin/ %	**Latent AM**	**Input AM**
0	**94.6** (91.9, 95.3)	62.6 (59.6, 65.5)
7	**93.8** (91.2, 94.8)	59.8 (56.7, 62.8)
13	**93.2** (90.9, 94.5)	58.4 (55.3, 61.4)
19	**92.9** (90.6, 94.3)	56.4 (53.3, 59.4)
25	**92.7** (90.6, 94.3)	54.8 (51.7, 57.9)
31	**92.6** (90.5, 94.2)	52.0 (48.9, 55.1)

For a trajectory to be realisable, we emphasise that the robot trajectory must also be *dynamically* feasible. By dynamically feasible we mean that the simulation cannot move in ways that would be physically impossible for a real robot. Such simulated trajectories may end up at a stable configuration, but, because they cannot be implemented, are uninteresting from the perspective of a real robot. We consider an episode infeasible if the simulated trajectory exceeds the maximum joint velocity of 12 rad/s, as reported

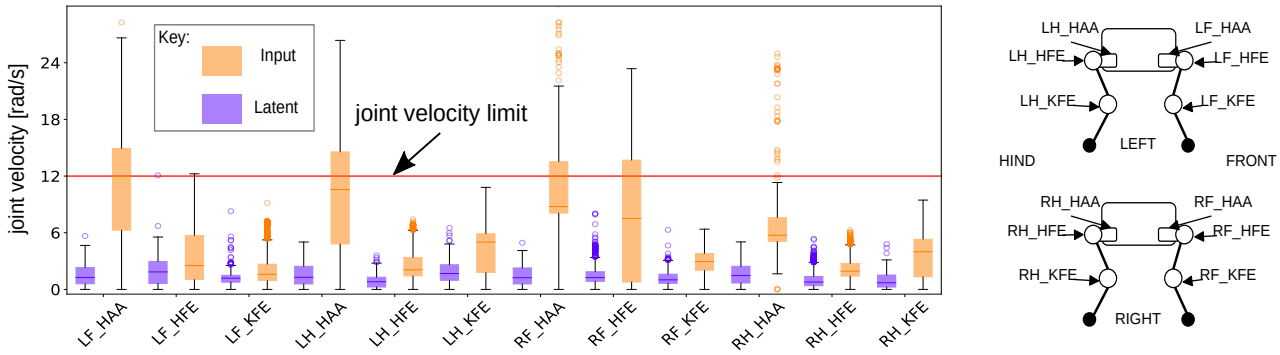


Fig. 4: A joint trajectory is only feasible if maximum permissible joint velocity (12 rad/s) is not exceeded. The proportion of input-AM trajectories which are infeasible is 59.3%, which contrasts latent-AM where no experiments violate the maximum joint velocity. The latent-AM median-maximum velocity is always lower than that of the input-AM results.

in the ANYmal motor specifications [1]. For replicability, we note that our trajectories are sampled at the target control frequency of 200 Hz.

Fig. 4 displays the maximum joint velocities from all the experiments, and input-AM joint velocity violations are visible. Furthermore, no latent-AM experiments exceed the joint velocity limit and therefore all successful trajectories are feasible. In contrast, only 59.3% of all input-AM trajectories were feasible with 95% confidence interval of (56.3%, 62.3%). Crucially, once success and feasibility are considered together, only 25.9% of input trajectories, with a 95% confidence interval of (22.9%, 28.4%) are practically useful for use on the robot. This compares to 94.6% feasibly successful latent trajectories.

Smooth trajectories are essential to avoid unsettling, tipping over, or even damaging the robot. We find that the latent-AM trajectories are significantly smoother than input-AM trajectories, as shown by the derivatives of each joint velocity in Fig. 5.

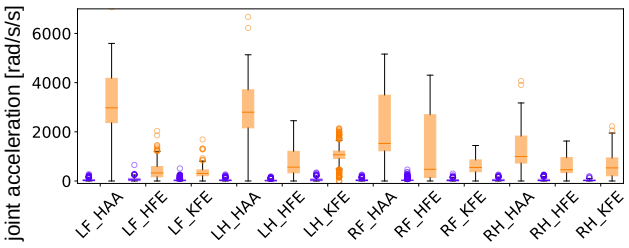


Fig. 5: The maximum joint acceleration serves as a metric of smoothness. Jerky trajectories will unsettle the robot and can cause it to fall over. Here we see that latent-AM trajectories are far smoother whilst exhibiting a narrower spread of absolute accelerations than the input-AM ones.

Analysis of the latent space show that four latent dimensions have a lower variance than the prior distribution. In contrast, the other latent units have a mean close to zero and a near unit variance, which closely map the prior distribution. Interestingly, only the four low-variance latent dimensions are updated by latent-AM, which further indicates that the latent space is exploiting the correlations which exist in the input, and that the effective latent space is four units.

Further investigation shows that the latent space is divided into stable and unstable clusters. Fig. 6 displays stable robot poses (green) surrounded by unstable configurations (red), along with a representative latent trajectory which moves from an unstable robot pose to a stable one. Below this are snapshots from a decoded trajectory sampled from a latent-AM trajectory, showing a smooth movement of the CoM to a stable position. In conclusion, the clustering of stable poses coupled with the correlations captured by the latent space could be the causes for the smoothness exhibited by the latent-AM trajectories.

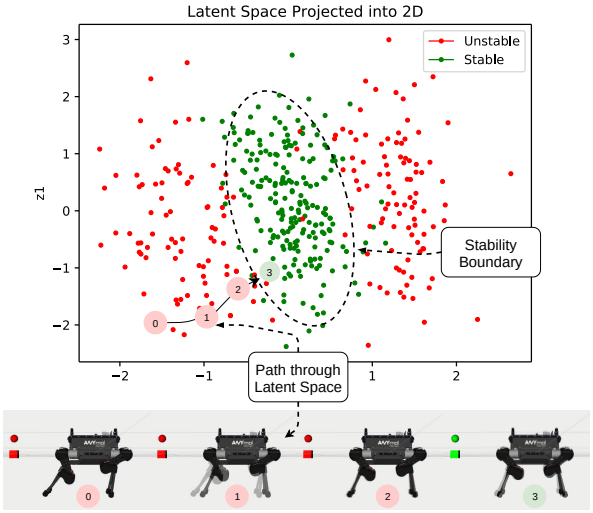


Fig. 6: We project the latent space down to 2D via PCA with an RBF kernel. Here we can see clustering of Stable robot configurations in green surrounded by unstable configurations in red. The robot configuration in pose 0 is initially unstable. Activation maximisation provides a smooth and physically realisable trajectory from this unstable pose to a stable one.

Finally, we note that evaluating a stability criterion takes 0.72ms to compute whilst a comparable analytical solution in

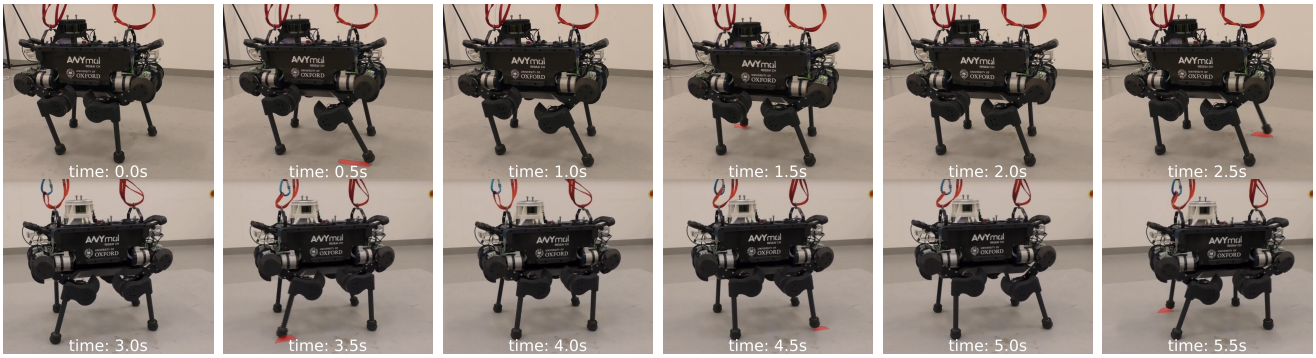


Fig. 7: The first six steps of the locomotion trajectory solved via optimisation in the latent space on the real robot. Here two separate runs with differing camera angles are displayed together. The swept area of each foot swing is displayed in red. A full video of the robot walking is found here.

[15] takes 6.56ms. This comparison uses an Nvidia Quadro P2000 on a laptop with an Intel Xeon(R) 2.9GHz CPU. This improvement in performance is attributed to the analytical solution requiring a series of sequential optimisation, as opposed to a single pass through a neural network on a GPU. Our approach, therefore, is more suitable for applications which require closed loop, online planning.

D. Locomotion

Having shown how AM is used to drive a simulated quadruped to find a smooth path that rejects unwanted perturbations, we show that an extended framework, which optimises over an entire trajectory is used to control other kinds of movement. In particular, we demonstrate that the addition of performance predictors for idealised stance configurations can be used to make a real ANYmal robot walk [3].

In our approach, the stability and stance performance predictors are sufficient to ensure that the locomotion trajectory is feasible and realisable. The overall locomotion cost is described by Eq. 8. It consists of a quadratic cost on the joint velocities and accelerations and of BCE losses. The BCE losses constantly encourage the stability of the trajectory while linearly increasing the probability of being in the next stance. For the walking demonstration, a horizon of $N = 400$ is chosen together with a step size α_{loco} of 2×10^{-4} in Eq. 9, while the Lagrange multipliers in the locomotion loss (Eq. 8) are both set to $\lambda_0 = \lambda_1 = 10^2$.

Snap-shots of a complete walk cycle executed on the real ANYmal robot are presented in Fig. 7. The controller in [7] tracked our locomotion trajectory with a tracking error of $(1.82 \times 10^{-3} rad/s)$, evaluated over a six second trajectory. Hence, this tracking error indicates that the trajectory executed on the real robot closely matches ours optimised via latent-space control. The locomotion results are qualitatively more dynamic than traditional static walk trajectories, as the centre of mass of the robot is at rest for short periods of time and only during phases where all the robot's feet are in contact. This is an encouraging result, which means that a dynamic constraint predictor could produce faster trajectories.

[3]A video of the locomotion trajectory executed on the robot is found at https://youtu.be/z1UopkfIPhI.

V. Conclusions

We propose an effective and novel approach to robot control based on high-level goal optimisation in a structured latent space, culminating in a demonstration of real-world quadruped locomotion. This is radically different from related works in that it directly exploits a statistical model of feasible robot configurations captured in a generative model to achieve smooth and realisable trajectories. These are able to stabilise a perturbed robot as well as optimise for stable and realisable locomotion — all while being computable an order of magnitude faster than equivalent analytical approaches. The optimisation is achieved via activation maximisation driven by performance predictors operating at a semantic level. This allows us to render complex, otherwise non-differentiable constraints directly usable in our approach. Our approach to latent-space control is compared with a feed-forward classifier network trained to predict stability directly from the input: the latter classifier also acting as a differentiable stability criterion. Compared to our approach, Input-AM is shown to both be significantly less successful at finding stable robot poses given an initially unstable pose and produces considerably fewer dynamically feasible trajectories than latent-AM. While we have shown the appeal and efficacy of performance predictors for applying complex non-differentiable constraints, we aim to investigate their robustness in future work. Finally, this approach and subsequent results shift the gaze from a traditional control viewpoint to a novel machine learning perspective leveraging deep generative models for complex, real-world robot control.

Acknowledgements

This research was supported by an EPSRC Programme Grant [EP/M019918/1], by the UKRI/EPSRC RAIN Hub [EP/R026084/1] and the EPSRC grant 'Robust Legged Locomotion' [EP/S002383/1]. This work was conducted as part of ANYmal Research, a community to advance legged robotics. The authors would like to thank Mathieu Geisert for his help with running the experiments on the real robot. We would also like to thank Oliver Groth for his advice throughout the project.

REFERENCES

[1] M. Hutter, C. Gehring, D. Jud, A. Lauber, C. D. Bellicoso, V. Tsounis, J. Hwangbo, K. Bodie, P. Fankhauser, M. Bloesch, R. Diethelm, S. Bachmann, A. Melzer, and M. Hoepflinger, "ANYmal - A Highly Mobile and Dynamic Quadrupedal Robot," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016.

[2] P. Fankhauser, M. Bjelonic, C. Dario Bellicoso, T. Miki, and M. Hutter, "Robust Rough-Terrain Locomotion with a Quadrupedal Robot," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2018.

[3] C. Semini, N. Tsagarakis, E. Guglielmino, M. Focchi, F. Cannella, and D. Caldwell, "Design of HyQ - A Hydraulically and Electrically Actuated Quadruped Robot," *Proceedings of the Institution of Mechanical Engineers. Part I: Journal of Systems and Control Engineering*, 2011.

[4] G. Bledt, M. J. Powell, B. Katz, J. Di Carlo, P. M. Wensing, and S. Kim, "MIT Cheetah 3: Design and Control of a Robust, Dynamic Quadruped Robot," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018.

[5] C. Gehring, P. Fankhauser, L. Isler, R. Diethelm, S. Bachmann, M. Potz, L. Gerstenberg, and M. Hutter, "ANYmal in the Field : Solving Industrial Inspection of an Offshore HVDC Platform with a Quadrupedal Robot," 2019-08-28, 12th Conference on Field and Service Robotics (FSR 2019); Conference Location: Tokyo, Japan; Conference Date: August 29-31, 2019.

[6] A. W. Winkler, C. D. Bellicoso, M. Hutter, and J. Buchli, "Gait and Trajectory Optimization for Legged Systems Through Phase-Based End-Effector Parameterization," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1560–1567, July 2018.

[7] C. D. Bellicoso, F. Jenelten, C. Gehring, and M. Hutter, "Dynamic Locomotion Through Online Nonlinear Motion Optimization for Quadrupedal Robots," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 2261–2268, July 2018.

[8] O. Melon, M. Geisert, D. Surovik, I. Havoutis, and M. Fallon, "Reliable Trajectories for Dynamic Quadrupeds using Analytical Costs and Learned Initializations," 2020.

[9] A. W. Winkler, F. Farshidian, D. Pardo, M. Neunert, and J. Buchli, "Fast Trajectory Optimization for Legged Robots Using Vertex-Based ZMP Constraints," *IEEE Robotics and Automation Letters*, vol. 2, no. 4, pp. 2201–2208, Oct 2017.

[10] T. Bretl and S. Lall, "Testing Static Equilibrium for Legged Robots," *IEEE Transactions on Robotics*, vol. 24, no. 4, pp. 794–807, Aug 2008.

[21] K. P. F.R.S., "LIII. On Lines and Planes of Closest Fit to Systems of Points in Space."

[11] D. Kingma and M. Welling, "Auto-Encoding Variational Bayes," in *International Conference on Learning Representations (ICLR)*, 2014.

[12] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic Backpropagation and Approximate Inference in Deep Generative Models," in *Proceedings of the International Conference on International Conference on Machine Learning (ICML)*, 2014.

[13] Y. Wu, S. Kasewa, O. Groth, S. Salter, L. Sun, O. Parker Jones, and I. Posner, "Imagine That! Leveraging Emergent Affordances for Tool Synthesis in Reaching Tasks," *arXiv preprint arXiv:1909.13561*, 2019.

[14] D. Erhan, Y. Bengio, A. Courville, and P. Vincent, "Visualizing Higher-Layer Features of a Deep Network," University of Montreal, Tech. Rep. 1341, Jun. 2009, also presented at the ICML 2009 Workshop on Learning Feature Hierarchies, Montréal, Canada.

[15] R. Orsolino, M. Focchi, S. Caron, G. Raiola, V. Barasuol, and C. Semini, "Feasible Region: an Actuation-Aware Extension of the Support Region," *arXiv preprint arXiv: 1903.07999*, 2019.

[16] D. Orin, A. Goswami, and S.-H. Lee, "Centroidal Dynamics of a Humanoid Robot," *Autonomous Robots*, vol. 35, 10 2013.

[17] J. Hwangbo, J. Lee, A. Dosovitskiy, D. Bellicoso, V. Tsounis, V. Koltun, and M. Hutter, "Learning Agile and Dynamic Motor Skills for Legged Robots," *CoRR*, vol. abs/1901.08652, 2019.

[18] S. Gangapurwala, A. Mitchell, and I. Havoutis, "Guided Constrained Policy Optimization for Dynamic Quadrupedal Robot Locomotion," 2020.

[19] A. Srinivas, A. Jabri, P. Abbeel, S. Levine, and C. Finn, "Universal Planning Networks," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2018.

[20] M. Watter, J. T. Springenberg, J. Boedecker, and M. A. Riedmiller, "Embed to Control: A Locally Linear Latent Dynamics Model for Control from Raw Images," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2015.

[22] G. A. Pratt and M. M. Williamson, "Series Elastic Actuators," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1995.

[23] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs)," 01 2016.

[24] E. B. Wilson, "Probable Inference, the Law of Succession, and Statistical Inference," *Journal of the American Statistical Association*, vol. 22, no. 158, pp. 209–212, 1927.

[25] M. Kalakrishnan, J. Buchli, P. Pastor, M. Mistry, and S. Schaal, "Fast, Robust Quadruped Locomotion over Challenging Terrain," in *2010 IEEE International Conference on Robotics and Automation*, 2010, pp. 2665–2670.

Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor **(only required where there isn't already a statement of contribution within the paper itself).**

Title of Paper	First Steps: Latent-Space Control with Semantic Constraints for Quadruped Locomotion
Publication Status	Published
Publication Details	Alexander L. Mitchell, Martin Engelcke, Oiwi Parker Jones, David Surovik, Siddhant Gangapurwala, Oliwier Melon, Ioannis Havoutis, Ingmar Posner Published at IROS 2020

Student Confirmation

Student Name:	Alexander Luis Mitchell		
Contribution to the Paper	Led efforts on: • Proposed work and created the team • Created and developed methodologies • Ran experiments • Created all figures Co-led efforts: • Paper writing • Experiment support • Optimising deployment code		
Signature		Date	02/OCT/2023

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title: Professor Ioannis Havoutis		
Supervisor comments These statements are correct.		
Signature Ioannis Havoutis	Date	02/OCT/2023

4

VAE-Loco: Versatile Quadruped Locomotion by Learning a Disentangled Gait Representation

This chapter presents *VAE-Loco* which creates a structured and disentangled latent representation for the dynamic trot gait. This chapter directly addresses the limitations of the previous chapter *First Steps*. The first limitation to be dispensed with is static dynamics. Indeed, the trot gait is dynamic as there are phases where only two feet are in contact with the ground. Since the robot has point feet, it is not possible to control the moment about the support line between the two feet. Hence the robot rolls freely about this line and the robot's momentum prior to and just after this phase must be carefully controlled. Therefore, any latent representation for trot locomotion planning must capture this information. *VAE-Loco's* architecture is a variational autoencoder (VAE) [19, 20] with a performance predictor to estimate the contact states of the robot's feet similarly to *First Steps*. The first difference between the *First Steps* and *VAE-Loco* is that the VAE here encodes a history of previous robot states and predicts a trajectory of future states.

The VAE and performance predictor are trained by minimising the beta-VAE loss [21] summed with the binary cross-entropy (BCE) loss from the performance predictors similarly to *First Steps*. Inspection of the learnt latent space reveals that the trot contact sequence is embedded within the space and that key gait characteristics such as footstep

length and height are encoded into two separate dimensions of the latent space. It is further found that injecting two oscillatory signals into these dimensions can control the robot's footstep length and height independently. Hence the representation of the trot gait in the latent space is disentangled with respect to the gait characteristics and amenable for locomotion planning. The latent space forms a planning manifold in a closed-loop control framework deployed firstly on the real ANYmal B platform. Once deployed, we show blind and dynamic trot locomotion and that it is possible to control the robot's cadence, and footswing heights by injecting a single *drive*-signal into the space. We show that uniquely using this method we can vary the robot's in-swing gait characteristics on-demand and independently from one another.

In this chapter, we investigate the contribution that back-propagating the BCE loss from the performance predictors has on structuring the latent space. To do this, we train two beta-VAEs [21] and performance predictor combinations: one where the gradients from the performance predictor update the encoder weights and another where they are detached at the latent space. Once both are trained to convergence, the latent space is visualised and it is found that when the gradients are detached, the latent space is no longer disentangled and the boundaries between contact states are no longer crisp meaning that during planning the contact state would not evolve smoothly. This result is found in Chapter 4, Fig. 6 which shows that the BCE loss is responsible for the disentanglement between the footswing length and height. This makes intuitive sense since the training data only contains trajectories with constant footswing length and heights. Therefore, the unmodified beta-VAE infers a correlation between footswing length and height since these are correlated in the dataset. There is only a non-zero footswing height length and height when the robot is walking and both of these heights and lengths are zero when the robot is at rest. The modification of allowing gradients from the performance predictors to update the VAE's encoder breaks this correlation since the performance predictor classifies classes given the robot's contact states. Hence, footswing height is correlated along a single dimension of the latent space. This naturally permits generalisation to locomotion with footswing characteristics broader than that seen during training. As a

result, this is an efficient training paradigm and this method has a unique capability to continuously vary the in-flight swing timings, lengths and heights.

Chapter 4 section VI compares the dynamic feasibility of the trajectories from the dataset created using a gradient-based centroidal dynamics model with those generated using the VAE. To compare the feasibility between the dataset and VAE's trajectories, the distribution of the zero-moment point (ZMP) relative to the support line is plotted during locomotion. These distributions are illustrated in Fig. 9 as box and whisker plots and the ZMP from support line distributions are similar between the dataset and the VAE. This analysis concludes that the VAE can produce trajectories which are consistent with the robot's dynamics equation and are dynamically feasible. In addition, the VAE's receptive field is plotted in Fig. 7 and elements of the input which are used to infer the robot's latent state are visible. This reveals that the contact forces are the most receptive part of the inputs attended to during inference.

The VAE is a generative model and can be used to characterise out-of-distribution events such as disturbances applied to the robot's base. As we encode robot states, the evidence lower-bound (ELBO) is calculated and if this rises above a certain threshold, it is determined that an out-of-distribution event occurred. Since we can adjust the robot's in-flight gait characteristics, we can react to disturbances and increase the stepping frequency to counteract the disturbance. We show that this improves the robustness of the VAE planner deployed on the real ANYmal B robot.

In conclusion, *VAE-Loco* shows that exposing a VAE to continuous trot gait locomotion creates a structured and disentangled latent space ideal for closed-loop locomotion planning. Furthermore, we show that the latent space is expressive enough to generalise across gait characteristics unseen during training. As a result, the VAE planner can control the in-swing gait characteristics continuously and on demand. This is a unique capability of this method. Finally, we show that the VAE can form a planner in a robust closed-loop planning framework and can detect and reject significant disturbances applied to the robot's base.

This chapter was published in IEEE Transactions on Robotics (T-RO) in 2023 as *VAE-Loco: Versatile Quadruped Locomotion by Learning a Disentangled Gait Representation*.

VAE-Loco: Versatile Quadruped Locomotion by Learning a Disentangled Gait Representation

Alexander L. Mitchell Wolfgang Merkt Mathieu Geisert Siddhant Gangapurwala
Martin Engelcke Oiwi Parker Jones Ioannis Havoutis Ingmar Posner

***Abstract*—Quadruped locomotion is rapidly maturing to a degree where robots are able to realise highly dynamic manoeuvres. However, current planners are unable to vary key gait parameters of the in-swing feet *midair*. In this work we address this limitation and show that it is pivotal in increasing controller robustness by learning a latent space capturing the key stance phases constituting a particular gait. This is achieved via a generative model trained on a single trot style, which encourages disentanglement such that application of a drive signal to a single dimension of the latent state induces holistic plans synthesising a continuous variety of trot styles. We demonstrate that specific properties of the drive signal map directly to gait parameters such as cadence, footstep height and full stance duration. Due to the nature of our approach these synthesised gaits are continuously variable online during robot operation. The use of a generative model facilitates the detection and mitigation of disturbances to provide a versatile and robust planning framework. We evaluate our approach on two versions of the real ANYmal quadruped robots and demonstrate that our method achieves a continuous blend of dynamic trot styles whilst being robust and reactive to external perturbations.**

I. Introduction

Quadruped locomotion has advanced significantly in recent years, extending their capability towards applications of significant value to industry and the public domain. Driven primarily by advances in optimisation-based [1]–[5] and reinforcement learning-based methods [6]–[8], quadrupeds are now able to robustly plan and perform dynamic manoeuvres, making them an increasingly popular and reliable choice for tasks such as inspection, monitoring, search and rescue or goods delivery in difficult. However, despite recent advances, important limitations remain. Due to the complexity of the system, models used for gait planning and control are often overly simplified and handcrafted for particular and pre-determined contact schedules [1], [9]. In the worst case, this can limit the versatility of the robot as the models deployed are failing to exploit the full capability of the underlying hardware (e.g. [1], [10], [11]). In particular, fixed gait parameters limit the ability of the robot to react to and reject external disturbances such as pushes to the robot's base. For example, the ability to adjust the robot's swing trajectories *on demand* (e.g. the swing height, length and timing) allows the robot to stabilise itself. The feet can be placed faster and into positions ahead of the centre of mass. There are works which are capable of making adjustments to a pre-determined contact schedule. The first category of method updates the contact schedule utilising simplified dynamics-models [12]. The second type solves for the contact timings in advance by adjusting the switching times between optimisation segments [13], [14] and uses reduced-order models. The last category optimises dynamics over

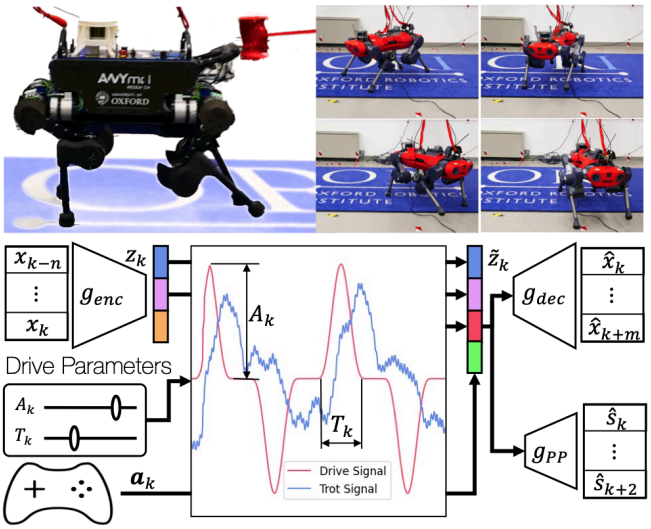


Fig. 1: Using a variational auto-encoder (VAE), our approach learns a structured latent space capturing key stance phases constituting a particular gait. The space is disentangled to a degree such that application of a *drive signal* to a single dimension of the latent variable induces gait styles which can be seamlessly interpolated between with continuous control over key gait parameters such as cadence, full-support duration, foot swing height, and footstep distance. This approach allows for precise base twist control and readily transfers from ANYmal B to ANYmal C, a dynamically dissimilar robot, without retraining. Additionally, we measure disturbances as out of distribution seen during training and adjust cadence as a rudimentary, but effective response.

This work was supported by a UKRI/EPSRC Programme Grant [EP/V000748/1], the EPSRC grant 'Robust Legged Locomotion' [EP/S002383/1], the EPSRC CDT [EP/L015897/1], the UKRI/EPSRC RAIN [EP/R026084/1] and ORCA [EP/R026173/1] Hubs and the EU H2020 Project MEMMO (780684). It was conducted as part of ANYmal Research, a community to advance legged robotics. *(Corresponding author: Alexander L. Mitchell.)*

Alexander L. Mitchell, Wolfgang Merkt, Siddhant Gangapurwala, Oiwi Parker Jones, Ioannis Havoutis, and Ingmar Posner are with the Oxford Robotics Institute, Department of Engineering Science, University of Oxford, U.K. (e-mail: mitch@robots.ox.ac.uk; wolfgang@robots.ox.ac.uk; siddhant@robots.ox.ac.uk, oiwi@robots.ox.ac.uk, ioannis@robots.ox.ac.uk, ingmar@robots.ox.ac.uk).

Mathieu Geisert is with Agility Robotics, U.S.A. Work done while at Oxford.

Martin Engelcke is with DeepMind Technologies Ltd., London, U.K. Work done while at Oxford.

gait schedules, footstep lengths and heights. These are often computationally expensive [3], [4] meaning that varying the gait parameters is not achievable in real time. A limitation of all these methods is that they are unable to adjust key gait parameters, in particular the contact timings, of the feet *midair*. Furthermore, these adjustments are limited to small perturbations around a default value. The result of which is that these methods are unable to react quickly to external perturbations irrespective of the terrain the robot is traversing. In contrast, the ability to control key gait parameters – such as contact timing, swing height, and full-support duration – *on-the-fly* would enable a smooth interpolation between dynamic manoeuvres allowing for swift reaction to external stimuli. This leads to significantly more versatile locomotion.

Inspired by recent work on a quadruped that achieves a crawl gait via the traversal of a learnt latent space [15], we approach the challenge of continuous contact schedule variation from the perspective of learning and traversing a structured latent-space. This is enabled by learning a *generative model* of locomotion data which, in addition to capturing relevant structure in the latent space, also enables the detection and mitigation of disturbances to provide a versatile and robust planning framework. In particular, we train a variational auto-encoder (VAE) [16], [17] on short sequences of state-space trajectories taken from a single gait type (trot), and predict a set of future states. We show that the resulting latent space has an interpretable structure, lending itself to the generation of a variety of trot styles, depending on how the latent space is traversed. In fact, examining trot trajectories in latent space reveals an oscillatory drive signal which controls fundamental aspects of the gait. We subsequently find that by overwriting this trajectory with a synthetic drive signal, we can *continuously* control the robot's gait properties whilst the robot is executing the motion. Parameters of this drive signal can be mapped explicitly to gait parameters such as cadence, footstep height, and full-stance support duration. With this, we can continuously vary the contact timings of feet midair. In fact we can accelerate the cadence of the swing feet from *four* steps a second at take off to *eight* at touch down within the duration of a single footstep. This constitutes a novel capability in quadruped control. We emphasise that this ability to generalise over gait styles emerges from training on a single gait type: a trot gait with constant parameters.

We illustrate the efficacy of our approach by generating a range of continuously blended trajectories firstly on the real ANYmal B quadruped robot – a medium-sized platform ($35\,\mathrm{kg}$) standing $0.5\,\mathrm{m}$ tall. Subsequently with no retraining, we repeat the experiment on the heavier ANYmal C quadruped, which weighs in at $50\,\mathrm{kg}$ and delivers twice the peak torque of the former platform. This demonstrates not only transfer from simulation to the real robot, but to a dynamically dissimilar platform crucially without retraining. While the latent space is learnt using examples only from a specific gait style, our approach is able to synthesise behaviours significantly beyond this training distribution. We choose to limit our analysis to locomotion on flat ground in order to fully explore and understand this novel control paradigm.

In addition, we leverage our generative approach to both characterise and react to external perturbations. A large impulse applied to the robot's base triggers a spike in the Evidence Lower Bound (ELBO) which clearly identifies the disturbance as out of the distribution seen during training. Inspired by [18], which states that an increase in cadence is both a response to slip and a form of push recovery in humans, our planner automatically increases the robot's cadence to aid in counteracting the disturbance. This demonstrates a marked improvement in robustness.

To the best of our knowledge, our method is the first which enables the continuous online adaptation of the robot's gait characteristics during a swing phase whilst the robot is walking. It provides a versatile and data-driven approach to quadruped locomotion which additionally allows for disturbance detection and recovery.

A. Statement of Contributions

The primary contribution of this paper and its conference version *Next Steps* [19] is a novel planning methodology, which realises locomotion with continuously variable gait parameters. Indeed, prior art is only able to alter the future timings of quadruped's contact schedule [12], [13], whilst the methods presented here facilitate the variation of timings for in-swing feet midair. Varying the foot step speed of the airborne feet in response to detected disturbance is shown to increase the robot's ability to reject large push disturbances. In this paper, we deploy the VAE without retraining on a brand new platform with significantly different dynamics: the ANYmal C.

This paper extends our previous work, *Next Steps* [19] in order to showcase in-depth analysis of the method previously proposed. This includes significant extensions to key sections of Next Steps as well as new analysis. In particular, we have extended the description and justification for the oscillatory drive-signal in Sec. III-A. We have significantly expanded the interpretation of the latent-space structure and explain how the trajectory in this space looks (Sec. VI-A). Additionally, we compare the nominal trajectory in latent space to that seen during the disturbance and recovery phase, significantly expanding on the analysis of the disturbance rejection experiments (Sec. VI-H). We also extend the discussion of the ablation study of the model's sensitivity to hyper-parameters (Sec. VI-C).

In this paper, we introduce new aspects of analysis. Firstly, we show how backpropagating the binary cross-entropy (BCE) through the VAE's encoder is essential for structuring the latent-space (Sec. VI-B). The result of which are crisp decision boundaries and axis-alignment of the latent space. This results in an interpretable and disentangled representation. Secondly, we analyse the receptive field of the VAE's encoder in order to understand how the robot's gait phase is inferred, see Sec. VI-D. Inference of the gait phase is crucial for successful closed-loop planning in latent-space. Thirdly, we compare the dynamic feasibility of the trajectories from the VAE-planner to those seen during training, see Sec. VI-F and show that the VAE-planner's locomotion trajectories remain dynamically feasible as they are significantly varied. Next, we compare the

distribution of realisable gait parameters output from the VAE-planner to those seen in the dataset, see Sec. VI-G. We show the range of gaits achievable using our approach and compare this to the distribution seen during training. Lastly, after showing that the approach can transfer from simulation to the real ANYmal B in *Next Steps*, we push this limit and deploy the VAE-planner on ANYmal C without retraining, (Sec. VI-I). This experiment demonstrates the extent to which the VAE-planner is able to generalise to out-of-distribution scenarios. In particular, ANYmal C exhibits significantly different dynamics to ANYmal B, which are detailed in Sec. V-I.

II. Related Work

Planning and control for quadruped locomotion have advanced in leaps and bounds in recent years. For example, hierarchical-optimisation frameworks split locomotion tasks into a series of smaller problems. Examples in this area include *Dynamic Gaits* (DG) [1], [12] and [20]. DG and [20] enable a quadrupedal robot (such as ANYmal) to execute a wide variety of dynamic gaits (e.g. trot, pace, lateral walk, jump) with real-time motion planning and control. However, to achieve this impressive range of behaviours, all these methods provide each gait type with its own contact schedule and utilise an environment-specific footstep planner, ultimately limiting their capabilities. The work in [21] addresses the shortcomings of hierarchical planning approaches by learning a set of heuristic operating ranges in order to increase the overall dynamic range of quadruped locomotion. This is similar in philosophy to the work we present here. However, we achieve a broad dynamic range by learning a distribution over feasible trot gaits. This distribution is then sampled via our drive signals resulting in flexible and dynamic trot locomotion. This allows us to continually adjust the foot-swing timing of the airborne feet and is achievable much faster and to a broader degree than prior art. We distinguish this ability from methods which perturb the swing-duration of feet which are yet to break contact [12]–[14]. In addition, [12] requires a heuristic metric for synchronising the contact scheduler with the current contact-state. Our approach sidesteps this requirement by performing closed-loop feedback directly in the learnt latent-space.

Latent space approaches for planning and control learn useful and typically low-dimensional representations that can be used to control complex dynamics, without relying on known system models. Classic examples include *Deep Variational Bayes Filters* (DVBF) [22] and *Embed to Control* (E2C) [23]. DVBF produces dynamically consistent trajectories by traversing continuous paths in latent space whilst E2C learns a linear system model in which control problems can be solved. *Conditional Neural Movement Primitives* (CNMP) [24] is a more recent latent space approach for robotic arms that generalises between a variety of tasks, such as pick-and-place and obstacle avoidance. Other recent works like UPN and PlaNet [25], [26] show impressive capabilities in simulation but are yet to be applied to real-world systems, including floating-base robots.

In the locomotion domain, Li et al. propose an approach in [27], which utilise a learnt latent-action model to create different locomotion trajectories. However, this latent-action space does not capture robot dynamics, and samples trajectories via a random shooting method. In contrast, *First Steps* [15] learns a structured latent space based on feasible robot *configurations* to capture the complete robot dynamics. First Steps defines a set of performance predictors that can be used in an optimisation framework to control the robot. In practice, these performance predictors can be viewed as symbolic inputs (e.g. 'left front leg up') but drive the robot in continuous space. However, because *First Steps* is trained on static snapshots of robot configurations, it does not learn from observable dynamics and thus requires more explicit structuring of the latent space than is necessary. Our previously published work *Next Steps* [19] addresses this shortcoming and significantly extends this framework to effective and robust closed-loop planning and control. In this paper, we provide significantly more in-depth study into continuous variation of the gait parameters via planning in a structured latent-space. In doing so, we further justify the utilisation of an oscillatory drive signal, analyse the quality of the VAE's trajectories, and push the limits of domain transfer through deployment on new robotic platform without retraining.

The *Motion VAE* (MVAE) [28] learns to represent dynamic trajectories in a structured latent space for the locomotion of computer-animated humanoids. This is similar to our emphasis here on learning representations for dynamic trajectories in the context of locomotion. However, moving from simulated to real physical systems, as is required for robotic applications, necessitates tackling additional complexities like latency, hard real-time requirements, and actuator dynamics. In this work, we tackle these challenges and demonstrate that a single gait style contains sufficient richness to learn a structured latent space that can be exploited to manipulate gait characteristics that generalise beyond the range seen during training. Unlike MVAE, our approach does not train on multiple gait styles, despite succeeding in producing them.

Other approaches build a model over multiple gait types and styles without utilising a structured latent-space. An example of which is [29]. This utilises a vision-based system to sample a footstep schedule and gait for the terrain ahead. These form the input to an MPC approach. Alternatively, we choose not to condition the gait parameters on vision, but allow the operator to directly choose the gait style. Additionally, our system is able to generate a broad range of trot trajectories while being trained on a single demonstration of a trot gait with fixed parameters. Hence, we propose our method as an alternative to [29] for creating a queryable model over different types of locomotion styles.

Finally, a study conducted concurrently to our own [30] yields variation between gait types (walk and trot). It utilises a reinforcement learning (RL) approach which employs a phase iterator similar to our drive signal. However, this phase iterator is enforced, whilst our gait dynamics are discovered automatically purely from exposure to trot trajectories with constant parameters.

III. Approach

Our aim is to use unsupervised learning to infer a structured latent-space which facilitates real-time and smooth variation of key gait parameters. We conjecture that structure can emerge from the exposure of a suitable generative model to a gait with predetermined and constant characteristics such as cadence, swing height, and full-support duration. Specifically, we propose that due to this structure, continuous latent trajectories result in robot locomotion, see Fig. 2. By inspecting this structure, we discover a disentangled latent-space where gait parameters are axis-aligned within this space (Sec. VI-A). Periodic trajectories in latent space can then be decoded back to smooth robot locomotion as depicted in Fig. 1. Subsequently, the VAE is deployed as a planner in a real-time control loop.

VAE Architecture: We train a VAE [16], [17] to create a structured latent-space using observed dynamic data. The input to the VAE $\mathbf{X}_k$ at time step k consists of N robot states sampled from simulated trot gaits with constant parameters (e.g. cadence and foot-step height). These state-space quantities are values we wish either to control or are required to infer the gait phase. These are joint angles; end-effector positions in the base frame; joint torques; contact forces; the base velocity; and the base pose evolution relative to a control frame, which is updated periodically. These quantities are denoted as $\mathbf{x}_k = [\mathbf{q}_k, \mathbf{ee}_k, \tau_k, \lambda_k, \dot{\mathbf{c}}_k, \Delta\mathbf{c}_k]$, where k is the time step. Note that velocities and accelerations do not form part of the VAE's input, but are inferred from the input history. This has dual benefits: first, it yields a lower-dimensional input space; and, second, it prevents sensitivity to fast-changing quantities such as the recorded joint accelerations during inference.

To deploy the VAE-planner in a closed-loop framework, we encode the input history at the control frequency f_c. However, due to restrictions on the VAE's size caused by the tight computation bounds required for real-time control, the encoder input $\mathbf{X}_k$ is constructed using states spaced at a frequency of f_{enc}:

$$\mathbf{X}_k = [\mathbf{x}_{k-r(N-1)}^\top, \ldots, \mathbf{x}_{k-r}^\top, \mathbf{x}_k^\top], \tag{1}$$

where $r = f_c/f_{\text{enc}}$ is the ratio between the control and encoder frequencies. The input $\mathbf{X}_k$ is created every time step by sampling from an input buffer which stores every robot state $\mathbf{x}$ from time step k to $k - r(N-1)$ at the control frequency. We provide full details of the setting used in our experiments in Sec. IV-B.

The VAE's decoder output $\hat{\mathbf{X}}_k^+$ predicts the current robot state $\mathbf{x}_k$ as well as M future ones sampled at a frequency of $f_{\text{dec}} = f_c$:

$$\hat{\mathbf{X}}_k^+ = [\hat{\mathbf{x}}_k^\top, \hat{\mathbf{x}}_{k+1}^\top, \ldots, \hat{\mathbf{x}}_{k+M}^\top]. \tag{2}$$

Here, $^+$ denotes future steps and $\hat{}$ denotes predictions.

As the *desired-feet-in-contact* is an input to the tracking controller, we also want to predict which of the four feet are in contact, $\mathbf{s}_k$, at the current time step, k, as well as J steps in the future. Inspired by *First Steps* [15], we therefore utilise a feet-in-contact performance predictor $g_{pp}(\mathbf{z}_k)$. This is attached to the latent space, which estimates the probability of each foot being in contact:

$$\hat{\mathbf{S}}_k = [\hat{\mathbf{s}}_k^\top, ..., \hat{\mathbf{s}}_{k+J-1}^\top]^\top \tag{3}$$

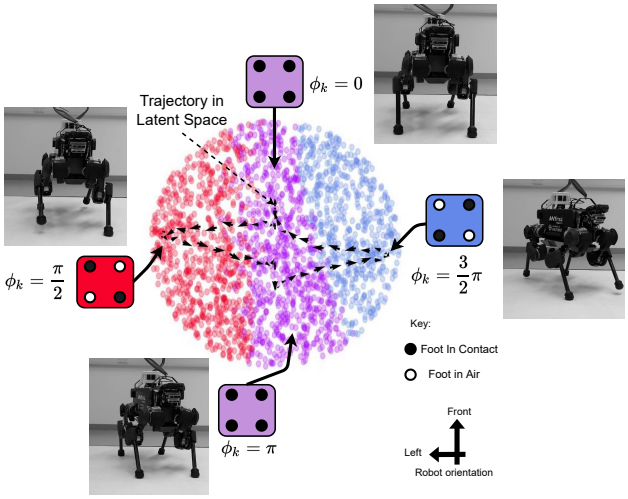


Fig. 2: A slice through the structured latent space, colour coded to illustrate the ordering and clustering of the distinct stances which make up the trot gait. The component of the latent-space trajectory (black) along the horizontal axis contributes to the robot's footstep height, whilst the vertical component gives rise to the footstep length. Snapshots of the robot controlled using the VAE-planner illustrates the inter-play between these two latent dimensions.

To command the base twist of the robot, a high-level action command $\mathbf{a}_k$ is utilised. This represents longitudinal (x), lateral (y), and yaw (θ) twist in the robot's base frame. The latent state $\mathbf{z}_k$ and the action $\mathbf{a}_k$ form the input to the decoder.

Training the VAE: We train the VAE and performance predictor together. The VAE's training loss is the modified ELBO formulation found in [31]. This loss consists of a reconstruction loss (mean-squared error) plus the Kullback–Leibler (KL) divergence D_{KL} between the inferred posterior $q(\mathbf{z}|\mathbf{X}_k)$ and the prior $p(\mathbf{z})$, multiplied by a hyper-parameter β:

$$\mathcal{L}_{\text{ELBO}} = \text{MSE}(\mathbf{X}_k^+, \hat{\mathbf{X}}_k^+) + \beta D_{\text{KL}}[q(\mathbf{z}|\mathbf{X}_k)||p(\mathbf{z})]. \tag{4}$$

These ELBO terms are then summed with the binary cross-entropy loss between the predicted feet in contact and the recorded ones. The latter term is scaled by γ, resulting in the overall loss

$$\mathcal{L} = \mathcal{L}_{\text{ELBO}} + \gamma \text{BCE}(\mathbf{S}_k, \hat{\mathbf{S}}_k) \tag{5}$$

The VAE training loss (Eq. 4), as seen in prior work [31], is responsible for any subsequent disentanglement found in the latent space. The reconstruction error is weighed against the decomposition of the latent space using the hyper-parameter β. This constraint encourages an efficient latent representation, containing only the required information for reconstruction, hence acting to regularise the latent space. As shown in [31], the D_{KL} term used with an isotropic unit Gaussian ($p(\mathbf{z}) = \mathcal{N}(\mathbf{0}, \mathbf{I})$) encourages conditional independence within $\mathbf{z}$.

In our approach, as well as that of *First Steps* [15], a structured latent space is encouraged by backpropagating

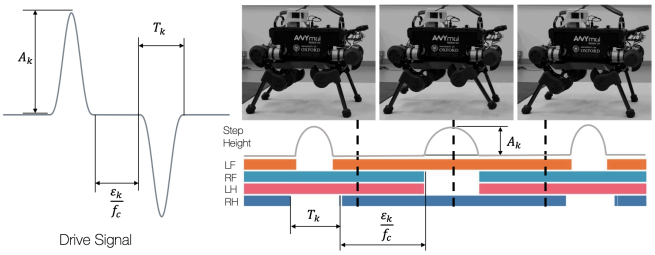


Fig. 3: An oscillatory *drive signal* overwrites the dimension with the smallest variance in the structured latent-space. The amplitude A_k, time-period T_k and stance duration counter ϵ_k of this signal control the robot's foot swing height, cadence and full-support duration in real time.

gradients from the performance predictor's loss through to the encoder input. However, we hypothesise that useful structuring of this space is inferred from the continuous trajectories used for training, and, in contrast to *First Steps*, no explicit labelling for each stance is required.

A. Control over the gait parameters

Once the VAE has been trained as described above, we find that the learnt latent-space is disentangled. We discover that an oscillation injected into one latent dimension decodes to the robot taking steps. Further analysis detailed in Sec. VI-A finds that adding a second oscillation into the latent space varies the robot's step length. The former oscillation we denote as the *drive signal*, and the latter as the *trot signal*. We find that in order to control the robot, we only need to inject the drive signal into latent space, and can infer the trot signal. By modulating and visualising the joint-space output from the decoder, we discover that varying the amplitude and phase of the drive signal leads to continuously variable trot locomotion. Therefore, we choose a specific drive signal with features which map directly to the gait parameters we wish to control. These parameters are the robot's cadence, stance duration and step height.

While any periodic oscillation (e.g. sin) decodes to robot locomotion, we choose a drive signal featuring specific components allowing for control over the robot's cadence, swing duration and step height. The drive signal chosen here is a modified $\sin^3$ oscillation with amplitude A_k, and phase ϕ_k:

$$\mathbf{z}_{k,d_z} = A_k \sin^3(\phi_k). \tag{6}$$

The amplitude A_k controls the foot swing height, and the phase ϕ_k governs cadence and support duration.

To control the robot's swing and stance duration separately, we set the drive signal's time-period T_k and we employ a stance counter ϵ_k. The time-period T_k is equal to the swing duration, whilst the time that the drive signal is equal to zero is extended by ϵ_k time steps to introduce a full-support duration of (ϵ_k / f_c) s. Hence, once both T_k and ϵ_k are used together, the phase dynamics are:

$$\phi_{k+1} = \begin{cases} \phi_k & \text{if } \phi_k \bmod \pi = 0 \text{ and } k_\epsilon < \epsilon_k \\ \phi_k + 2\pi/T_k & \text{otherwise} \end{cases} \tag{7}$$

and, in tandem, the counter ϵ_k is updated as:

$$k_\epsilon \leftarrow \begin{cases} k_\epsilon + 1 & \text{if } \phi_k \bmod \pi = 0 \text{ and } k_\epsilon < \epsilon_k \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

A $\sin^3(\cdot)$ drive signal is chosen as it is smooth over its domain meaning that the decoded trajectories will also be smooth. Additionally, a $\sin^3(\cdot)$ drive signal's gradient is zero at $\phi_k = K\pi$ for $K \in \mathbb{Z}$. Therefore, there is a continuous transition between the $\sin^3(\cdot)$ part of the drive signal and the points where the drive signal is held artificially zero for ϵ_k control-ticks. The resulting signal is shown in Fig. 3 along with the decoded contact schedule and swing trajectory.

B. Planning for closed-loop control

Once the VAE is trained, it is fast enough to act as a planner in a closed-loop controller. Thus, our approach can react to external disturbances and mitigate against real-world effects such as unmodelled dynamics and hardware latency. For closed-loop control, we begin by encoding a history of robot states from the raw sensor measurements to infer the current gait phase. We store a buffer of past robot states and sample from this at f_{enc} to create the encoder's input.

With an estimate of the current latent variable, we overwrite latent dimension d_z with the drive signal (see Sec. III-A). Next, we employ a second-order Butterworth filter to smooth the latent trajectory and further smooth the locomotion plan. In essence, the drive signal encourages the decoder to output the next open-loop prediction while the other latent variables infer the gait phase from the raw sensor input. This process is repeated at the control frequency $(400\,\text{Hz})$.

The latent variable $\mathbf{z}_k$ and a desired base twist $\mathbf{a}_k$ are decoded to produce $\hat{\mathbf{X}}_k^+ = g_{\text{dec}}(\mathbf{z}_k, \mathbf{a}_k)$. From this, the joint-space trajectory $\hat{\mathbf{Q}}_k$, and local base velocity $\hat{\dot{\mathbf{C}}}_k$ are extracted and derived or integrated to produce the base and joint positions, velocities and accelerations. These quantities and the predicted contact schedule are sent to the whole-body controller (WBC) [32]. The WBC solves a hierarchical optimisation problem to calculate the joint torques which are commands sent to the actuators. The series of constraints enforced by the WBC are: contact creation, friction constraints and torque limits. Next, the WBC applies forward kinematics to the VAE's trajectory to track it in task-space. Note that the WBC does not compensate for infeasible plans, i.e. the VAE's trajectories shown in Sec. VI-E are dynamically consistent otherwise the robot fails to walk.

C. Disturbance detection and response

Our approach is able to both detect and react to disturbances. The VAE is trained using canonical feasible trajectories. Therefore, any disturbances are characterised as out of distribution with respect to the training set. Given the generative nature of our approach, this discrepancy is quantified during operation by the trained model via the Evidence Lower-Bound (ELBO, Eq. 4 where β is set to one). We will show in the evaluation (Sec. VI-H) that even a rudimentary response strategy serves to increase the range of disturbance the system can reject.

IV. Implementation Details

In order to deploy the VAE-planner on-board the ANYmal quadruped, there are a number of real-world constraints which affect the VAE-planner. The first of which is that in order to deploy the VAE in a real-time control loop, there is a constraint on the VAE's inference time. Secondly, we discuss how we address the simulation to reality gap. We discuss the VAE's specific architecture, the hyper-parameters used to train the model, and finally, specific the specific criteria required for domain transfer.

A. Dataset generation

To train the VAE and create the structured latent space, we require a set of continuous trot trajectories. We restrict these trajectories to quadruped locomotion over flat ground *only* so that we can study and understand how to generate versatile locomotion utilising a structured latent-space. The training trajectories are generated using *Dynamic Gaits* (DG) [1]. DG is a hierarchical planning and control framework which is used with a *fixed* contact schedule and predefined footstep heights. The swing, full-stance durations and footstep heights set to $0.5\,\mathrm{s}$, $75\,\mathrm{ms}$ and $0.10\,\mathrm{m}$, respectively.

DG is made up of a footstep planner, a base motion planner and a whole-body controller (WBC). The footstep planner computes the next four steps over the gait period using an inverted-pendulum model. The base motion planner solves for the base trajectory over the gait period using a centroidal dynamics model [11]. The latter is constrained with a Zero-Moment Point (ZMP) [10] criterion, the footstep positions and schedule from the footstep planner. The WBC [32] outlined in Sec. III-B converts the task space trajectories to joint feedforward torques, reference positions and velocities: these are sent to the actuators.

The dataset is generated by uniformly sampling desired base twist and executing DG in the *RaiSim* physics simulator [33]. The fidelity of the simulation is improved by modelling the dynamics of the Series-Elastic Actuators (SEA) [34] in the ANYmal's joints using an *actuator network* [6]. We specifically utilise the network found in [8]. This network takes into account the commanded positions, velocities, feed-forward torques, and low-level PD gains The actuator network is essential for good performance as the input response of SEAs depends on a history of states, inputs, and the low-level control law.

B. VAE architecture details

Fig. 1 outlines the approach's architecture, and here we describe the VAE's specific details. Prior to the ablation study in Sec. VI-C, the VAE's encoder, decoder and stance performance predictor have two hidden layers and widths of 256 units, using ELU non-linearities [35]. The encoder input is created using $N = 80$ robot states sampled at $200\,\mathrm{Hz}$ – representing a history of $0.4\,\mathrm{s}$ – from the encoder input, which is of size 5120 units. The input is compressed via a latent space of 125 units which is concatenated with an action of 3 units. Next, the decoder outputs the current state and the next $M = 19$ robot states at the control frequency of $400\,\mathrm{Hz}$ (preview horizon $47.5\,\mathrm{ms}$, output size: 1216 units). The performance predictor predicts the current feet in contact and two future states. Finally, hyper-parameters used for training are $\beta = 1.0$, $\gamma = 0.5$, with a learning rate of 1×10^{-3} using the Adam optimiser. Training is terminated after 1×10^6 gradient steps.

C. Domain transfer

To achieve domain transfer and deployment on the real robot, some modifications are required. The contact forces are artificially set to zero when the robot measures no contact (as determined by a probabilistic contact estimator) [36]. This is necessary since the real-world ANYmal robot measures large contact forces even during swing motion, as these forces are inferred from torque residuals and are affected by model error. In contrast, simulators estimate no contact force during swing. As mentioned in Sec. III-B, a Butterworth filter with a cutoff frequency of $10\,\mathrm{Hz}$ is employed to smooth the latent trajectory. Due to the strict computation budget for real-time control, inference times for the VAE are restricted to at most $1\,\mathrm{ms}$, which imposes constraints on the capacity of the model (see Sec. VI-C for an ablation of model hyperparameters). Our largest model takes approximately $1\,\mathrm{ms}$ for the VAE computation, which is roughly equal to the computation time of the WBC.

V. Experimental Design

In this section, we explain how the latent-space properties are discovered and how we assess the performance of the VAE-planner deployed on the real robot. In doing so, we motivate the following guiding questions. The aim of these questions is to analyse the latent space once the VAE is trained and to analyse the capabilities of the VAE as a flexible and robust locomotion-planner.

We investigate (i) the structure induced in the latent space (Sec. VI-A), (ii) the effect backpropagating the binary cross-entropy gradients through the encoder has on the latent-space structure (Sec. VI-B) (iii) the sensitivity of our approach to variations in key hyper parameters (Sec. VI-C), (iv) which parts of the encoder's input are used to infer the robot's gait phase (Sec. VI-D), (v) to what extent the locomotion parameters can be varied online (Sec. VI-E), (vi) the feasibility of the locomotion plans produced (Sec. VI-F), (vii) a comparison between the gait parameters seen during training and those shown in experiments using the VAE-planner (Sec. VI-G), (viii) the degree to which disturbance detection, coupled with a rudimentary recovery strategy, further increases the robustness of our approach (Sec. VI-H), and finally, (ix) whether the VAE-planner can be deployed successfully on the next generation ANYmal C robot without retraining (Sec. VI-I). Please see the following video for an extended set of experiments along with a brief description of our approach (https://youtu.be/GT2WLh2Ackc).

A. Investigating the latent space

We wish to examine if there is any structuring in the latent space as well as investigate if any locomotion properties are disentangled within. This knowledge is crucial in understanding how to solve for locomotion trajectories in latent space.

Latent Space Structure: Structure in latent space manifests itself as clustered latent variables. During training, we expect that points in latent space which are of the same gait type will become gathered together. To verify this, we sample a set of random latent variables which we pass through the stance performance predictor. These points are plotted and colour-coded based upon their predicted stance. The resulting plot can be found in Sec. VI-A, specifically in Fig. 5.

Latent Space Disentanglement: We wish to see what trajectories in latent space look like, and if any of the dimensions within are interpretable. State-space trajectories of the robot trotting from the test set are encoded into latent-space, and the subsequent latent-space paths are visualised. These paths in latent space are oscillatory and each oscillation has its own phase.

In order to understand what each oscillation encodes, we artificially inject sine waves into each latent dimension in turn. Decoding these trajectories and visualising the joint-space paths reveals that the latent space is indeed disentangled. See Sec. VI-A for full details. This revelation informs how we solve for locomotion paths in latent space.

B. Backpropagating the binary cross-entropy through the encoder

As in *First Steps* [15], we encourage the latent space to become structured by backpropagating the BCE loss for the contact state of the feet through the encoder. We train an alternative VAE where we detach the BCE gradients at the latent space. This means that the latent-space structure is not informed by the prediction loss of the feet in contact performance predictor. The resulting latent-space of the VAE with detached gradients is compared to the original. Firstly, we look for the crispness of the decision boundaries between stance clusters. Secondly, we look to see if the decision boundaries are aligned with any dimensions in latent space. This analysis is aimed to investigate the importance of using the contact performance predictor to generate feasible locomotion.

C. Analysing the encoder's receptive field

In our experiments on the real robot, the VAE is able to infer the robot's gait phase from raw sensor inputs. We wish to understand which parts of the encoder's input are utilised to infer this. We utilise activation maximisation (AM) [37] to measure the flow of gradients through the encoder. This procedure requires backpropagating from some randomly sampled input encoding until the encoder output matches a predetermined target value. The target is chosen to match key points along the robot's gait phase and are chosen such that the drive signal's phase is equal to $0.0, \pi/2, \pi/4, \pi/8, \pi$. The gradients at the encoder input resulting from this optimisation are recorded. The magnitude of these values reveal which parts of the input are required to infer the gait phase.

D. Studying the VAE's sensitivity to key hyper-parameters

The VAE's architecture is ablated and the VAE-planner is tested in simulation until the gait phase can no longer be inferred. This leads to noisy and jerky trajectories in simulation. Firstly, the size of the latent dimension is reduced from 125 to a minimum of 6 in strides of 32 units. Secondly, the number of input states N is reduced from 80 units to 60, sampled at $200\,\mathrm{Hz}$, meaning that the encoded input history reduces from $0.4\,\mathrm{s}$ to $0.3\,\mathrm{s}$. Thirdly, the encoder frequency is halved to $100\,\mathrm{Hz}$ with $N = 80$. Lastly, the widths of the encoder, decoder and performance predictor are reduced in increments of 32 units from 256 to a minimum of 64.

E. Varying the gait parameters online

We wish to vary the robot's gait parameters continuously whilst the robot is walking. This is straight-forward for the robot to do. We utilise a simple ROS publisher with which sets the drive signal's amplitude, time-period, and stance duration. The robot's base twist is commanded independently using the action $\mathbf{a}_k$. In fact, we utilise a navigation waypoint following controller which produces the action so that the robot walks around a square trajectory in our lab.

F. Analysing the dynamic feasibility of the VAE's trajectories

The previous guiding question asks how broad the set of manoeuvres from the VAE-planner are. We now compare the dynamic feasibility of these trajectories to plans from DG. Please note that this dynamic feasibility comparison is undertaken as a post-experiment analysis. On the robot, the output of the VAE-planner is sent directly to the WBC unaltered.

To evaluate the dynamic feasibility of trajectories synthesised by our VAE, we measure the distance of the Zero Moment Point (ZMP) [10] to the support line (i.e. when one pair of legs are swinging) and then compare this distance with that in the synthetic dataset. The ZMP is a commonly used criterion in model-based legged robot control [1], [10]. This analysis is performed for the trajectories in the dataset as a baseline to which we can compare. Dynamically stable trajectories have the ZMP as close as possible to the support line.

G. Measuring the distribution of gait parameters achievable with the VAE

The gait parameters set during the robot experiments are recorded meaning that we can measure the distribution of visited states. To show the entire range of movements and compare against those seen during training, we plot a box and whisker plot of the stance and swing durations. This shows the ability of the VAE-planner to generalise producing gait parameters not seen in the training distribution.

H. Detecting and reacting to disturbances

We wish to analyse the degree to which disturbances can be detected and whether increasing the robot's cadence helps

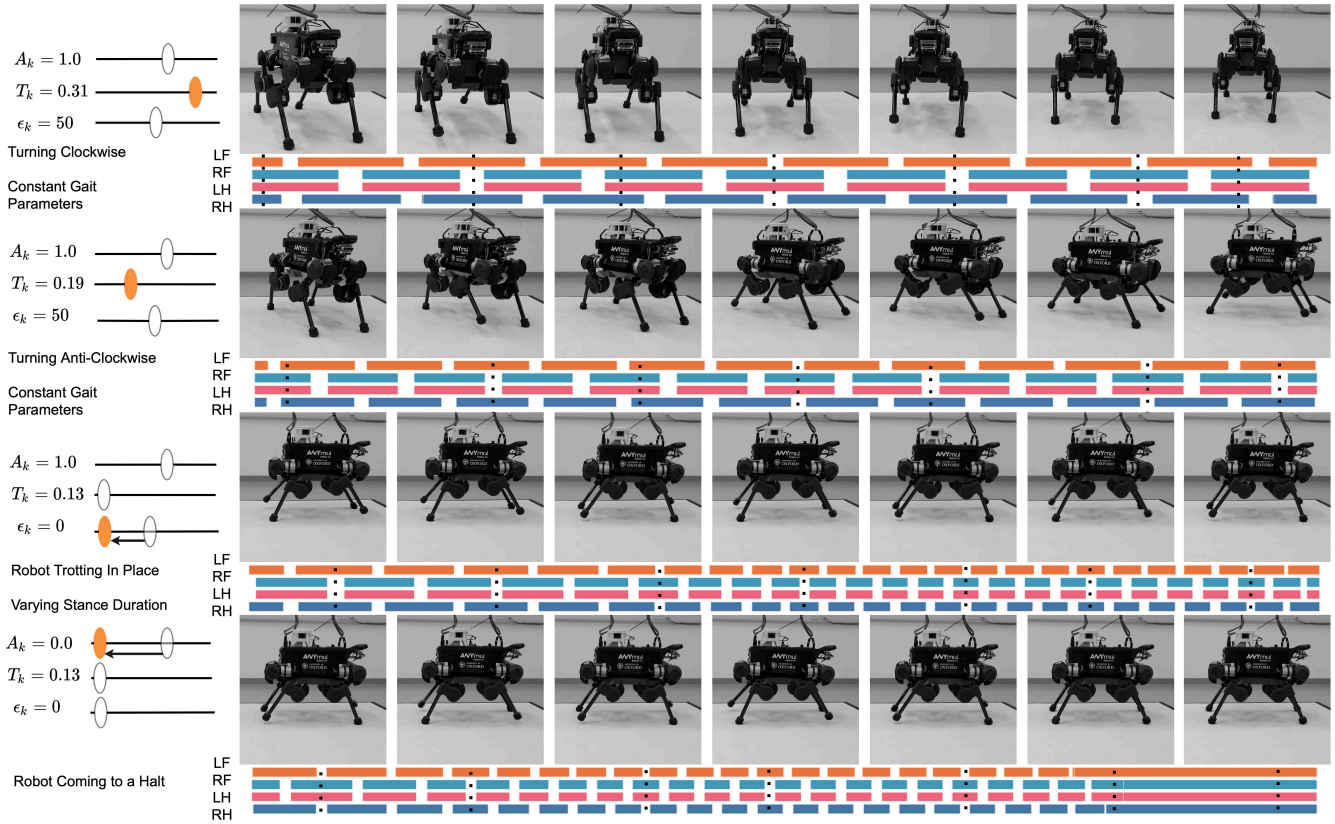


Fig. 4: Closed-loop control of the real ANYmal quadruped using our VAE-planner. This demonstrates user-controlled variation of gait parameters on the fly. Here, coloured rectangles represent the full-stance phase, whilst white space denotes the swing duration. The top row shows a trot gait with an introduced quadrupedal stance phase (gait cycle of $0.75\,\mathrm{s}$ – swing $312.5\,\mathrm{ms}$, stance $62.5\,\mathrm{ms}$; a gait cycle consists of a swing phase for each of the leg pairs). Next, the swing duration T_k is reduced using the time-period slider (gait cycle of $0.5\,\mathrm{s}$ – swing $188\,\mathrm{ms}$, stance $62.5\,\mathrm{ms}$). The third row illustrates the effect of reducing the stance duration counter ϵ_k to produce trot with reducing full-stance phases (gait cycle of $250\,\mathrm{ms}$ – swing $125\,\mathrm{ms}$, stance $0.0\,\mathrm{s}$). Finally, transition into standing occurs when the drive signal amplitude A_k is reduced to zero. To view the full range of movements, please see the following video https://youtu.be/GT2WLh2Ackc.

the robot to recover from them. As mentioned, we monitor the ELBO as this is a measure of the evidence for an encoded input relative to the learnt distribution. Hence, a large spike in this value (given the formulation in Eq. 4) is a consequence of a disturbance.

To perturb the robot, we utilise a push broom and disturb the robot's base. If the ELBO value surpasses a pre-determined constant, we characterise the event as a disturbance. The ELBO value is calibrated such during normal operation, the ELBO remains below this "disturbance" threshold. This is easily found by walking the robot using the VAE-planner for $2\,\mathrm{min}$, whilst recording the ELBO.

Following disturbance detection, the VAE-planner automatically reduces the drive signal's time-period to increase the robot's cadence. This response is inspired by human locomotion in response to slippage [18]: a group of participants encounter a slippery surface and their reaction is recorded, resulting in an increase in cadence.

We evaluate the effectiveness of increasing the robot's cadence by repeating the push experiment with and without the cadence increase. The base velocity is utilised to measure the size of the disturbance. Larger velocities arise from bigger pushes. We make a comparison between the reactive VAE-planner and a constant cadence version.

In addition, the latent-space trajectory is inspected during and after a disturbance. We are interested in how this is affected by out of distribution occurrences like a shove from a push broom or a kick to the robot's base.

I. Deployment on ANYmal C without retraining

The VAE which is trained using simulated ANYmal B locomotion data is deployed on ANYmal C without retraining. Since our generative model has successfully transferred from simulation to the real robot, a key question is can the VAE-planner work on a different robot with similar kinematics, but significantly different dynamics? Therefore, we deploy the VAE-planner on a different robot which has a few key differences. Firstly, the ANYmal C has around twice the torque limit of ANYmal B, $80\,\mathrm{N\,m}$ and $40\,\mathrm{N\,m}$ respectively. Secondly, ANYmal C is significantly heavier than ANYmal B weighing in at $50\,\mathrm{kg}$ to B's $35\,\mathrm{kg}$. Finally, ANYmal C's actuators have a lower bandwidth due to the additional reduction gearing.

The only alterations made to the VAE-planner are firstly the torques are standardised to reflect the difference in peak torque demand, and secondly, the WBC's internal dynamics' model is updated to ANYmal C.

As previously discussed, the VAE-planner can be utilised to measure differences between the learnt distribution and the encoded one. Therefore, this approach is able to characterise differences between the two robots. Again, we utilise the ELBO for this, and we record this quantity when the VAE-planner is deployed on ANYmal C. The ELBO distribution can be separated in the KL divergence and the reconstruction error. The former provides insight into the distribution of encoded values and the latter acts as a prediction loss.

To verify if the differences in KL-divergence collected from both robots are statistically significant, we utilise a Mann-Whitney U-test [38]. Our null hypothesis is that the median of the distribution from ANYmal B is equal to the median from the distribution from ANYmal C. The values of KL-divergence are collected from deploying the same VAE-planner on both robots. In total, we analyse 2.5×10^4 values from each robot, which is equivalent to $62.5\,\mathrm{s}$ at $400\,\mathrm{Hz}$. The null-hypothesis is rejected if the resulting p-value is less than our chosen statistical significance value of $0.1\,\%$. This procedure is repeated for the two sets of reconstruction error gathered from each robot.

VI. Experimental Results

Now that we have posed our guiding questions and stated how to answer them, we present the results. These questions can be split up into two categories: (a) introspection of the VAE and (b) analysis of the VAE as a planner deployed on two real-world platforms. Please see the following video for a complete set of experiments along with a brief description of our approach: https://youtu.be/GT2WLh2Ackc.

A. Structure induced in the latent space

The latent space is inspected to discover what structure exists and if any locomotion properties are disentangled within.

Latent Space Structure: Fig. 5 shows samples from the latent space colour-coded by their *predicted* stance. For trot, there are four stances: the full-support phase, left front and right hind in contact, another support phase and finally right front plus left hind in contact. Fig. 5 reveals that the latent space has emerged clustered by stance and that, due to the ordering of these stances, a periodic trajectory decodes to a trot gait. This favourable structure is inferred from the continuous trot trajectory input during training.

The ordering of the clusters matches the sequence of the trot gait. The reason for this is that the encoder's input contains information about the evolution of the robot's state over time. This additional information coupled with the training paradigm means that continuous trajectories in latent space match with continuous trajectories in state space. This is in contrast to the work in *First Steps* [15]. *First Steps* generates a crawl gait via gradient descent in a structured latent-space. However, the input to that model is a static snapshot of the robot's state with no temporal information (e.g. $\mathbf{x}_k$). Since there is no information about the evolution of the robot's state in the encoder's input, additional structuring was required to order the clusters in the crawl gait sequence.

Latent Space Disentanglement: By examining the latent variables, we discover that oscillations injected into just two dimensions in the latent space decode to continuously varying trot trajectories. This result stems from a latent space where variation in footstep length and cadence is aligned along one dimension, while variation in footstep distance lies along another. Specifically, the time period of what we denote the *drive signal* oscillation controls the robot's cadence, whilst its amplitude is proportionate to the footstep height. In addition, the amplitude of the second signal, which is $\pi/2$ out of phase with the drive signal, controls foot swing length and as such is denoted as the *trot signal*. Given that other work has tried to explicitly build this structure into locomotion systems [30], it is important to emphasise that this disentanglement *emerges* in our study as a result of the training paradigm and data. Only the synthetic drive-signal needs to be injected into the latent space for closed-loop control; the trot signal is inferred.

Visualising The Latent-Space Trajectory: We plot the injected drive-signal and inferred trot signal in Fig. 5 in red and blue respectively. When these two signals are plotted against one another, they combine to form a cycle in latent space. This is plotted as the black-vector field in Fig. 5. This cycle is annotated to show how the latent-space trajectory visits each stance cluster in turn. We also show the corresponding contact schedule with images of the robot in their matching configurations.

To begin at point (1), the robot has four feet in contact, before moving anti-clockwise tracing out a triangular lobe in the red region to point (2). This lobe forms the first foot-swing. The trajectory continues anti-clockwise through the magenta region (3) before tracing another lobe through the blue area back to point (1). This cycle repeats as the robot takes more steps.

B. Backpropagating the BCE loss through the encoder

We compare the latent space created by backpropagating the BCE gradients through the encoder to the vanilla case, where these gradients are detached before the latent space in Fig. 6. The latent space structured using the BCE gradients exhibits crisp decision boundaries for the stance clustering. These boundaries are aligned perpendicular to the drive-signal dimension in latent space, meaning that stance classification results mostly from this dimension. In contrast, the detached gradient latent-space has fuzzy decision boundaries. This is less than ideal as the predicted contact state is an input to the controller, and vitally affects the propagation of the dynamics model. If this is a noisy prediction, the robot will change contact state in an unpredictable way. Note that the decision boundaries in the detached case are rotated at roughly 30° in comparison the nominal case. This results in the latent space being no longer axis aligned with the drive signal. In order to create locomotion trajectories and have independent control over the gait parameters, namely step height and length, the drive signal needs to be rotated to align with

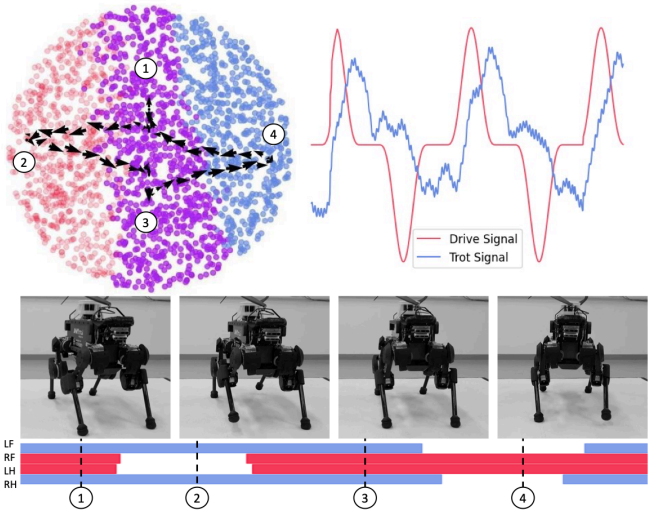


Fig. 5: The latent space and trajectory in black during unperturbed operation. The latent space trajectory is result of the combination of the red drive-signal along the horizontal dimension and the blue trot-signal in the vertical. The drive signal is user controlled, whilst the trot-signal is inferred. The subsequent robot locomotion and contact schedule is included.

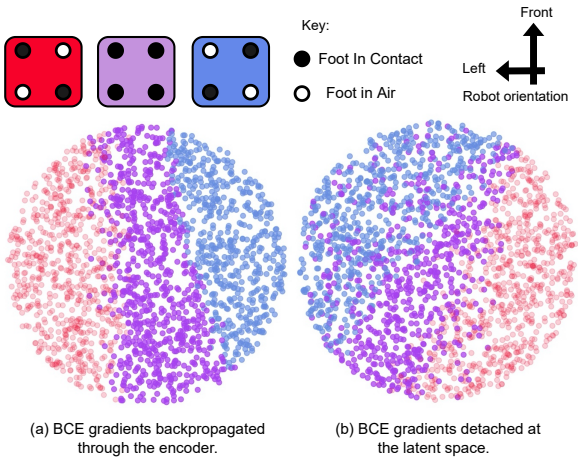


Fig. 6: A comparison of a latent space actively structured using BCE gradients which flow through the encoder's input (sub-figure (a)) to the vanilla case, where BCE gradients are detached at the latent space (sub-figure (b)). Points in latent space are coloured by their stance. In the key above the latent-space images, a filled-in circle represents a closed contact and a circular outline denotes an open contact.

the decision boundaries. This is not required for the nominal case, where the BCE gradients are backpropagated through the encoder. Therefore, the crisp decision boundaries and the axis-alignment justifies our decision – and more crucially highlights the importance – of using the BCE gradients to structure the latent space.

C. Sensitivity To Hyper Parameters

As mentioned in Sec. IV-C, real-time performance is only achievable if VAE inference can be performed in under a $1\,\mathrm{ms}$. Therefore, we report the results of our ablation study where the VAE's channel capacity is systematically reduced. Table I summarises the results of this study. Firstly, the latent dimension is reduced from 125 to a minimum of 6. The VAE with latent size of 6 units is deployed successfully in simulation and on the real robot. Next, the VAE's width is reduced in 32-unit increments and a limit of 128 units is found. This corresponds to a reduction in channel capacity by $52.8\,\%$.

Additionally, the window of time used to construct the VAE's input is reduced from $0.4\,\mathrm{s}$ to $0.3\,\mathrm{s}$ whilst maintaining an encoder frequency of $200\,\mathrm{Hz}$. This results in poor open-loop performance as the VAE is no longer able to learn the gait phase. This is a result of the swing duration in the dataset which is $0.45\,\mathrm{s}$. Therefore, an input history over the last $0.3\,\mathrm{s}$ is insufficient to capture the gait phase. Next, we investigated reducing the encoder's sampling frequency f_{enc} by halving it to $100\,\mathrm{Hz}$ whilst the history remains sampled over $0.4\,\mathrm{s}$. Though this speeds up inference, the resulting trajectories are less smooth than the $200\,\mathrm{Hz}$ encoder and the robot is not stable during closed-loop operation.

As a result of these analyses, the minimum VAE architecture requires a latent space of 6 units, an input history of $0.4\,\mathrm{s}$ sampled at $200\,\mathrm{Hz}$, and a hidden layer size of 128 neurons. We confirm this by deploying this reduced model on the real robot.

TABLE I: We perform an ablation study to find the minimum channel capacity required for the VAE-planner to transfer successfully to the real robot. The original VAE model denoted as (O) is found in the bottom right-hand corner of this table. The latent-space size is reduced to six whilst the width remains 256. The width is reduced to 64 in units of 32. The smallest model which transfers to the real robot has width of 128 and latent size of six.

	Model Width				
Latent Size	64	96	128	192	256
6	**F**	**F**	**T**	**T**	**T**
29	-	-	-	-	**T**
61	-	-	-	-	**T**
125	-	-	-	-	**O**

D. Analysing the encoder's receptive field

It is crucial that the VAE is able to infer the robot's gait phase from raw sensor input to permit successful latent-space planning. This poses the question which parts of the encoder input are used to infer this.

Fig. 7 depicts a saliency map where a bright colour denotes areas of high gradient. These are areas where the VAE's encoder focuses to infer the gait phase. The left map in Fig. 7 shows the receptive field of the encoder over the entire input $\mathbf{X}_k$. Here, the input $\mathbf{X}_k$ is reshaped such that each row is the robot state $\mathbf{x}_k$, see Eq. 1. The columns represent specific robot quantities which we have highlighted in Fig. 7. The areas of high gradient are the contact forces and the most recent joint-torques. The left part of Fig. 7 are the contact forces normal to

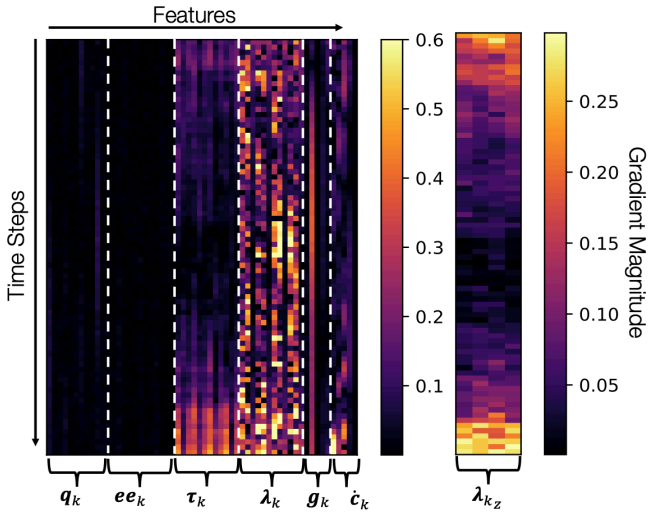


Fig. 7: Visualisation of the encoder's receptive field with respect to the gait phase. The encoder input is reshaped such that robot state quantities such as joint torques are in the same column. We also show the contact forces normal to the ground plane λ_{zk}. The lightest areas in these two sub-figures are of highest gradient meaning that the encoder focuses on these parts in order to infer the gait phase.

the ground plane (i.e. the z-direction). This plot reveals that the most recent and earliest contact forces are utilised to infer the gait phase. High values of contact force correlate well with the contact state. The joint torques are related to the contact forces through the robot dynamics' equation [39]. Lastly, the base-velocity has a high gradient value. This quantity is utilised in inferring the robot's momentum.

E. Varying locomotion parameters online

We leverage the disentangled latent-space to smoothly transition between gait parameters whilst the robot is walking. Crucially, cadence, stance duration, and footstep height can be varied during any phase of the gait by modulating the drive signal's parameters (see Sec. III-A). This results in operating modes which vary from those seen during training. Examples of walking motion using the VAE-planner on the real robot are shown in Fig. 4.

Swing Duration: The swing duration is varied over a large operating window on the ANYmal robot. This begins with a swing time-period starting at $312.5\,\mathrm{ms}$, and is smoothly varied until the swing duration reaches $125\,\mathrm{ms}$, more than doubling the step rate. In parallel, we alter the robot's heading, demonstrating the independence of the action and the latent-space dynamics. Specifically, the top row of Fig. 4 shows the nominal swing duration of $312.5\,\mathrm{ms}$ as the robot turns clockwise, tracking a constant angular velocity command. Following this, we demand a slightly faster swing $188\,\mathrm{ms}$ and a constant angular velocity anti-clockwise, before transitioning to the fastest swing ($125\,\mathrm{ms}$) in the third row. Here, the coloured contact schedule captures the changes in swing duration as it occurs in real-time.

Stance Duration: Following a successful reduction in cadence, the stance duration is reduced and the robot transitions into a trot with negligible full-stance phase: $\epsilon_k = 0$. Trot gaits with little to no full-support phase are particularly challenging manoeuvres for the system in general, as there is reduced control authority to correct for accrued base pose error. During the swing phase of this gait style, only feet across the diagonal are in contact resulting in a line contact limiting the robot's ability to steady its base. The transition to a negligible full-stance phase is captured in the third row, where the coloured stance duration reduces in length. Note that the full-stance duration does not reduce to zero seconds. The mean lowest stance-duration is found to be $19.4\,\mathrm{ms}$. We can see this in Fig. 10, which shows the spread of achievable swing and stance durations compared to the those in the dataset. The right sub-plot shows the stance durations, where we see that the bottom whisker of plot does not quite touch zero. The reason for this is that the full-support region in the latent-space must be traversed. This part of latent space is highlighted in purple in Fig 2. It is possible to introduce a non-smooth drive-signal which jumps this region, but this discontinuous drive-signal would introduce a large acceleration into the resulting locomotion trajectory. This might not be trackable by the WBC. Instead we recommend utilising a smooth and continuous drive-signal.

Footstep Height: We vary the amplitude of the drive signal smoothly to zero as is seen in the bottom row (Fig. 4): The footstep height reduces to zero as the white-space in the contact schedule disappears and the robot remains standing. Beyond versatility, e.g. to increase swing heights to overcome irregular ground height, this capability further enables a safe, smooth and natural transition into and out of the VAE control mode (i.e. to start and come to a halt).

F. ZMP of planned motion as a measure of dynamic feasibility

On the robot, the trajectories from the VAE-planner are sent directly to the WBC unaltered. Here, we compare dynamic feasibility of the trajectories shown in Sec. VI-E after the experiment in order to compare to the training dataset's distribution generated using DG. The position of the ZMP relative to the support line is utilised as a metric for this comparison. The optimal trajectory will have the ZMP lie on the support line.

The results of this are summarised in Fig. 9. The distribution the ZMP positions is similar for both DG (dataset) and a range of the VAE-planner's operating modes. Crucially, this remains true despite the maximum swing duration generated by the VAE-planner being up to 3.2 times faster than in the training data. We conclude that the representation is good enough to generalise to the robot's dynamics shown here. Empirically, we have further been able to steer the robot with arbitrary and fast changing input actions for x, y, and yaw rates, issued from a remote control while being able to interpolate the gait style.

G. Generalisation to different trot-gait styles

The VAE is able to produce a far broader range of different trot styles than those in the dataset. The ranges of swing

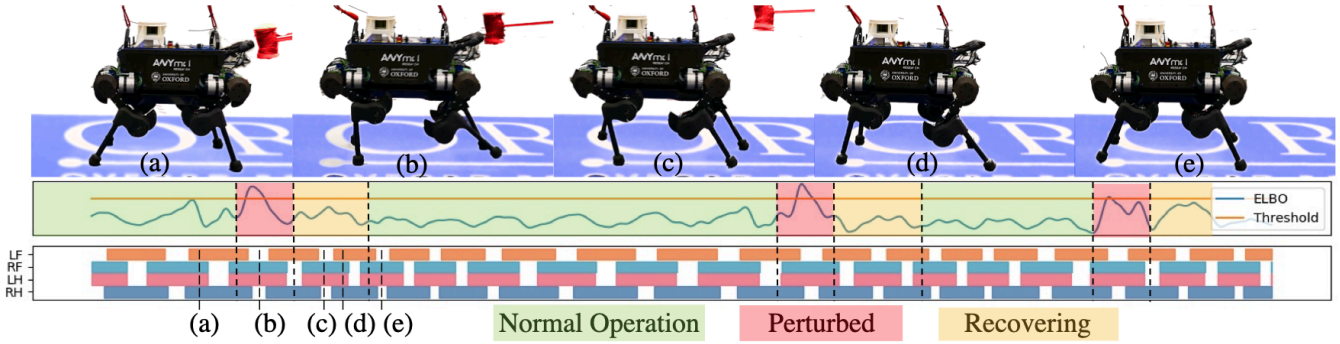


Fig. 8: Push recovery following a shove to the base. A disturbance to the robot causes the ELBO of the form in Eq. 4 to rise above a predetermined threshold (red areas). This identifies a disturbance and triggers an increase in cadence from 250 ms to 125 ms as a rudimentary response. This reduction in the swing duration is visible in the contact schedule. To appreciate the results fully, please see the following video https://youtu.be/GT2WLh2Ackc.

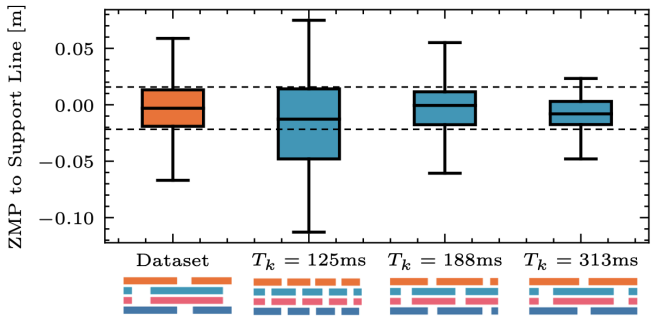


Fig. 9: Distribution of the signed distance between the ZMP and the support line during the 2-stance phases (i.e., one pair of legs was in swing). In orange, the distribution with upper and lower bounds (dashed lines) from the dataset which uses a swing duration of $T_k = 0.5$ s. In blue, the distribution of the VAE trajectories shown over a range of modes deployed on the real robot.

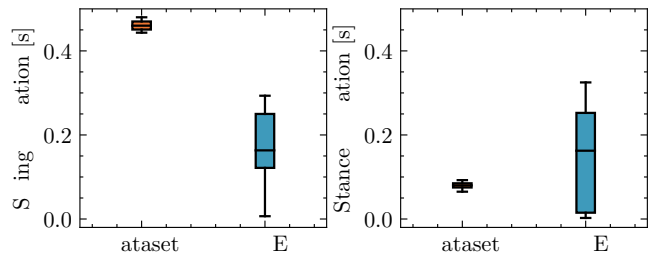


Fig. 10: Comparison of the distributions of swing and full-stance durations between the dataset used for training and the motions executed on the real robot with the VAE and different drive signal parameters during the test presented in Sec. VI-E.

and stance durations produced by the VAE in Sec. VI-E and those in the dataset are summarised in Fig. 10. As discussed previously, the dataset is constructed utilising a pre-determined and constant swing duration of 500 ms, with stance duration of 75 ms. The VAE-planner's swing duration varies between 312.5 ms and 125 ms and its stance ranges between 325 ms to negligible duration. This demonstrates that the VAE-planner is able to generalise to a variety of trot styles by navigating the structure within the disentangled latent-space, even when trained using a very restrictive training set.

H. Disturbance detection and recovery

Our approach is able to detect and mitigate disturbances applied to the robot using a heuristic policy which monitors the ELBO loss whilst the robot is walking. The robot is pushed with a broom resulting in a large spike in the ELBO loss. A spike above a certain threshold is characterised as a disturbance. For our push experiments, we choose the ELBO threshold value to be 11.0. We find this value by walking the robot around the lab using the VAE-planner and choosing a value above what is seen during normal operation.

The VAE-planner in nominal conditions is able to reject a wide range of impulses applied to the robot's base. However, this operating window is enlarged by increasing the robot's cadence as soon as a disturbance is detected. This *reactive* VAE-planner halves the robot's swing duration from a nominal 250 ms to 125 ms once the ELBO threshold is surpassed.

We show the VAE-planner detecting and reacting to the push broom disturbance in Fig. 8. The top row shows the robot being pushed violently before recovering in three to four steps. At point (b) the ELBO spikes above the threshold indicating the disturbance. Throughout the next 1.5 s, the cadence is halved. This is visible in the contact schedule—the white space reduces in width, see point (c). At points (d) and (e) the robot has fully recovered. The ELBO and contact schedule plots continue on and the robot is pushed twice more in this figure.

A comparison between the reactive VAE-planner and a constant cadence version is drawn in Fig. 12. The constant cadence VAE-planner uses swing durations of 313 ms and 250 ms, whilst the reactive version halves the robot's cadence to 125 ms as described. Here, we see that the range of rejectable push disturbances increases as the swing duration decreases.

Finally, we illustrate what happens to the latent-space trajectory during and after the disturbance. The nominal latent-space trajectory forms a figure of eight and is typified by triangular lobes. This is visible in Fig. 11 sub-figures (a) and (e). When the robot is perturbed these lobes deform and elongate e.g. in

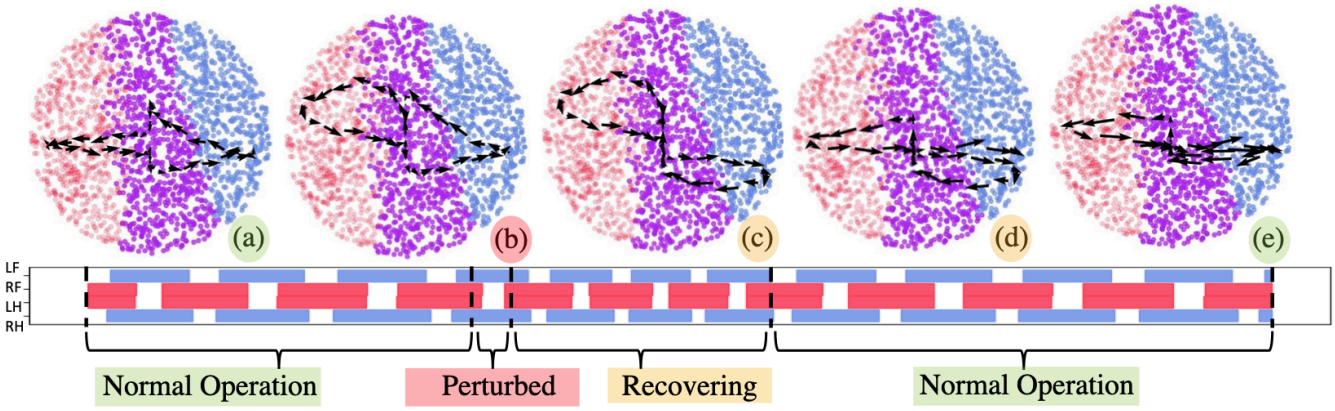


Fig. 11: We show the latent-space trajectory in a series of sub-figures as the robot is walking normally, is perturbed, and recovers. The latent-space trajectory (in black) is plotted on the latent-space colour-coded to show the stances of the trot-gait. In normal operation (sub figures (a) and (e)) the latent-space path is a figure of eight with triangular lobes during the robot's footswing (blue and red areas). As the robot is perturbed (sub figure (b)), the lobes are far more elongated. Crucially, this elongated trajectory reverts back to the nominal trajectory as the robot recovers. The accompanying footstep schedule is plotted along side.

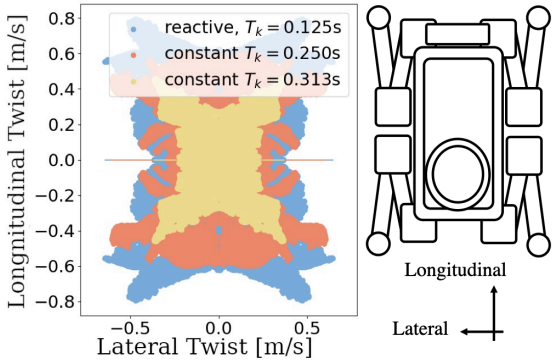


Fig. 12: The range of rejectable base-velocity disturbances increases when automatically modulating the cadence. This plot is symmetrical for ease of view. The range of swing durations plotted is 313 ms, 250 ms, and 125 ms.

sub-figures (b) and (c). The latent-space trajectory converges back to nominal causing the robot to recover as in sub-figures (c) and (d). Finally, the robot has fully recovered and the latent-space trajectory return to nominal (sub-figure (e)).

I. Transfer to ANYmal C

We successfully deploy the VAE-planner trained using simulated ANYmal B data on ANYmal C. Please note that the VAE is not retrained. The differences between the two robots are significant as ANYmal C's joint torque limits are 80.0 N m instead of 40.0 N m, and ANYmal C's total mass is 50 kg instead of 35 kg. Also note that the WBC's internal model for both kinematics and dynamics are updated when the VAE-planner is deployed on ANYmal C. However, the joint trajectory output from the VAE-planner is the input to the WBC meaning that joint trajectories from the VAE-planner are suitable for ANYmal C.

Despite not retraining, the VAE-planner is able to command ANYmal C's heading and control the robot's gait parameters. In essence, the experiment in Sec. VI-E is repeated with the same VAE-planner, but deployed on ANYmal C. Fig. 13 shows ANYmal C with the VAE-planner deployed onboard, along with the contact schedule. The latter shows the variation in both swing and stance duration as the robot walks.

Comparing ANYmal B and C using ELBO: The ELBO can be used to further compare the differences between encoding the raw ANYmal B and C data. As mentioned in Sec. V-I, key parameters such as joint torque limits are standardised before encoding. Analysis of the ELBO is split into comparison of the KL divergence and mean-squared reconstruction error between ANYmal B and ANYmal C.

The KL divergence term is particularly affected by latency in the input. For example, if the states in the encoder input are not sampled exactly at the encoder frequency, the KL divergence term will increase. The distributions of these values are quite different between each robot. The KL-divergence values for ANYmal B show a steady mean drift upwards which is negligible for ANYmal C. The median value for ANYmal C is also noticeably lower than for ANYmal B.

As described in Sec. V-I, we compare the KL-divergence values from both robots together using a Mann-Whitney U-test [38]. As a reminder, we record the KL-divergence for both robots and analyse sets of size 2.5×10^4. The resulting U-statistic is 1.82×10^9 and the p-value is $p < 0.001$. Therefore, the null-hypothesis is rejected and the difference in the median values of the KL-divergence from both robots is statistically significant.

The reason for ANYmal C's lower median value is down to the relative performance of the onboard computers on the two ANYmals. ANYmal C has a much faster CPU meaning that the VAE-planner's control loop is comfortably under the 2.5 ms required for real-time control. The result of which is the encoder's input is sampled at the desired encoder frequency. In contrast, on average the control loop on ANYmal B violates the real-time control law roughly 20 % of the time during the experiment run in Fig. 10. This introduces latency into the

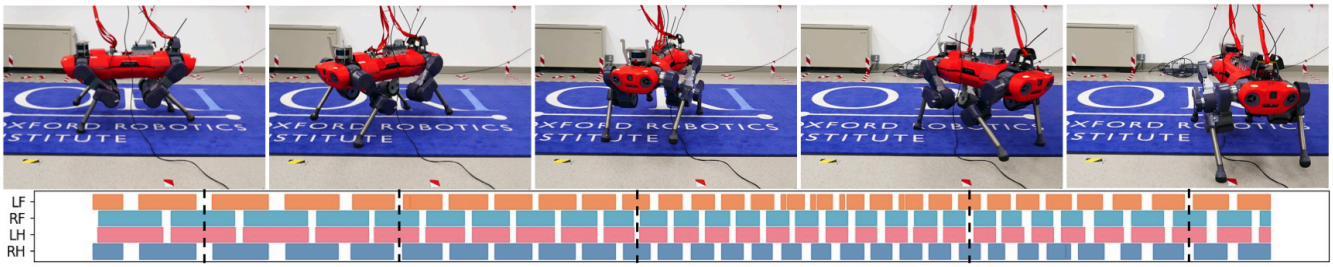


Fig. 13: We deploy our VAE-planner on a different robot: ANYmal C. The VAE used here is trained using data from a simulated ANYmal B robot. ANYmal C is significantly heavier than ANYmal B (50 kg to 35 kg) with twice as much maximum torque, but has actuators with lower bandwidth. With the use of the WBC, the VAE-planner is able to control ANYmal C effectively. The user control over the robot's base twist using the action and the gait parameters are varied as on ANYmal B using the drive-signal parameters. The variation in gait parameters is captured by the swing and stance durations in the contact schedule below the images of the robot. Please view the ANYmal C locomotion in our video found at https://youtu.be/GT2WLh2Ackc.

control loop which causes the increase in KL-divergence.

The reconstruction error for ANYmal B is, unsurprisingly, lower than for ANYmal C. The median value for ANYmal B reconstruction error is 5.74×10^{-3}, whilst for ANYmal C it is 1.58×10^{-2}. These median values are compared using the Mann-Whitney U-test conducted over two sets containing 2.5×10^4 values. The U-statistic is 8.26×10^7 resulting in a p-value of $p < 0.001$. The results in the rejection of the null-hypothesis, meaning that the two distributions are statistically different. Since the VAE-planner is trained using data from the ANYmal B, a lower reconstruction error is expected.

VII. Future work

The most significant limitation of the current system is that, despite the VAE-planner being able to interpolate between a broad range of dynamic manoeuvres, the resulting locomotion gaits are constrained to follow the trot contact sequence and operate on flat ground. This stems from the decision to only train the model using demonstrations of the trot gait with fixed parameters and on flat ground. These decisions were made so that we could inspect and interpret the resulting latent-space thoroughly in this restrictive domain.

To generate different contact schedules other than trot, it is necessary to train a new VAE with different gait trajectories. These different contact schedules should similarly become embedded in latent space. This provides an opportunity to train the VAE with even more dynamic gaits such as bound as well as asymmetrical gaits.

Another limitation is that the current VAE-planner has not been exposed to the trajectories required in order to traverse uneven terrain. Therefore, we wish to train the model on a variety of different locomotion gaits generated when the quadruped walks in unstructured environments. The resulting latent-space will be analysed and the VAE deployed as a planner so that the quadruped can operate in these environments. This will discover firstly how well the VAE-planner reacts to external disturbances resulting from uneven terrain, and secondly the range of motion the VAE-planner can produce in response to the unstructured terrain.

VIII. Conclusion

In this paper, we present a robust and flexible approach for locomotion planning from the perspective of traversing a structured latent-space. This is achieved utilising a deep generative model to capture relevant structure from locomotion data and enables the detection and mitigation of disturbances. The latent space is disentangled to a degree that key salient locomotion features are automatically discovered from a single style of trot gait. An investigation of this latent space reveals a two-dimensional representation which encapsulates the underlying dynamics of the system. This disentanglement is exploited using a drive signal with which dynamically consistent locomotion is generated. Crucially, the amplitude and phase of the drive signal directly control the gait characteristics, namely the cadence, swing height, and full-support duration. Once deployed, the ease with which modulation of the drive signal gives rise to seamless interpolation between gait parameters is demonstrated. Despite generalising to remarkably distinct trot styles compared to the training distribution, the entire range of VAE trajectories remains dynamically consistent. Additionally, utilising a generative model affords the ability to characterise disturbances as out of the distribution seen during training. Though the VAE-planner is able to reject a broad range of impulses applied to the robot's base, this window is broadened by increasing the cadence as soon as the disturbance is detected. Finally, we show that the approach readily transfers to a kinematically similar but dynamically different platform without needing to be retrained. This helps to showcase the VAE's ability transfer to new unseen domains.

Acknowledgements

The authors would like to acknowledge the use of the SCAN facility, and thank Oliver Groth for useful discussions.

References

[1] C. D. Bellicoso, F. Jenelten, C. Gehring, and M. Hutter, "Dynamic locomotion through online nonlinear motion optimization for quadrupedal robots," *IEEE Robot. Automat. Lett. (RA-L)*, vol. 3, no. 3, pp. 2261–2268, 2018.

[2] C. Mastalli, W. Merkt, J. Marti-Saumell, H. Ferrolho, J. Sola, N. Mansard, and S. Vijayakumar, "A feasibility-driven approach to control-limited ddp," *Autonomous Robots*, 2022.

[3] O. Melon, R. Orsolino, D. Surovik, M. Geisert, I. Havoutis, and M. Fallon, "Receding-horizon perceptive trajectory optimization for dynamic legged locomotion with learned initialization," in *IEEE Int. Conf. Rob. Autom. (ICRA)*, 2021.
[4] A. W. Winkler, C. D. Bellicoso, M. Hutter, and J. Buchli, "Gait and trajectory optimization for legged systems through phase-based end-effector parameterization," *IEEE Robot. Automat. Lett. (RA-L)*, vol. 3, no. 3, pp. 1560–1567, July 2018.
[5] C. Mastalli, W. Merkt, G. Xin, J. Shim, M. Mistry, I. Havoutis, and S. Vijayakumar, "Agile maneuvers in legged robots: a predictive control approach," 2022. [Online]. Available: https://arxiv.org/abs/2203.07554
[6] J. Hwangbo, J. Lee, A. Dosovitskiy, D. Bellicoso, V. Tsounis, V. Koltun, and M. Hutter, "Learning agile and dynamic motor skills for legged robots," *Science Robotics*, vol. 4, no. 26, 2019.
[7] S. Gangapurwala, A. Mitchell, and I. Havoutis, "Guided constrained policy optimization for dynamic quadrupedal robot locomotion," *IEEE Robot. Automat. Lett. (RA-L)*, vol. 5, no. 2, pp. 3642–3649, 2020.
[8] S. Gangapurwala, M. Geisert, R. Orsolino, M. Fallon, and I. Havoutis, "Rloc: Terrain-aware legged locomotion using reinforcement learning and optimal control," *IEEE Transactions on Robotics*, vol. 38, no. 5, pp. 2908–2927, 2022.
[9] A. W. Winkler, F. Farshidian, D. Pardo, M. Neunert, and J. Buchli, "Fast trajectory optimization for legged robots using vertex-based ZMP constraints," *IEEE Robot. Automat. Lett. (RA-L)*, vol. 2, no. 4, pp. 2201–2208, Oct 2017.
[10] M. Vukobratovic and B. Borovac, "Zero-moment point - thirty five years of its life." *Int. J. Hum. Rob. (IJHR)*, vol. 1, pp. 157–173, 2004.
[11] D. Orin, A. Goswami, and S.-H. Lee, "Centroidal dynamics of a humanoid robot," *Autonomous Robots*, vol. 35, 10 2013.
[12] N. Rathod, A. Bratta, M. Focchi, M. Zanon, O. Villarreal, C. Semini, and A. Bemporad, "Model predictive control with environment adaptation for legged locomotion," *IEEE Access*, vol. 9, pp. 145 710–145 727, 2021.
[13] F. Farshidian, M. Neunert, A. W. Winkler, G. Rey, and J. Buchli, "An efficient optimal planning and control framework for quadrupedal locomotion," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, may 2017.
[14] C. Mastalli, M. Focchi, I. Havoutis, A. Radulescu, S. Calinon, J. Buchli, D. G. Caldwell, and C. Semini, "Trajectory and foothold optimization using low-dimensional models for rough terrain locomotion," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 1096–1103.
[15] A. L. Mitchell, M. Engelcke, O. Parker Jones, D. Surovik, S. Gangapurwala, O. Melon, I. Havoutis, and I. Posner, "First steps: Latent-space control with semantic constraints for quadruped locomotion," in *IEEE/RSJ Int. Conf. Intell. Rob. Sys. (IROS)*, 2020, pp. 5343–5350.
[16] D. Kingma and M. Welling, "Auto-encoding variational bayes," in *Int. Conf. on Learn. Repr. (ICLR)*, 2014.
[17] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," in *Int. Conf. on Mach. Learn. (ICML)*, 2014.
[18] B. E. Moyer, A. J. Chambers, M. S. Redfern, and R. Cham, "Gait parameters as predictors of slip severity in younger and older adults," *Ergonomics*, vol. 49, pp. 329–343, 2006.
[19] A. L. Mitchell, W. Merkt, M. Geisert, S. Gangapurwala, M. Engelcke, O. P. Jones, I. Havoutis, and I. Posner, "Next steps: Learning a disentangled gait representation for versatile quadruped locomotion," in *IEEE Int. Conf. Rob. Autom. (ICRA)*, 2022, pp. 10 564–10 570.
[20] R. Grandia, A. J. Taylor, A. D. Ames, and M. Hutter, "Multi-layered safety for legged robots via control barrier functions and model predictive control," *CoRR*, 2020.
[21] G. Bledt and S. Kim, "Extracting legged locomotion heuristics with regularized predictive control," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 406–412.
[22] M. Karl, M. Soelch, J. Bayer, and P. van der Smagt, "Deep variational bayes filters: Unsupervised learning of state space models from raw data," in *Int. Conf. on Learn. Repr. (ICLR)*, 2016.
[23] M. Watter, J. T. Springenberg, J. Boedecker, and M. A. Riedmiller, "Embed to control: A locally linear latent dynamics model for control from raw images," in *NeurIPS*, 2015.
[24] M. Y. Seker, M. Imre, J. H. Piater, and E. Ugur, "Conditional neural movement primitives," in *Rob.: Sci. Sys. (RSS)*, 2019.
[25] A. Srinivas, A. Jabri, P. Abbeel, S. Levine, and C. Finn, "Universal planning networks," in *Int. Conf. on Mach. Learn. (ICML)*, 2018.
[26] D. Hafner, T. Lillicrap, I. Fischer, R. Villegas, D. Ha, H. Lee, and J. Davidson, "Learning latent dynamics for planning from pixels," in *Int. Conf. on Mach. Learn. (ICML)*, vol. 97, 2019, pp. 2555–2565.
[27] T. Li, R. Calandra, D. Pathak, Y. Tian, F. Meier, and A. Rai, "Planning in learned latent action spaces for generalizable legged locomotion," *CoRR*, vol. abs/2008.11867, 2020.
[28] H. Ling, F. Zinno, G. Cheng, and M. Panne, "Character controllers using motion VAEs," *ACM Trans. Graph.*, vol. 39, 07 2020.
[29] A. Agrawal, S. Chen, A. Rai, and K. Sreenath, "Vision-aided dynamic quadrupedal locomotion on discrete terrain using motion libraries," *CoRR*, vol. abs/2110.00891, 2021.
[30] Y. Yang, T. Zhang, E. Coumans, J. Tan, and B. Boots, "Fast and efficient locomotion via learned gait transitions," in *Conf. on Rob. Learn. (CoRL)*, 2021.
[31] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner, "β-VAE: Learning basic visual concepts with a constrained variational framework." in *Int. Conf. on Learn. Repr. (ICLR)*, 2017.
[32] C. Dario Bellicoso, F. Jenelten, P. Fankhauser, C. Gehring, J. Hwangbo, and M. Hutter, "Dynamic locomotion and whole-body control for quadrupedal robots," in *IEEE/RSJ Int. Conf. Intell. Rob. Sys. (IROS)*, 2017, pp. 3359–3365.
[33] J. Hwangbo, J. Lee, and M. Hutter, "Per-contact iteration method for solving contact dynamics," *IEEE Robot. Automat. Lett. (RA-L)*, vol. 3, no. 2, pp. 895–902, 2018.
[34] G. A. Pratt and M. M. Williamson, "Series elastic actuators," in *IEEE/RSJ Int. Conf. Intell. Rob. Sys. (IROS)*, 1995.
[35] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (elus)," in *Int. Conf. on Learn. Repr. (ICLR)*, 2016.
[36] M. Hutter, C. Gehring, D. Jud, A. Lauber, C. D. Bellicoso, V. Tsounis, J. Hwangbo, K. Bodie, P. Fankhauser, M. Bloesch, R. Diethelm, S. Bachmann, A. Melzer, and M. Hoepflinger, "ANYmal - a highly mobile and dynamic quadrupedal robot," in *IEEE/RSJ Int. Conf. Intell. Rob. Sys. (IROS)*, 2016.
[37] D. Erhan, Y. Bengio, A. Courville, and P. Vincent, "Visualizing Higher-Layer Features of a Deep Network," University of Montreal, Tech. Rep. 1341, Jun. 2009, also presented at the ICML 2009 Workshop on Learning Feature Hierarchies, Montréal, Canada.
[38] H. B. Mann and D. R. Whitney, "On a Test of Whether one of Two Random Variables is Stochastically Larger than the Other," *Ann. Math. Statist.*, vol. 18, no. 1, pp. 50–60, 03 1947.
[39] M. Mistry, J. Buchli, and S. Schaal, "Inverse dynamics control of floating base systems using orthogonal decomposition," in *IEEE Int. Conf. Rob. Autom. (ICRA)*, 2010.



Alexander L. Mitchell received his M.Eng. from the University of Oxford in 2018. Subsequently, he enrolled at the University of Oxford as a Doctoral candidate. He is currently co-supervised by Professor Ingmar Posner and Dr Ioannis Havoutis at the Oxford Robotics Institute. Alexander's research interests include optimal control and learning-based methods for path planning and control of legged robots.



Wolfgang Merkt received the B.Eng.(Hns) degree in mechanical engineering with management and the M.Sc.(R) and Ph.D. degrees in robotics and autonomous systems from the University of Edinburgh, Edinburgh, U.K., in 2014, 2015 and 2019, respectively.

He is currently a Senior Researcher at the Oxford Robotics Institute, University of Oxford with I. Havoutis. During his Ph.D., he worked on trajectory optimization and warm starting optimal control for high-dimensional systems and humanoid robots under the supervision of S. Vijayakumar. His research interests include fast learning- and optimization-based methods for planning and control, loco-manipulation, and legged robots.



Mathieu Geisert received his M. Eng. in Aerospace from Institut Supérieur de l'Aéronautique et de l'Espace SUPAERO (Toulouse, France) in 2013. He then joined the humanoid robotics team Gepetto at LAAS-CNRS (Toulouse, France) where he worked for 5 years in different positions and received his PhD in 2018. From 2018 to 2021, he worked on quadruped robots as a post doctoral researcher in the Dynamic Robot Systems (DRS) group (Oxford Robotics Institute - University of Oxford). From 2021, he has been working as a controls research engineer at Arrival. His research focuses on Control and Learning for Legged Robots Locomotion.



Siddhant Gangapurwala completed his Bachelor of Engineering (B.E.) in Electronics from the University of Mumbai in 2016. In 2017, he joined the AIMS program as a doctoral candidate at the University of Oxford. The following year, Siddhant joined the Dynamic Robot Systems (DRS) group of the Oxford Robotics Institute (ORI) to pursue his DPhil degree with focus on machine learning and optimal control based approaches for robotic locomotion over uneven terrain. He continues to work at DRS as a post-doctoral researcher on locomotion, manipulation and loco-manipulation.



Oiwi Parker Jones received a D.Phil. in natural language processing at the University of Oxford, Oxford, U.K. From there he went on to do postdoctoral work in language neuroscience, computational modelling, and the development of neuroimaging methods for large datasets at University College London and at the University of Oxford. He is currently a senior postdoc in the Applied AI Lab (A2I) at the Oxford Robotics Institute, honorary fellow in the Nuffield Department of Clinical Neurosciences, and Hugh Price Fellow in Computer Science at Jesus College, University of Oxford.



Martin Engelcke received the M.Eng. and D.Phil. (Ph.D.) from the University of Oxford, Oxford, UK. He conducted his doctoral and subsequent postdoctoral research between 2015 and 2021 at the Oxford Robotics Institute under the supervision of Prof. Ingmar Posner. He is now a research scientist at DeepMind. His research interests include generative models, representation learning, and reinforcement learning.



Ioannis Havoutis received the M.Sc. in artificial intelligence and Ph.D. in Informatics degrees from the University of Edinburgh, Edinburgh, U.K.

He is a Lecturer in Robotics at the University of Oxford. He is part of the Oxford Robotics Institute and a co-lead of the Dynamic Robot Systems group. His focus is on approaches for dynamic whole-body motion planning and control for legged robots in challenging domains. From 2015 to 2017, he was a postdoc at the Robot Learning and Interaction Group, at the Idiap Research Institute. Previously, from 2011 to 2015, he was a senior postdoc at the Dynamic Legged System lab the Istituto Italiano di Tecnologia. He holds a Ph.D. and M.Sc. from the University of Edinburgh.



Ingmar Posner is Professor of Engineering Science at Oxford University, where he leads the Applied Artificial Intelligence Lab. His research aims to enable machines to robustly act and interact in the real world - for, with, and alongside humans. Ingmar's track record in machine perception and decision-making includes seminal work on large-scale learning from demonstration, representation learning for scene understanding and prediction as well as 3D object detection and machine introspection. Currently his research interests revolve around the use of structured latent spaces for robot perception, planning and control.

Ingmar received an M.Eng. degree in Electronic Systems Engineering from Aston University, Birmingham, U.K., and a D.Phil. degree in bioacoustics from the University of Oxford. He is a founding Director of the Oxford Robotics Institute and, in 2014, co-founded Oxbotica, multi-award winning provider of mobile autonomy software solutions.

Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor **(only required where there isn't already a statement of contribution within the paper itself).**

Title of Paper	VAE-Loco: Versatile Quadruped Locomotion by Learning a Disentangled Gait Representation
Publication Status	Published
Publication Details	Alexander L. Mitchell, Wolfgang Merkt, Mathieu Geisert, Siddhant Gangapurwala, Martin Engelcke, Oiwi Parker Jones, Ioannis Havoutis, Ingmar Posner Published in the Journal of IEEE Transactions on Robotics

Student Confirmation

Student Name:	Alexander Luis Mitchell		
Contribution to the Paper	Led efforts on: • Proposed work and created the team • Created and developed methodologies • Ran experiments • Created all figures Co-led efforts: • Paper writing • Experiment support • Optimising deployment code		
Signature		Date	02/OCT/2023

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title: Professor Ioannis Havoutis		
Supervisor comments These comments are correct.		
Signature Ioannis Havoutis	Date	02/OCT/2023

5

Gaitor: Learning a Unified Representation Across Gaits for Real-World Quadruped Locomotion

This chapter introduces *Gaitor*, which investigates the opportunities and advantages that learning a unified representation across gaits yields. This direction of research is inspired by the previous chapter *VAE-Loco*, which discovers a disentangled and structured latent representation for a single gait type, the trot gait. The representation in *VAE-Loco* is expressive enough to capture a broad range of trot gait characteristics and it is found that planning in this space can produce locomotion with continuously variable swing speed, lengths and heights. However, *VAE-Loco* is limited to a single gait and blind locomotion on flat ground. Therefore, the methods in this chapter address multiple gait types (trot, crawl, and pace) and the perceptive traversal of uneven terrain.

A variational autoencoder (VAE) [19, 20] in combination with performance predictors [18] similar to the one in *VAE-Loco* is trained using examples of distinct gait type, trot, crawl and pace. The VAE and performance predictors are trained by minimising the beta-VAE [21] loss summed with the binary cross-entropy loss (BCE) for predicting the correct contact state similarly to *VAE-Loco*. Inspection of the latent space reveals that a unified and interpretable representation across gaits is created. As a result, it is possible to find continuous transitions between the gait types in the space via novel intermediary

gaits. This capability is realised since the VAE captures correlations across all the gait in the latent space. Gait transitions occur since the relative phase differences between the legs are inferred and embedded in the structure of the latent space. For example, in the pace gait, legs on the same side of the robot's body swing together in phase, whilst during trot, these legs move in the opposite direction and are out of phase. During gait transitions from the trot to pace gait, the legs move out-of-phase to in-phase progressively and during the mid-point, only the leg swings at once producing a crawl gait. Hence, smooth and realisable gait transitions occur between all three of the gaits seen during training.

As mentioned, the architecture used in this chapter is similar to that of *VAE-Loco* but incorporates a terrain encoder and includes a learnt planner in the latent space. The input to the terrain encoder is the heights of the terrain sampled at the locations of the future footholds. These heights are processed before encoding and this process is outlined in Fig. 3 of this chapter. The future foothold locations are estimated using the robot's base velocity and the robot's kinematics. These heights are estimated just before the robot takes a physical step which is much slower than the encoding frequency of the VAE. This means that these heights vary in a non-smooth fashion. To create a smooth input signal for the terrain encoder, we use a processing stage. The processing stage puts these heights through a second-order filter and the output varies predictably and smoothly. The VAE and terrain encoder are trained together, whilst the learnt planner's weights are optimised afterwards. It is discovered that the resulting latent space is meaningfully disentangled in a similar way to *VAE-Loco* meaning that the footswing lengths and heights can be varied independently using in this paper elliptical trajectories in the space. The eccentricity of the trajectory in latent space can vary the swing heights and lengths. Therefore, the learnt planner takes the terrain encoding and desired gait phase as input and predicts the radius of the ellipse in polar coordinates. Using polar coordinates, the gait phase is the angle and the radius dictates the eccentricity of the ellipse.

The VAE with terrain encoder and planner are deployed on the real ANYmal C robot in a closed-loop planning framework. As in *VAE-Loco*, a history of robot states is encoded to create a robot encoding and infer the robot's current momentum-aware state. The terrain is processed as described and encoded creating the terrain encoding. The robot

encoding, terrain encoding and user inputs such as the desired gait type, and desired base velocity are concatenated together. The operator controls how many footswings occur per minute and the gait phase evolution. This gait phase and the terrain encoding are inputs to the learnt planner which outputs the next states in the latent space. The latent state is decoded and the joint-space trajectory is tracked using an optimisation-based whole-body controller (WBC) [33].

The latent space structure is inspected during terrain traversal and reveals that the swing characteristics are encoded differently depending on the terrain in front of the robot. Additionally, the learnt planner in the latent space adjusts the robot's footstep characteristics in response to terrain. The latent space structure and learnt planner promote robust foothold placement during the initial climb phase. In particular, longer footswing lengths are encouraged as the robot approaches a pallet. As the robot nears the step, it swings its feet further away from the edge of the step. This is beneficial as this foothold is safer than an alternative location nearer the edge of the step.

To the best of our knowledge, *Gaitor* is the first method showing that a data-driven approach can create a unified representation of locomotion across multiple gait types via a disentangled and interpretable latent space. We demonstrate in our experiments that the latent representation forms a suitable planning manifold to generate continuous transitions between gaits via novel intermediate contact schedules and can perceptively traverse terrains. Inspection of the latent space reveals that the gait transitions are interpretable and the transition mechanism is a result of the inferred correlations between the gaits. In addition, as the robot approaches changes in terrain, the structure of the space adjusts to promote the generation of robust footholds and we show that with a learnt planner in the space, *Gaitor* can be used as a planner in a closed-loop framework to climb a $12.5\,\mathrm{cm}$ platform.

The *Gaitor* article is submitted to the Conference on Robot Learning (CoRL) 2024.

Gaitor: Learning a Unified Representation Across Gaits for Real-World Quadruped Locomotion

Alexander L. Mitchell, Wolfgang Merkt, Aristotelis Papatheodorou, Ioannis Havoutis, & Ingmar Posner

***Abstract*—The current state-of-the-art in quadruped locomotion is able to produce robust motion for terrain traversal but requires the segmentation of a desired robot trajectory into a discrete set of locomotion skills such as trot and crawl. In contrast, in this work we demonstrate the feasibility of learning a single, unified representation for quadruped locomotion enabling continuous blending between gait types and characteristics. We present *Gaitor*, which learns a disentangled representation of locomotion skills, thereby sharing information common to all gait types seen during training. The structure emerging in the learnt representation is interpretable in that it is found to encode phase correlations between the different gait types. These can be leveraged to produce continuous gait transitions. In addition, foot swing characteristics are disentangled and directly addressable. Together with a rudimentary terrain encoding and a learned planner operating in this structured latent representation, *Gaitor* is able to take motion commands including desired gait type and characteristics from a user while reacting to uneven terrain. We evaluate *Gaitor* in both simulated and real-world settings on the ANYmal C platform. To the best of our knowledge, this is the first work learning such a unified and interpretable latent representation for *multiple* gaits, resulting in on-demand continuous blending between different locomotion modes on a real quadruped robot.**

I. Introduction

Recent advances in optimal control [1]–[5] and reinforcement learning (RL) [6]–[10] have enabled robust quadruped locomotion over uneven terrain. This has made quadrupeds a promising choice for the execution of inspection, monitoring and search and rescue tasks. Current state-of-the-art methods such as [7], [11] are highly capable and produce a variety of complex motions. However, these methods tend to be either engineered *top-down*, incorporating expert knowledge of the locomotion process via appropriate inductive biases, or *bottom-up* in a data-driven way. Typical examples of top-down approaches include leveraging the gait phase via an architectural inductive bias (e.g. [12], [13]) or selecting between different specialised policies to deliver individual skills (e.g. [6]). Such top-down approaches treat gaits as independent skills and are unable to benefit from synergies in terms of efficient learning or added capability beyond the prescribed skill set. In contrast, bottom-up methods tend to be black-box models of a blend of locomotion skills such as a single distribution over gaits learnt using a large transformer model [11]. While successful in terms of delivering locomotion capability, such models are typically unamenable to interpretation and generally do not yield insights into the emergence of the phenomenological inductive biases often used by domain experts when designing top-down approaches.

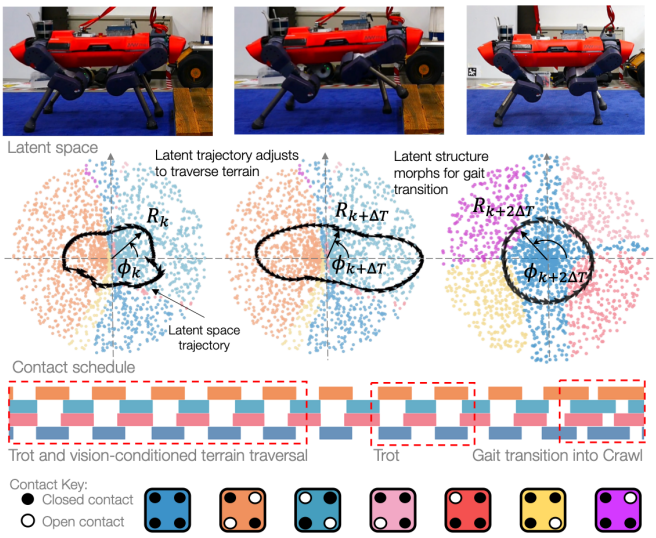


Fig. 1: *Gaitor* is a data-driven approach to learning a single-disentangled representation for quadruped locomotion. This representation is expressive enough to capture the robot's dynamics across multiple gaits (e.g. trot and crawl) and permits continuous gait transition via automatically discovered intermediaries. The latent space is conditioned using a rudimentary terrain encoding facilitating terrain traversal. A learnt planner in the latent space adjusts the gait characteristics in response to the terrain. The structure of the latent space alters as the operator demands a gait transition from trot to crawl.

A notable exception in the data-driven regime, albeit in the context of only a single gait type on flat ground, is *VAE-Loco* [14], [15]. *VAE-Loco* demonstrates that gait-phase relationships emerge when learning a disentangled latent representation based on relatively narrow demonstration data. In particular, the swing heights and lengths are encoded in two independent dimensions of *VAE-Loco's* latent space. Modulating trajectories in this two-dimensional space results in independent control of the robot's in-swing gait characteristics (e.g. swing heights and lengths) on-demand.

Inspired by *VAE-Loco*, and in contrast to related works modelling multiple gaits, in this paper we present *Gaitor*, a data-driven approach to learning and exploiting a single, unified representation of locomotion dynamics in a disentangled latent-space across multiple gaits.

While our work is located in the domain of quadruped locomotion, in contrast to other work in this area our overarching goal is not in the first instance to achieve state-of-the-

art performance in this area, but to explore the challenges and opportunities afforded by learning a single, disentangled representational space to encode the synergies between traditionally separated locomotion skills. In particular, we demonstrate that the structured latent space learnt by *Gaitor* is expressive enough to capture not only the individual gaits, but also continuous transitions between them via intermediary contact patterns, which do not exist in the training data. *Gaitor* further incorporates a rudimentary terrain encoding, which adapts the latent-space structure in response to the ground beneath the robot in a readily interpretable way. In particular, the swing characteristics are encoded differently in response to changes in terrain. *Gaitor* couples these representations with a learnt planner operating in the latent space, and results in robust terrain-aware trajectories capable of traversing significant obstacles. We demonstrate this capability on a real ANYmal C robot [16].

To the best of our knowledge, *Gaitor* is the first work that demonstrates how a data-driven approach can create a unified representation for multiple modes of locomotion via a *disentangled* latent-space. *Gaitor* demonstrates significant advances over *VAE-Loco*, which forms inspiration for this paper. The first improvement of *Gaitor* is to integrate a perception pipeline for uneven terrain traversal. Next, *Gaitor* uses a learnt planner to generate adaptive trajectories in the latent space using a polar coordinate framework to adjust the robot's gait characteristics (footswing lengths and heights) to traverse terrain robustly. After training *Gaitor*, it is discovered that the latent space automatically restructures itself to represent distinct gaits such as trot, crawl and pace. In contrast, the latent space generated using *VAE-Loco* is static and oscillatory trajectories are injected into the space for blind operation on flat ground. As an operator requests a different gait, *Gaitor's* latent space changes shape and novel unseen gait transitions are discovered. Note that the expert pace gait is unstable on the real robot hence only the crawl/pace gait is deployed using our method. Gait transitions arise since the learnt representation captures meaningful correlations across the gait types resulting in a two-dimensional planning manifold in which a variety of complex locomotion tasks are solved. In our experiments on the real ANYmal C robot, we show continuous gait transitions *on-demand* from trot to crawl to a crawl/pace hybrid and vision-aware climbing onto a $12.5\,\mathrm{cm}$ platform.

II. Related Work

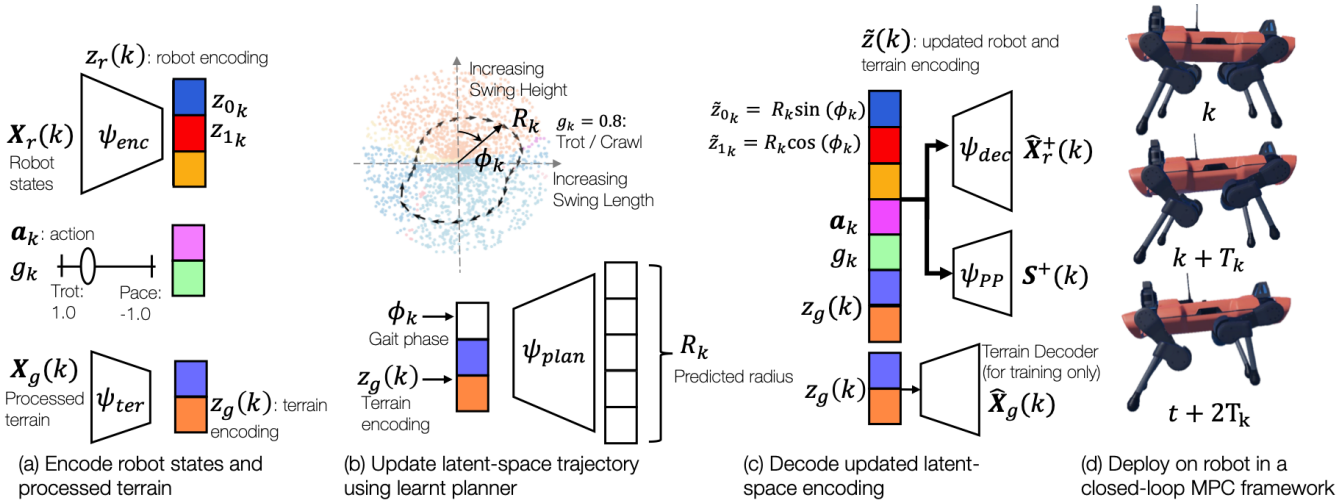
Reinforcement learning (RL) [8], [10], [17] is an increasingly popular choice for locomotion due to advances in simulation capabilities. A recent trend for RL is to solve locomotion problems by learning a set of discrete skills with a high-level planner arbitrating which skill is deployed at any one time. For example, [6] learns a set of 5 distinct skills and a high-level planner to select the appropriate sub-skill to traverse the terrain ahead of the robot. These sub-skills include jumping across a gap or climbing over an obstacle. This method produces impressive behaviour, but no information is shared across tasks, meaning that each skill learns from scratch and correlations across skills are not leveraged. Another impressive work *Locomotion-Transformer* [11] learns a generalist policy for multiple locomotion skills for traversing an obstacle course. This approach uses a transformer model to learn a multi-modal policy conditioned on terrain. In contrast, *Gaitor* learns an explicit and disentangled 2D representation for the dynamics of multiple locomotion gaits. The space infers the phase relationships between each leg and embeds these into the latent-space structure. Unseen intermediate gaits are automatically discovered by traversing this structured latent-space.

Work by [13] tackles gait switching by learning to predict the phases between each leg. The gait phase and the relative phases of each leg form this RL method's action space. The predicted phases create a contact schedule for a model predictive controller (MPC), which computes the locomotion trajectory. In [13], the gait phase is an inductive bias designed to explicitly determine the swing and stance properties of the gait. The utilisation of the gait phase as an inductive bias for locomotion contact scheduling is common in the literature. For example, work by [12] conducted contemporaneously to *Gaitor*, utilises a latent space in which different gaits/skills are embedded. A gait generator operates from this space and selects a particular gait to solve part of the locomotion problem. This utilises a series of low-level skills prior to learn the latent representation using pre-determined contact schedules. In contrast, *Gaitor* infers the relationships between gait types and automatically discovers the intermediary gaits needed for smooth transitions from discrete examples.

Locomotion planning in learnt latent-representations is a growing field of interest. These methods are efficient at learning compact representations of robot dynamics. For example, [18] learns a latent action space and a low-level policy from expert demonstrations using imitation learning. Locomotion is achieved via an MPC formulation, which is solved via random shooting using a learnt dynamics model. Similarly, *VAE-Loco* [14] learns a structured latent-space in which the characteristics of the trot gait are disentangled. This structure is exploited leading to locomotion where the cadence, footstep height and length are varied continuously during the swing phase. *VAE-Loco* is constrained to a single gait and blind operation on flat ground only. This work forms the inspiration for *Gaitor*, which explores learning a single, unified latent representation for quadruped locomotion across *multiple* gaits to achieve perceptive locomotion in uneven terrains.

III. Gaitor

This section outlines the components of *Gaitor*. A variational autoencoder (VAE) [19], [20] similar to *VAE-Loco* [14] is exposed to distinct gaits, trot, pace and crawl. This creates a shared representation across all the gaits and is conditioned on a learnt terrain encoding. Once the representation is created, a learnt planner is trained and operates in the latent space. This planner adjusts the robot's gait characteristics in response to changing terrain.



(a) Encode robot states and processed terrain

(b) Update latent-space trajectory using learnt planner

(c) Decode updated latent-space encoding

(d) Deploy on robot in a closed-loop MPC framework

Fig. 2: *Gaitor* learns a unified representation for locomotion dynamics across multiple gaits in a disentangled latent-space. There are three components to *Gaitor*. The first is a variational autoencoder (VAE) to encode the robot states and build the latent space and is comprised of an encoder ψ_{enc} and decoder ψ_{dec}, see panels (a) and (c). The second is a terrain encoder ψ_{ter} which estimates a rudimentary encoding of the ground ahead of the robot. Lastly, a learnt planner ψ_{plan} creates trajectories in the latent space, see panel (b). The planner's trajectory exploits the latent-space structure to vary the robot's gait characteristics (footswing heights and lengths) in response to terrain. The operator can demand different gaits using the slider g_k and this changes the structure of the latent space and results in an adjustment of the robot state trajectory and contact schedule. The updated latent-space trajectory is decoded to predict a set of future robot states $\mathbf{X}_r^+(k)$ and contact states $\mathbf{S}^+(k)$ using the performance predictor ψ_{PP}, see panel (c). The future trajectory is sent to a whole-body controller as depicted in panel (d).

A. Overview

There are three components to *Gaitor*. These include a VAE [19], [20], a terrain encoder and planner in latent space. Fig. 2 (a) depicts the VAE's encoder ψ_{enc} and the terrain encoder ψ_{ter}. The VAE's decoder ψ_{dec} is shown in Fig. 2 (c). The VAE takes a history of robot states as input $\mathbf{X}_r(k)$ to infer the robot's momentum via a robot encoding $\mathbf{z}_r(k)$ at the current time step k. The terrain encoder operates on processed information taken from a 2.5D representation of the ground around the robot to estimate $\mathbf{z}_g(k)$. The robot encoding and terrain encoding are concatenated with operator-controlled variables such as the desired gait label g_k and the desired base velocity action $\mathbf{a}_k$. This concatenation is input into the decoder ψ_{dec} and a performance predictor ψ_{PP} [21] as depicted in Fig. 2 (c). The decoder reconstructs a small trajectory containing future predictions of the robot states $\hat{\mathbf{X}}_r^+(k)$. The performance predictor estimates which feet are in contact $\mathbf{S}^+(k)$ during the decoded trajectory. During deployment, the predicted trajectory is sent to a tracking controller which estimates the joint torques for the actuators.

The planner predicts a latent-space trajectory, which adapts the gait characteristics (e.g. footswing heights and lengths) in response to the terrain encoding. Similarly to *VAE-Loco*, inspection of *Gaitor's* learnt latent-space in Sec. VI-C reveals that an elliptical trajectory in two dimensions of the representation is sufficient for independent control of the swing heights and lengths. Therefore, the latent-space planner is designed to predict an elliptical trajectory in polar coordinates, where the angle is equal to the gait phase ϕ_k and the radius is the planner's output R_k as seen in Fig. 2 (b). We explain how to process the data into the form required for the planner and its training regime in Sec. IV-B.

B. Learning a Unified Representation

The VAE operates over the robot states only. The robot state at time step k is $\mathbf{x}_k = [\mathbf{q}_k, \mathbf{ee}_k, \tau_k, \lambda_k, \dot{\mathbf{c}}_k, c_{\theta_x}, c_{\theta_y}, \Delta\mathbf{c}_k]$ and represents the joint angles, end-effector positions in the base frame, joint torques, contact forces, base velocity, roll and pitch of the base and the base-pose evolution respectively. The VAE's encoder ψ_{enc} takes a history of these states as input. These samples are recorded at a frequency equal to f_{enc}. The input to the VAE is $\mathbf{X}_k$ and is constructed using N past samples stacked together into a matrix. Similarly the output $\hat{\mathbf{X}}_k^+$ is a trajectory of states made from M predictions into the future from time step k sampled using a frequency f_{dec}. The robot's base is floating in space and must be tracked so that the decoded trajectories can be represented in the world frame. The encoder tracks the evolution of the base pose $\Delta\mathbf{c}_k$ from a fixed coordinate. This coordinate is the earliest time point in the encoder input and all input poses are relative to this frame. All pose orientations are represented in tangent space. This is a common representation for the Lie-Algebra of 3D rotations since composing successive transformations using a Euclidean or a Quaternion representation is non-linear, but in tangent space, this composition is linear, see [22], [23] for examples. The output of the encoder parameterises a Gaussian distribution to create a latent-space estimate of the robot's trajectory $\mathbf{z}_r(k)$.

The decoder predicts the future robot states $\hat{\mathbf{X}}_r^+(k)$ using the inferred latent variable and user-controlled inputs. This includes the base-pose evolution, which is expressed relative to the base pose at the previous time-step. The inputs to the decoder are $\mathbf{z}_r(k)$ the desired action $\mathbf{a}_k$, the desired gait label g_k, and terrain encoding $\mathbf{z}_g(k)$. Importantly, due to our formulation of learning a unified latent space across gaits, $g_k \in \mathbb{R}$ is a *continuous* value used to select a gait or intermediate gait type. Distinct gaits such as trot, crawl, and pace are selected when $g_k = \{1, 0, -1\}$ respectively.

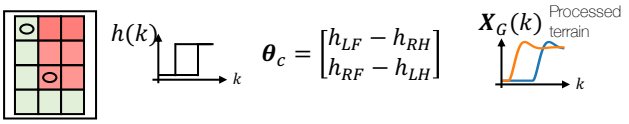
The performance predictor ψ_{PP} estimates the contact states of the feet $\mathbf{S}(k)$ and takes the robot encoding as well as the desired action and gait as input, see Fig. 2. The contact states are predicted during operation and sent to the tracking controller to enforce the contact dynamics.

C. Terrain Encoder

The terrain encoding is required so that the decoder and planner can react to terrain height variations. The robot's perception module provides a 2.5D height map of the terrain. This is constructed from depth images from four depth cameras on the robot. This raw map is filtered and in-painted in a similar fashion to [24]. The height map is sampled at the locations of the next footholds and the heights $\mathbf{h}(k)$ are processed to create the terrain encoder's input $\mathbf{X}_g(k)$. The processing pipeline takes the heights of the future footholds and finds the differences between left front, right hind and right front and left hind respectively. In this work, we define these relative differences as the control pitch θ_c, see Fig. 3 (c). The control pitch varies discontinuously and only updates when the robot takes a step. To convert this discontinuous variable to a continuous one, we filter it using a second-order linear time-invariant (LTI) filter. This converts the step-like control pitch sequence into a continuous time varying signal $\mathbf{X}_g(k)$ of the type shown in Fig. 3 (d). This is achieved by setting the LTI's rise time to occur when the foot in swing is at its highest point and using a damping factor of 0.5. The input to the terrain encoder ψ_{ter} is a matrix of N stacked values $\mathbf{X}_g(k)$ sampled at the control frequency f_c. The output of the terrain encoder $\mathbf{z}_g(k)$ forms part of the decoder's input $\tilde{\mathbf{z}}(k)$ as shown in Fig. 2 (c). During training $\mathbf{z}_g(k)$ is input to a terrain decoder used to reconstruct a prediction of $\mathbf{X}_g(k)$. This element is discarded after training.

D. Latent-Space Planner

The planner produces latent-space trajectories and is designed to adjust the robot's swing length and height to help the robot traverse terrain. It is discovered in *VAE-Loco* [14] that the footswing heights and lengths are encoded into the two dimensions with lowest variance in latent space. These dimensions are denoted as z_0 and z_1 and a 2D elliptical trajectory in these dimensions controls the swing heights and lengths independently. In this paper, the same phenomenon is discovered in Sec. VI-C when the latent space is inspected as shown in Fig. 5. Therefore, we design the planner to exploit the disentangled space by only adjusting z_0 and z_1.



(a) Top-view of height map from vision (b) Sample foot heights $h(k)$ for future steps (c) Calculate control pitch θ_c from height samples (d) Filter $\boldsymbol{\theta}_c$ using LTI filter

Fig. 3: Terrain processing pipeline. The onboard perception module produces a 2.5D height map of the terrain around the robot. This map is sampled at the footfall positions to estimate the heights of the terrain at these locations, see panels (a) and (b). The control-pitch angle θ_c is defined as the difference in heights between the front and rear footfalls, see panel (c). These values are input to a second-order filter to create terrain encoder's input $\mathbf{x}_g$ (panel (d)).

The planner is parameterised in polar coordinates and takes the gait phase $\phi(k)$ and terrain latent $\mathbf{z}_g(k)$ as input to predict the radius of the elliptical trajectory $R(k)$, see Fig. 2 (b). This trajectory updates the values of the two latent dimensions so that

$$\tilde{z}_0(k) = R(k)sin(\phi(k)), \tag{1}$$

$$\tilde{z}_1(k) = R(k)sin(\phi(k) + \pi/2). \tag{2}$$

There are large variations in the values of $R(k)$ as is shown in Sec. VI-D. To ensure the planner reproduces these variations to a high fidelity, an architecture inspired by [25] is used as seen in Fig. 2 (b). The planner predicts a probability distribution r_c for the radius over C discrete values via a softmax. A weighted sum of the predicted probabilities multiplied by the bin values estimates the radius $R(k)$.

IV. Training

In this section, we provide details on the loss functions used to train the individual components of *Gaitor*. The weights of the VAE and terrain AE are optimised in concert to help structure the latent space. The planner is trained last as it uses operates in the learnt space.

A. Training the VAE and Terrain AE

The VAE and performance predictor are optimised using the evidence lower bound (ELBO) and a binary cross entropy (BCE). The performance predictor estimates which feet are in contact $\mathbf{S}^+(k)$ given the robot-state input $\mathbf{X}_r(k)$. The BCE loss optimises the performance predictor's parameters and gradients from this loss pass through the VAE's encoder similarly to *VAE-Loco* [14]. It is found in *VAE-Loco* that backpropagating the BCE's loss through the VAE's encoder structures the latent space. *Gaitor's* latent-space structure is inspected in Sec. VI-C. The ELBO loss used here takes the form of

$$\mathcal{L}_{\text{ELBO}} = \text{MSE}(\mathbf{X}_r^+(k), \hat{\mathbf{X}}_r^+(k)) + \beta D_{\text{KL}}[q(\mathbf{z}|\mathbf{X}_r(k))||p(\mathbf{z})], \tag{3}$$

and is summed with the BCE loss creating

$$\mathcal{L}_{\text{VAE}} = \mathcal{L}_{\text{ELBO}} + \gamma\text{BCE}(\mathbf{S}^+(k), \hat{\mathbf{S}}^+(k)). \tag{4}$$

The VAE is optimised using GECO [26] to tune β during training. Simultaneously, the terrain encoder's weights are updated using the MSE loss between predicted $\hat{\mathbf{X}}_g(k)$ and the ground-truth data $\mathbf{X}_g(k)$ and gradients from the VAE decoder's MSE loss. This creates the terrain autoencoder loss

$$\mathcal{L}_{\text{TER}} = \text{MSE}(\mathbf{X}_r^+(k), \hat{\mathbf{X}}_r^+(k)) + \text{MSE}(\mathbf{X}_g(k), \hat{\mathbf{X}}_g(k)). \tag{5}$$

The terrain decoder is only used for training purposes and is not required for deployment.

B. Training the Planner

The planner in the latent space predicts the trajectory which is decoded to create the locomotion trajectory. In general, the planner can be trained using any framework such as an online or offline RL method, or a supervised learning approach. In this work, a behavioural cloning method is used to train the planner for simplicity. Expert trajectories are encoded into the VAE producing latent trajectories, $\mathbf{z}_0^*, ..., \mathbf{z}_D^*$, where D is the number of time steps in the encoded episode. Only the dimensions z_0 and z_1 are retained to create the dataset for the planner and are converted to polar coordinates. Thus, $z_0^*(0), ..., z_0^*(D)$ and $z_1^*(0), ..., z_1^*(D)$ are converted to radii $R^*(0), ..., R^*(D)$, and phases $\phi(0), ..., \phi(D)$. The relationships between gait phase, radius and latent trajectories are

$$\phi(k) = \text{atan2}(z_0^*(k), z_1^*(k)) \tag{6}$$

$$R^*(k) = \sqrt{{z_0^*}^2(k) + {z_1^*}^2(k)} \tag{7}$$

The planner predicts the radius $R(k)$ given the gait phase $\phi(k)$ and the terrain encoding $\mathbf{z}_g(k)$ as inputs. The total loss is the sum between the MSE to reconstruct the target radius $R^*(k)$ and the cross-entropy loss between the prediction r_c and the label r^*:

$$\mathcal{L}_{\text{PLAN}} = \text{MSE}(R(k), R^*(k)) - \sum_{c=0}^{C-1} \log\left(\frac{\exp(r_c)}{\sum_{i=0}^{I-1}\exp(r_c)} r^*\right). \tag{8}$$

V. Deployment of Gaitor

The VAE, AE and its planner form a trajectory optimiser in a closed-loop controller. During deployment, the robot and terrain encodings are estimated using the VAE's and AE's encoders respectively. The operator chooses the base velocity action $\mathbf{a}_k$, the desired gait g_k and the gait speed $\Delta\phi_k$. The gait speed $\Delta\phi_k$ controls the robot's cadence (number of footfalls per minute) and updates the gait phase every time step as

$$\phi(k) = \phi(k-1) + \Delta\phi_k. \tag{9}$$

The latent-space planner uses the updated phase and the terrain encoding to predict the latent radius. This is used to overwrite dimensions z_0 and z_1 in the robot encoding $\mathbf{z}_r(k)$ as per Eq. (1). The updated encoding $\tilde{\mathbf{z}}(k)$ is formed by concatenating the updated robot encoding, the terrain encoding, action and gait label. It is then decoded using ψ_{dec} and ψ_{pp}. The predicted joint angles, torques and future base-pose are extracted from $\hat{\mathbf{X}}^+(k)$ and are sent to a whole-body controller (WBC) [27]. During our deployment, we discovered a phase-lead is required for best base-orientation tracking. Therefore, we set the robot's base pitch to the average of the control pitch θ_c.

The entire approach operates on the CPU of the ANYmal C in a real-time control loop. There is a strict time budget of $2.5\,\text{ms}$ for $400\,\text{Hz}$ control. Therefore, we deploy *Gaitor* using a bespoke C++ implementation and vectorised code.

VI. Experimental Results

Our evaluation of *Gaitor* is designed to address the following guiding questions: 1) What does the structure of the latent-space look like for multiple gaits and how do transitions occur? 2) How does the latent-space structure and the generated trajectory adjust to changes in terrain? 3) How do the capabilities of our method compare to the current state-of-the-art?

In the following sections we first describe the generation of the training dataset and detail hyper-parameter settings used for the experiments before addressing each of these questions.

A. Dataset Generation

Gaitor is effectively trained using a narrow set of skill demonstrations. The data used to train all components are generated using *RLOC* [17]. This employs a vision-based RL footstep planner and uses *Dynamic Gaits* [27] to solve for the task-space trajectories. The inputs to the controller are the desired base heading $\mathbf{a}_k$ and the raw depth map provided by the robot. We generate roughly $30\,\text{min}$ of data of the robot traversing pallets of varying heights up to $12.5\,\text{cm}$ in both trot and crawl. The pace gait is unable to traverse uneven terrain so data for pace are gathered on flat ground. Our implementation of *Dynamic Gaits* is unable to pace stably at speeds greater than $0.1\,\text{m/s}$ in simulation and is unstable at all speeds on the real robot. Therefore, we do not deploy any pace gaits.

B. Architecture Hyper-parameters

The VAE's encoder and decoder frequencies are set to $50\,\text{Hz}$ and $400\,\text{Hz}$. The input history contains 80 timesteps, whilst the output predicts a small 20 step trajectory. All networks are 256 units wide and three layers deep with the robot latent-space and the terrain latent-space both of dimensionality 10. The operator controlled base-heading action size is three units and the gait label is one unit. During terrain traversal, we utilise a desired swing-duration of $0.36\,\text{s}$.

C. The Latent-Space Structure Results in Gait Switching

Gaitor's latent-space structure is investigated to understand how the space represents multiple gait types. Next, the space is explored revealing that continuous gait transitions are attainable using *Gaitor*. This analysis provides insights into how *Gaitor* represents shared information across gaits in the latent space and how transitions between gaits arise naturally as a result of the latent-space structure.

Inspecting the Latent-Space Structure: Inspired by *VAE-Loco*, we inspect the structure of the latent space. This reveals

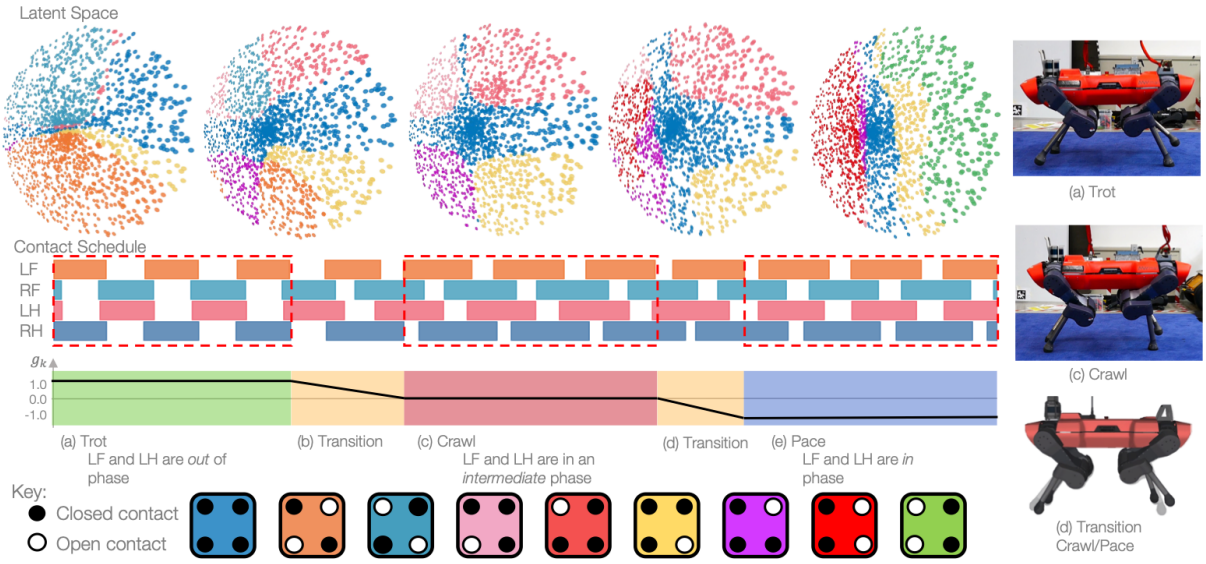


Fig. 4: The latent-space structure encodes the contact schedule of the different gaits in distinct clusters (coloured segments). The same 2D slice of latent-space is plotted in all figures, and as the gait slider moves, the clustering of space changes smoothly. We include a key in the bottom row to map each coloured point in latent space to the stance phases of the robot. The phase between the front and rear legs shifts from out-of-phase for trot to in phase for pace via an intermediary phase, producing the crawl gait. The change in the phase is reflected in the structure of the latent-space: for trot the full-stance phase in blue is aligned horizontally. The full-stance phase rotates to the vertical for pace via the crawl gait structure. The trot, crawl and a dynamic crawl/pace hybrid gait are executed on the real robot in a single run lasting 50 s. The expert pace gait is unstable on the real robot so we do not deploy any pace gaits on the real system.

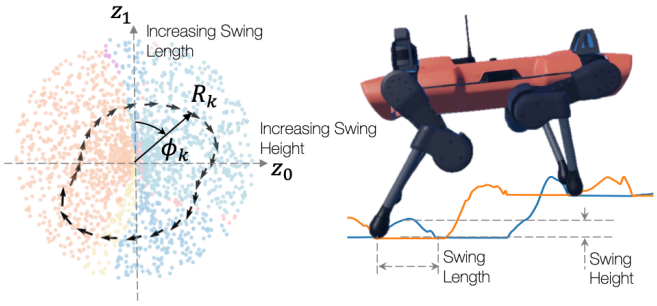


Fig. 5: The latent space is disentangled and structured such that displacements in the horizontal z_0 axis increase the robot's step height, whereas movements in the vertical axis z_1 correlate to the robot's swing length.

that *Gaitor's* latent space is both structured and disentangled. In particular, the footswing lengths and heights are encoded into two orthogonal dimensions of the latent space, meaning that these gait characteristics can be controlled independently using trajectories in these dimensions. Furthermore, analysis reveals that these two latent-space dimensions (z_0 and z_1) have the lowest variance. Fig. 5 shows points in the 2D plane made up by z_0 and z_1. The points in this plane are colour coded by contact state, where orange corresponds to left front and right hind. Teal is the opposite contact state (right front and left hind), and blue represents all four feet in contact. The latent-space structure and its relationship to the contact states and schedules for multiple gaits are explored in the following sub-sections.

Gait Switching: User-controlled and continuous gait switching from trot to crawl to a dynamic crawl/pace hybrid is observed when *Gaitor* is deployed on the real robot. The dynamic crawl/pace gait is characterised by a contact schedule where the hind foot makes contact as the front foot breaks contact simultaneously. To analyse the gait switching mechanism, latent dimensions z_0 and z_1 are plotted for a variety of gait types in Fig. 4. In Fig. 4, every point in latent space is colour-coded to denote which feet are in contact with the terrain, and Fig. 4 includes a key in the bottom row. As the gait label is toggled from 1.0 (trot) to -1.0 (pace), the latent-space structure changes in concert with the contact schedule shown in the row below.

Gait Correlations: It is observed that *Gaitor* produces gait switching in a particular order: trot to crawl to pace and the reverse only. To investigate why gait transitions occur in this order, we inspect the latent-space structure for each gait and observe the phase relationships between legs during transition. During trot, the left front (LF) and left hind (LH) are out-of-phase with one another in the contact schedule (see row 2 of Fig. 4): the LF is in contact as the LH is in swing. However during pace, the LF and LH are in phase as these feet are swinging together. As the operator demands a gait transition from trot to pace or pace to trot, the crawl gait representation and contact schedule emerges at the midpoint of the transition. This is because the phase between the LF and LH is midway between in phase and out-of-phase for the crawl gait: only one foot swings at once during crawl. *Gaitor* is able to infer this relationship and uses it to represent the multiple

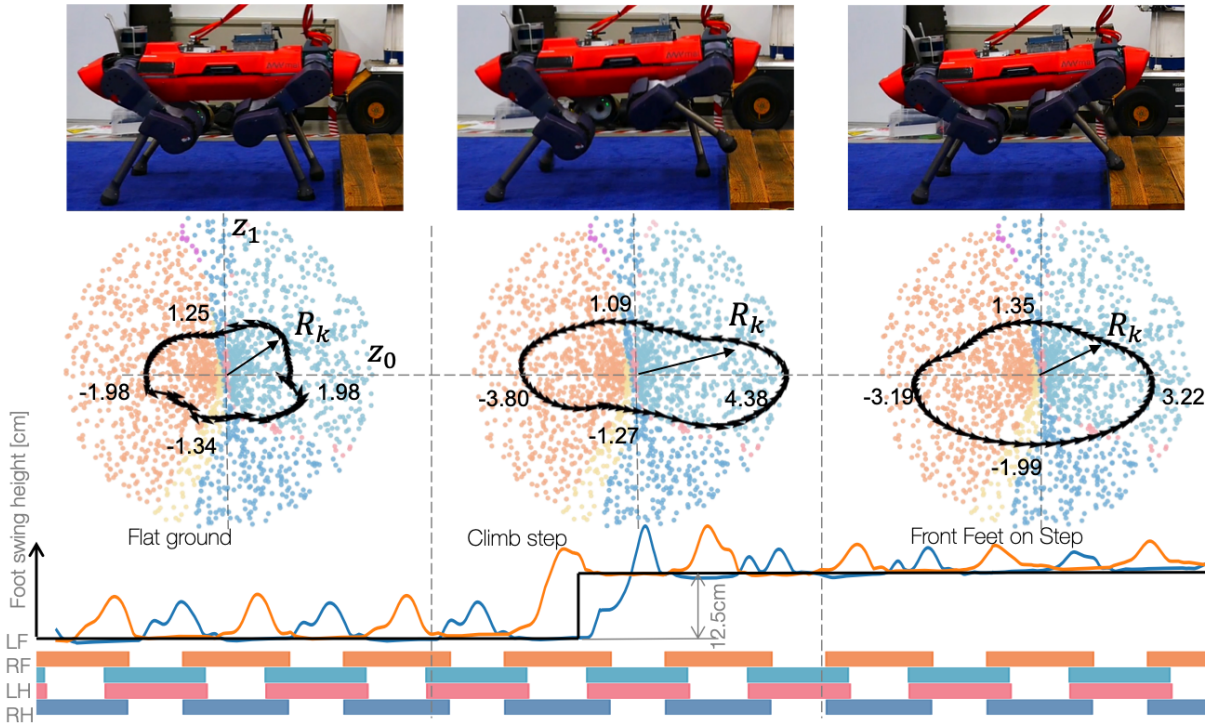


Fig. 6: *Gaitor* deployed on the robot climbing a 12.5 cm platform. The planner and latent-space structure act in concert to alter the locomotion trajectories in response to the terrain. The first phase is flat ground where the planner produces a circular ellipse. When the robot steps onto the platform, the planner radius increases dramatically along the horizontal axis, but is reduced in the vertical axis compared to flat ground operation. This results in shorter footstep heights, but longer swing distances. Once both front feet are in contact with the elevated terrain, the latent-space trajectory returns to a broadly circular path.

different gaits in the representation efficiently. Furthermore, *Gaitor* finds intermediary latent-space structures to represent the contact schedules of the transitionary gaits e.g. trot/crawl or crawl/pace, see panel (b) and (d) in Fig. 4 and these produce smooth and continuous gait transitions. The ordering of the gaits in latent space using their in-phase relationships is similar to the analytical analysis found in [28]. The model used for quadruped kinematics in [28] consists of a pair of bipeds linked by a rigid body. Gaits are generated by varying the phases between the front and rear bipeds in a fashion similar to what here is *inferred* by *Gaitor*.

Feasibility Metrics: The quality of the gaits produced by *Gaitor* is measured using the joint-space error between input and output of the whole-body controller (WBC) measured over a 5 s window sampled at 400 Hz. The WBC is an optimisation-based tracking controller and is able to adjust the VAE's output to satisfy its dynamics constraints outlined in [29]. The root-mean-squared error (RMSE) is used as a measure of how optimal the VAE's trajectories are with respect to the dynamics of the WBC. The RMSE for the trot, crawl and dynamic pace/crawl gaits are reported in Table I. In all cases the tracking error is low (less than 6 deg), and is larger for the crawl and dynamic crawl gaits. The larger joint RMSE results from a small discrepancy between the robot's actual and desired heights when operating in the crawl gaits.

D. Latent-Space Evolution During Terrain Traversal

Perceptive terrain-traversal is demonstrated using the trot gait to successfully climb a 12.5 cm step. During the climb phase, the nominal swing heights and lengths are observed to adjust in response to the terrain. In particular, the robot places its feet in a robust position further away from the edge of the step. To investigate how *Gaitor* produces this behaviour, the learnt planner's trajectories in latent space are visualised and how the gait characteristics are encoded into the space is inspected. Finally, the feasibility of the *Gaitor's* trajectories is compared to that of the dataset.

Latent-Space Trajectory In Response To Terrain: To investigate the planner's response to the terrain, we plot the latent-space trajectory during three distinct phases of climbing the terrain in Fig 6. The first phase is nominal locomotion on flat ground (phase I), the second is a short transient (phase II), where the front feet step up onto the palette and the final is a new steady state (phase III), where the robot has its front feet on the terrain and its rear feet on the ground.

On flat ground the (phase I), the latent-space trajectory is broadly symmetrical and *Gaitor* produces locomotion with footswing heights and lengths of (8.30 ± 0.58) cm and (10.40 ± 0.53) cm respectively. During the climb, the planner more than doubles the nominal horizontal displacement of the latent-space trajectory from 1.98 units to 4.38 units, see the second row of Fig. 6. This results in a large increase in the swing length to (13.90 ± 1.65) cm, but only a modest increase in

	Dynamic Gaits (Trot)	Trot (Flat Ground)	Trot (Climb Phase)	Crawl	Dynamic Crawl/Pace
RMSE [rad]	0.021	0.012	0.013	0.058	0.098
Swing Height [cm]	8.5 ± 0.40	8.3 ± 0.58	9.68 ± 4.15	5.42 ± 0.65	6.22 ± 0.31
Swing Length [cm]	14.6 ± 3.99	10.4 ± 0.53	13.9 ± 1.65	-	-

TABLE I: The feasibility and gait characteristics for *Gaitor* and *Dynamic Gaits*, which is used to generate the training data. *Gaitor's* gait characteristics and tracking root-mean-squared error (RMSE) values are recorded during four distinct modes of operation: trot on flat ground, trot whilst climbing the terrain, crawl gait, and the dynamic crawl/pace gait. The RMSE values are measured between the joint-space trajectories of the *Gaitor* or *Dynamic Gaits* and the whole-body controller's (WBC) joint trajectories during a 5 s interval. The WBC uses a centroidal dynamics model to optimise the torque values. The lower the RMSE, the closer the VAE trajectories are to optimality under the assumptions of the WBC. All measurements are generated from experimental logs from real-robot experiments. The swing lengths are not measured for crawl and dynamic crawl as the robot is operated on the spot during these runs. Both trot data come from the terrain climb experiment, see Sec. VI-D.

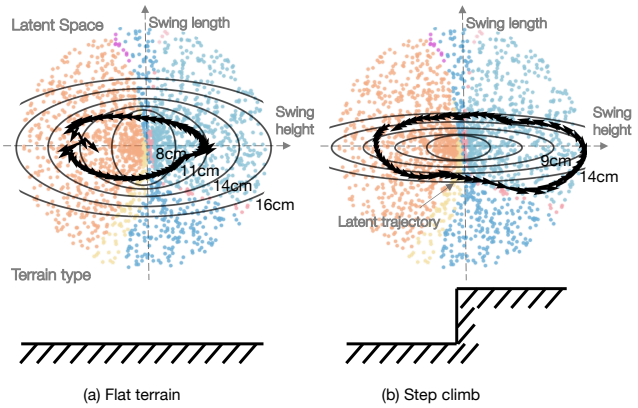


Fig. 7: Level sets are plotted in latent space where isocontours decode to equal swing heights and lengths. The latent-space trajectories and isocontours are plotted during flat-ground operation (a) and as the robot approaches the step (b). The latent-space structure prioritises swing length during climb.

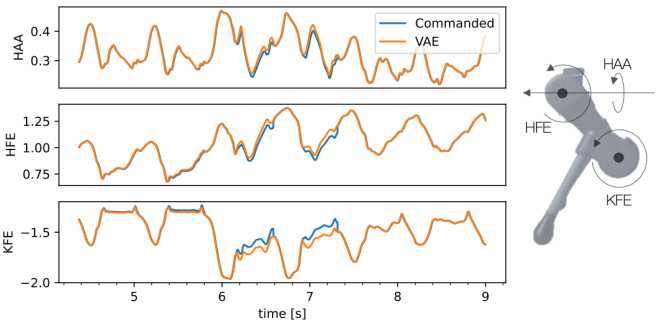


Fig. 8: The joint-space trajectory in radians for the right front leg as it climbs terrain. The VAE's joint space trajectory is tracked well by the WBC's commanded values meaning that the VAE's output is optimal with respect to the WBC's dynamics formulation. A leg is plotted next to the tracking plots and annotated with the joint names.

swing height of $(9.68 \pm 4.15)\,\mathrm{cm}$ in the robot's base frame. As mentioned, this increases the robustness of the foothold selection as this footfall location is further from the edge of the step. This beneficial response of elongating only the footswing length is investigated in the next paragraph. In phase III, the latent trajectory returns to a more symmetrical shape similar to phase I. This is expected once the front feet have climbed the terrain, the locomotion is similar to flat ground operation.

Latent-Space Structure During Climb: During the climb (phase II), *Gaitor* increases the footswing length significantly more than the footstep height. This is investigated by inspecting how the encoding of the gait characteristics vary as the terrain changes. To visualise this, level sets in latent space are plotted in Fig. 7. Here, isocontours decode to locomotion trajectories with equal footswing height and length relative to the robot's base frame. For example, decoding the inner most contour in Fig. 7 (a) results in locomotion with a footswing height and length of $8\,\mathrm{cm}$ (the swing length is measured relative to the base frame). For both flat-ground operation and the climb phase, the latent-space trajectories follow the isocontours. Crucially, as the edge of the pallet is encoded, the latent-space structure adapts such that a small displacement in the vertical axis of the latent space (z_1 dimension) decode to much longer footsteps. This is reflected in the planner's trajectory which adjusts similarly to the latent-space structure to follow the deformation of the isocontours. If the flat-ground latent-trajectory is used during terrain traversal, the robot does not lift its feet high enough over the step and the feet collide with the step. However, *Gaitor's* planner reacts to the oncoming terrain and produces a large footswing height with clearance of $4.6\,\mathrm{cm}$ above the edge of the pallet. In conclusion, the planner is necessary during terrain traversal in order to fully exploit the latent-space structure, which usefully deforms as the robot approaches terrain.

Feasibility And Comparison To Expert: To quantify the feasibility of the *Gaitor's* perceptive locomotion, we compute the joint-space error between the input and the output of the WBC. As before, root-mean-squared error (RMSE) is used to measure the optimality of the VAE's trajectories with respect to the WBC's centroidal dynamics [30]. The lower the RMSE value, the closer the VAE's trajectory is to satisfying the WBC's optimal solution. A 5 s window during flat ground operation and the climb phase (middle panel in Fig. 6) is compared with the dataset RMSE in Table I. The dataset uses the *Dynamic Gaits* [27] formulation, which is designed to operate with the WBC. The RMSE for the VAE during both flat-ground and climb phases is lower than that of *Dynamic Gaits* (see Table I) showing that *Gaitor's* trajectories are

Approach	RL vs Representation Learning	Discovers Intermediate Skills	Explicit vs Implicit Representation
Caluwaerts et al. [11]	RL (Bottom-Up)	No	Implicit
Hoeller et al. [6]	RL (Top-Down)	No	Implicit
Yang et al. [13]	RL + MPC	No	Explicit Inductive Bias
Gaitor	Rep. Learn	Yes	Explicit Inferred

TABLE II: The characteristics and differentiating features of *Gaitor* are compared to a range of locomotion approaches. Bottom-up RL methods such as [11] tend to employ black-box models to learn a single distribution over multiple locomotion skills. Top-down RL refers to hierarchical approaches [6] to locomotion, which use a high-level planner to select a particular skill from a series of sub-policies optimised for one particular gait. RL and MPC methods in combination [13] tend to use RL to solve for trajectories constrained by difficult to model parts of the robot's dynamics and use model-predictive control (MPC) to generate task-space trajectories. In contrast, *Gaitor* learns a unified representation across multiple gaits and in doing so is uniquely able to automatically discover new intermediate gaits and this ability is compared between approaches in the second column. The third column compares each method's representation for locomotion. The first two RL methods have an internal implicit representation, whilst the Yang et al. [13] and *Gaitor* use the gait phase as explicit representation. The explicit representation of gait phase naturally leads to sharing of gait-specific correlations across skills and facilitates gait transitions as described in Sec. VI-C. However, Yang et al., use the gait phase as an inductive bias and a decision variable during training and execution. In contrast, *Gaitor* infers the gait phase across multiple gaits from expert examples of the trajectories.

feasible for the WBC to track with little to no adjustments. In contrast, the WBC makes minor adjustments to the trajectories from *Dynamic Gaits*, which accounts for the larger RSME values. In addition, the VAE's joint trajectory, which is the input to the WBC and the WBC's output (the commanded values) for the right-front leg are plotted in Fig. 8. The tracking error is insignificant (less than one degree) except during the initial touch down phase. This small increase in RMSE is attributed to vision error. The height of the step is slightly underestimated by the robot's perception module, and does not affect the stability of the locomotion. After a subsequent footswing, the VAE adjusts and the tracking error remains extremely low.

E. Gaitor's Capabilities in Relation to Other Methods

The characteristics of Gaitor and key differentiators are compared to current methods in Table II. As shown in Sec. VI-C, *Gaitor* is able to learn a unified representation across multiple gaits. In the same section, it is shown that this naturally yields continuous and smooth transitions between gaits in an interpretable way. Top-down hierarchical methods such as [6] use multiple independent policies and learn a planner to select the correct controller as required. This ignores correlations between the gait types and continous gait switching via intermediate gaits is impossible. Caluwaerts el al. [11] learn a multi-modal distribution policy over gait skills using a large transformer model. This does not produce an explicit representation, but an implicit one internal to the transformer. In contrast, *Gaitor* creates a dense and explicit latent-representation and once trained this representation facilitates the automatic discovery of useful intermediary gaits leading to smooth and continuous gait transitions in an interpretable way as seen in Sec. VI-C and Sec. VI-D. In particular, it is demonstrated in Sec. VI-C that *Gaitor* infers the correlations, namely the phase relationships, between the gait types and uses these to structure the latent representation. In essence, *Gaitor* has inferred an inductive bias from expert data that is typically used in the formulations of other methods. For example, [12], [13] uses the gait phase as an explicit control parameter to generate blind gait transitions. In contrast, *Gaitor* infers the gait phase relationship across gaits and builds this directly into its locomotion representation.

VII. Limitations and Future Work

The focus of this paper is exploring the opportunities that learning a unified representation for locomotion across gaits provide and not pushing the state-of-the-art in one aspect of locomotion. As such, we have shown that *Gaitor* learns a representation capable of discovering continuous transitions between gaits and conditioned with a rudimentary terrain-encoding generates perceptive locomotion to traverse over significant obstacles. However, a number of limitations currently remain. The primary limitation of *Gaitor* is the requirement of high-quality expert demonstrations. We posit that this is a limitation currently of all learning-from-demonstration methodologies and is lessened by the availability of quality locomotion controllers designed for specific tasks/gaits.

In future, the latent-planner could be used to control the gait phase, desired heading and gait of the robot. These parameters are currently user-controlled and as we have shown in Sec. VI-C can be readily varied on-the-fly. A more sophisticated planner could be able to tackle more challenging locomotion tasks.

VIII. Conclusions

In this paper, we explored the opportunities afforded by learning a single, disentangled representation for locomotion across gaits to encode correlations between separate skills. In doing so, we discovered that the learnt latent-space embeds not only each individual gait into the space, but also orders each gait representation such that continuous transitions via novel intermediary contact schedules are discovered from traversing the learnt latent-space. In addition, *Gaitor* incorporates a rudimentary terrain encoding which adapts the structure of the latent space and in conjunction with a learnt planner adjust the robot's gait characteristics. During our evaluation of *Gaitor*, we investigate the gait transitions and terrain

traversal. This reveals that the latent-space structure changes to represent different gaits during transition and adapts the gait characteristics in response to the terrain encoding. This change in gait characteristics is beneficial during the initial climb phase produces more robust footholds further away from the edge of the step. In conclusion, we show that learning a single unified latent representation for locomotion across gaits results in a methodology capable of continuous gait transitions and perceptive locomotion to traverse significant obstacles.

REFERENCES

[1] C. Mastalli, R. Budhiraja, W. Merkt, G. Saurel, B. Hammoud, M. Naveau, J. Carpentier, L. Righetti, S. Vijayakumar, and N. Mansard, "Crocoddyl: An efficient and versatile framework for multi-contact optimal control," in *IEEE Int. Conf. Rob. Autom. (ICRA)*, 2020.

[2] C. Mastalli, W. Merkt, J. Marti-Saumell, H. Ferrolho, J. Sola, N. Mansard, and S. Vijayakumar, "A feasibility-driven approach to control-limited ddp," *Autonomous Robots*, 2022.

[3] R. Grandia, A. J. Taylor, A. D. Ames, and M. Hutter, "Multi-layered safety for legged robots via control barrier functions and model predictive control," *CoRR*, 2020.

[4] F. Jenelten, T. Miki, A. E. Vijayan, M. Bjelonic, and M. Hutter, "Perceptive locomotion in rough terrain – online foothold optimization," *IEEE Robot. Automat. Lett. (RA-L)*, vol. 5, no. 4, pp. 5370–5376, 2020.

[5] O. Melon, R. Orsolino, D. Surovik, M. Geisert, I. Havoutis, and M. Fallon, "Receding-horizon perceptive trajectory optimization for dynamic legged locomotion with learned initialization," in *IEEE Int. Conf. Rob. Autom. (ICRA)*, 2021.

[6] D. Hoeller, N. Rudin, D. Sako, and M. Hutter, "Anymal parkour: Learning agile navigation for quadrupedal robots," 2023.

[7] N. Rudin, D. Hoeller, M. Bjelonic, and M. Hutter, "Advanced skills by learning locomotion and local navigation end-to-end," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 2497–2503.

[8] S. Gangapurwala, A. Mitchell, and I. Havoutis, "Guided constrained policy optimization for dynamic quadrupedal robot locomotion," *IEEE Robot. Automat. Lett. (RA-L)*, vol. 5, no. 2, pp. 3642–3649, 2020.

[9] J. Hwangbo, J. Lee, A. Dosovitskiy, D. Bellicoso, V. Tsounis, V. Koltun, and M. Hutter, "Learning agile and dynamic motor skills for legged robots," *Science Robotics*, vol. 4, no. 26, 2019.

[10] N. Rudin, D. Hoeller, P. Reist, and M. Hutter, "Learning to walk in minutes using massively parallel deep reinforcement learning," 2022.

[11] K. Caluwaerts, A. Iscen, J. C. Kew, W. Yu, T. Zhang, D. Freeman, K.-H. Lee, L. Lee, S. Saliceti, V. Zhuang, N. Batchelor, S. Bohez, F. Casarini, J. E. Chen, O. Cortes, E. Coumans, A. Dostmohamed, G. Dulac-Arnold, A. Escontrela, E. Frey, R. Hafner, D. Jain, B. Jyenis, Y. Kuang, E. Lee, L. Luu, O. Nachum, K. Oslund, J. Powell, D. Reyes, F. Romano, F. Sadeghi, R. Sloat, B. Tabanpour, D. Zheng, M. Neunert, R. Hadsell, N. Heess, F. Nori, J. Seto, C. Parada, V. Sindhwani, V. Vanhoucke, and J. Tan, "Barkour: Benchmarking animal-level agility with quadruped robots," 2023.

[12] J. Wu, Y. Xue, and C. Qi, "Learning multiple gaits within latent space for quadruped robots," 2023.

[13] Y. Yang, T. Zhang, E. Coumans, J. Tan, and B. Boots, "Fast and efficient locomotion via learned gait transitions," in *Conf. on Rob. Learn. (CoRL)*, 2021.

[14] A. L. Mitchell, W. X. Merkt, M. Geisert, S. Gangapurwala, M. Engelcke, O. P. Jones, I. Havoutis, and I. Posner, "Vae-loco: Versatile quadruped locomotion by learning a disentangled gait representation," *IEEE Transactions on Robotics*, pp. 1–16, 2023.

[15] A. L. Mitchell, W. Merkt, M. Geisert, S. Gangapurwala, M. Engelcke, O. P. Jones, I. Havoutis, and I. Posner, "Next steps: Learning a disentangled gait representation for versatile quadruped locomotion," in *IEEE Int. Conf. Rob. Autom. (ICRA)*, 2022, pp. 10 564–10 570.

[16] M. Hutter, C. Gehring, D. Jud, A. Lauber, C. D. Bellicoso, V. Tsounis, J. Hwangbo, K. Bodie, P. Fankhauser, M. Bloesch, R. Diethelm, S. Bachmann, A. Melzer, and M. Hoepflinger, "ANYmal - a highly mobile and dynamic quadrupedal robot," in *IEEE/RSJ Int. Conf. Intell. Rob. Sys. (IROS)*, 2016.

[17] S. Gangapurwala, M. Geisert, R. Orsolino, M. Fallon, and I. Havoutis, "Rloc: Terrain-aware legged locomotion using reinforcement learning and optimal control," *IEEE Transactions on Robotics*, vol. 38, no. 5, pp. 2908–2927, 2022.

[18] T. Li, R. Calandra, D. Pathak, Y. Tian, F. Meier, and A. Rai, "Planning in learned latent action spaces for generalizable legged locomotion," *CoRR*, vol. abs/2008.11867, 2020.

[19] D. Kingma and M. Welling, "Auto-encoding variational bayes," in *Int. Conf. on Learn. Repr. (ICLR)*, 2014.

[20] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," in *Int. Conf. on Mach. Learn. (ICML)*, 2014.

[21] A. L. Mitchell, M. Engelcke, O. Parker Jones, D. Surovik, S. Gangapurwala, O. Melon, I. Havoutis, and I. Posner, "First steps: Latent-space control with semantic constraints for quadruped locomotion," in *IEEE/RSJ Int. Conf. Intell. Rob. Sys. (IROS)*, 2020, pp. 5343–5350.

[22] J. Solà, J. Deray, and D. Atchuthan, "A micro lie theory for state estimation in robotics," 2021.

[23] C. Mastalli, W. Merkt, G. Xin, J. Shim, M. Mistry, I. Havoutis, and S. Vijayakumar, "Agile maneuvers in legged robots: a predictive control approach," 2022. [Online]. Available: https://arxiv.org/abs/2203.07554

[24] M. Mattamala, N. Chebrolu, and M. Fallon, "An efficient locally reactive controller for safe navigation in visual teach and repeat missions," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2353–2360, apr 2022.

[25] N. Ruiz, E. Chong, and J. M. Rehg, "Fine-grained head pose estimation without keypoints," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2018, pp. 2155–215 509.

[26] D. J. Rezende and F. Viola, "Taming VAEs," 2018.

[27] C. D. Bellicoso, F. Jenelten, C. Gehring, and M. Hutter, "Dynamic locomotion through online nonlinear motion optimization for quadrupedal robots," *IEEE Robot. Automat. Lett. (RA-L)*, vol. 3, no. 3, pp. 2261–2268, 2018.

[28] C. Tiseo, S. Vijayakumar, and M. Mistry, "Analytic model for quadruped locomotion task-space planning," in *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2019, pp. 5301–5304.

[29] C. Dario Bellicoso, F. Jenelten, P. Fankhauser, C. Gehring, J. Hwangbo, and M. Hutter, "Dynamic locomotion and whole-body control for quadrupedal robots," in *IEEE/RSJ Int. Conf. Intell. Rob. Sys. (IROS)*, 2017, pp. 3359–3365.

[30] D. Orin, A. Goswami, and S.-H. Lee, "Centroidal dynamics of a humanoid robot," *Autonomous Robots*, vol. 35, 10 2013.

Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor **(only required where there isn't already a statement of contribution within the paper itself).**

Title of Paper	Gaitor: Learning a Unified Representation Across Gaits for Real-World Quadruped Locomotion
Publication Status	Submitted
Publication Details	Alexander L. Mitchell, Wolfgang Merkt, Aristotelis Papatheodorou, Ioannis Havoutis, Ingmar Posner Submitted to CoRL 2024

Student Confirmation

Student Name:	Alexander Luis Mitchell		
Contribution to the Paper	Led efforts on: • Proposed work and created the team • Created, developed and implemented methodologies • Ran experiments • Created all figures and completed all writing Co-led efforts: • Experiment support		
Signature		Date	15/JUN/2024

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title: Professor Ioannis Havoutis		
Supervisor comments These statements are correct.		
Signature Ioannis Havoutis	Date	15/JUN/2024

6

From Primates to Robots: Brain-like Latent Dynamics Emerge in Robot Systems during Walking and Reaching

Inspired by the robot-specific representations of locomotion, we posit that there exist such representations in nature for locomotion. This chapter introduces *Primates to Robots* which discusses the similarities between the oscillatory dynamics found in the learnt latent representation for robot locomotion to those found in the motor cortices of monkeys. Churchland et al. [25] discover rotational signals during locomotion and reaching in the motor cortices of primates. To do this, Churland et al. use a novel principal component analysis (PCA) method called jPCA to project the firing of neural populations into two-dimensional slices.

In this chapter, we show that the learnt oscillatory trajectories in the robot's latent space are similar to signals in the monkey's brain. In particular, the locomotion latent trajectories are characterised by a primary drive signal and a secondary oscillation $\pi/2$ out of phase with the primary one. Furthermore, the period of these signals is similar to the locomotion motor cortex trajectories. Inspired by the similarities between the neural populations in the motor cortex and those in the robot quadruped during locomotion, we investigate the properties of a manipulation-specific latent representation. In particular, we investigate whether explicitly rotational dynamics are capable of solving real-world

reaching problems for the robot manipulator. This is in contrast to other prior work, which use implicitly rotational signals to solve planning and control problems [45]. Our analysis of the manipulator latent space reveals it also has a useful structure such that longitudinal and lateral movements of the end-effector map to individual dimensions of the space. This result neatly mirrors the disentangled representation in the locomotion domain, where the gait characteristics are disentangled in the latent space. As a result, we show that we can solve reaching tasks on the real robot using pairs of oscillatory signals similar to those found in the motor cortices of primates during reaching.

We investigate the nature of the correlations captured in the latent space and apply lesions to specific parts of the locomotion VAE. In particular, we lesion parts of the decoder as we reconstruct oscillatory signals in latent space. The lesioning method chosen is a cascade failure, where the output of the activation layers can become and remain zero with a certain fixed probability. This results in the sudden failure meaning that the robot can no longer walk successfully. In particular, the robot is unable to lift pairs of its feet during trot. Further analysis reveals that it is more likely for particular pairs of feet to fail rather than the alternative pairs. We also re-encode the resulting motions into latent space and inspect the trajectories. We find that on the point of failure, the trajectories are erratic but quickly converge to asymmetrical, but stable limit cycles. This hints that though the failure occurs suddenly, it is a stable failure mode.

We include both an extended abstract and the journal versions of the work in this chapter. The extended abstract works as a succinct introduction to this work. The abstract was accepted and presented at the Conference of Cognitive Neuroscience (CCN) 2023 and the journal has been published in Nature Scientific Reports as *Oscillating Latent Dynamics In Robot Systems During Walking And Reaching*.

From Primates to Robots:
Emerging Oscillatory Latent-Space Dynamics for Sensorimotor Control

Alexander Luis Mitchell (mitch@robots.ox.ac.uk)
A2I & DRS, Oxford Robotics Institute, 23 Banbury Rd
Oxford, OX2 6NN, United Kingdom

Oiwi Parker Jones (oiwi@robots.ox.ac.uk)
A2I, Oxford Robotics Institute, 23 Banbury Rd
Oxford, OX2 6NN, United Kingdom

Jun Yamada (jyamada@oxfordrobotics.institute)
A2I, Oxford Robotics Institute, 23 Banbury Rd
Oxford, OX2 6NN, United Kingdom

Wolfgang Merkt (wolfgang@robots.ox.ac.uk)
DRS, Oxford Robotics Institute, 23 Banbury Rd
Oxford, OX2 6NN, United Kingdom

Ioannis Havoutis (ioannis@oxfordrobotics.institute)
DRS, Oxford Robotics Institute, 23 Banbury Rd
Oxford, OX2 6NN, United Kingdom

Ingmar Posner (ingmar@robots.ox.ac.uk)
A2I, Oxford Robotics Institute, 23 Banbury Rd
Oxford, OX2 6NN, United Kingdom

Abstract

Oscillatory signals drive reaching and locomotion for both robots and primates. Recent results in neuroscience have shown that periodic signals are present in the motor cortex of primates during rhythmic tasks such as locomotion as well as during linear movements notably reaching. Our works on learning latent motion representations in robotics revealed that oscillatory latent dynamics emerge automatically from training data for quadruped locomotion as well as manipulation tasks. Inspired by these works, we recreate the locomotion latent-spaces as well as create manipulation specific versions. With the latter we show that manipulation problems can be solved using periodic signals in a suitable latent-space. We see that these trajectories are reminiscent of those seen in the motor cortex. We artificially lesion the decoder of the locomotion model to understand how the correlations are captured. This results in deformation of both the oscillatory signals in the latent space and degradation in the locomotion trajectories.

Keywords: Representation learning; machine learning; robotics; neuroscience; sensorimotor control

Introduction

Understanding the intricacies of locomotion and manipulation is a key driver for both neuroscientists and roboticists. Indeed, recent work in neuroscience has revealed that locomotion in primates is controlled by oscillations in the monkey's motor cortex (Churchland et al., 2012). These oscillations are observed in low-dimensional projections of neural populations in the motor cortex, whilst the monkey is performing rhythmic movement such as swimming and walking. As a matter of fact, these oscillations are also found in the motor cortex when the primate is moving its limbs linearly during reaching.

In conjunction with the previous results, the utilisation of generative models in the field of robotics (Mitchell et al., 2023, 2022) has revealed that robot locomotion can be represented as oscillatory signals in low-dimensional representations. In our previous work (Mitchell et al., 2023, 2022) we create a structured latent-space for locomotion utilising a deep generative model in order to control a real quadruped robot. In particular, a variational auto-encoder (VAE) (Kingma & Welling, 2014; Rezende, Mohamed, & Wierstra, 2014) is utilised to create a low-dimensional representation of locomotion data (e.g. joint positions, torques, etc.) via an information bottleneck, which is the latent-space. It is found that the properties of locomotion such as cadence and footstep height are disentangled in the space and that cyclic signals injected within produce continuous locomotion.

The work in Churchland et al., 2012 broke new ground with their discovery that reaching is also periodic in the motor cortex. Similarly, the oscillations, which emerge in the latent space, are a consequence of their learnt structure. We thus posit that the cyclic signals in the motor cortex are also a result of order in the neural populations of the primate's motor cortex.

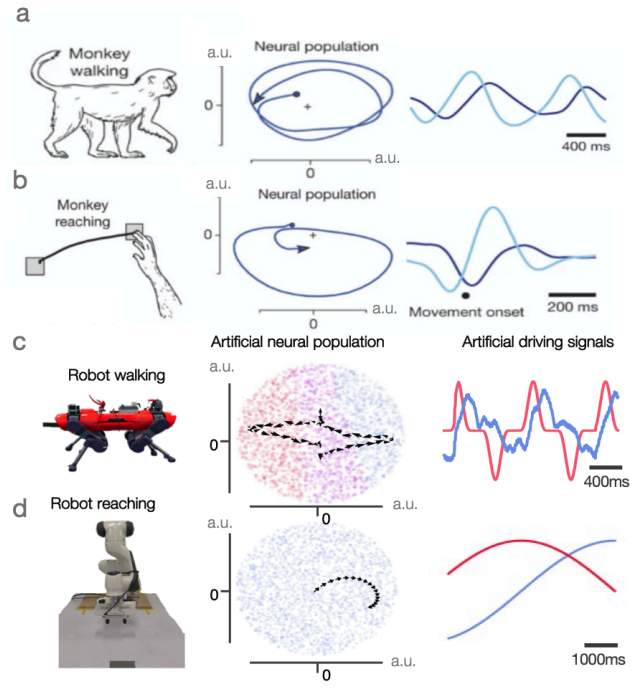


Figure 1: Oscillatory signals are found in neural populations of primate's motor cortex in both locomotion (panel (a)) and reaching (panel (b)). These two panels are reproduced with permission from Churchland et al., 2012. Quadruped locomotion can be solved via planning in a structured latent-space (panel (c)). The trajectories utilised are also cyclic. A suitably trained latent space for robot manipulation also facilitates solving reaching tasks (panel (d)).

To investigate this, we present the following paper, which is a summary of our previous work Parker Jones et al., 2022. Here, we explore the structure of the learnt latent-space for locomotion and show that key gait-specific properties emerge automatically. As a result, we are able to generate new movements unseen during training. In order to investigate if robotic manipulation is also cyclic in a learnt latent-space, we create another similar space and show that a low-dimensional representation with interpretable properties emerges. With this understanding, we inject oscillatory signals similar to those used during locomotion to solve the reaching tasks. When the latent-space signals for locomotion and manipulation are plotted and compared to those found in the motor cortex (Churchland et al., 2012) as seen in Figure 1, there is a visual similarity. Finally, we take the analogy between the latent space and the motor cortex further and artificially lesion the VAE's decoder and visualise the results. We find that the locomotion trajectories degrade in a predictable and repeatable way and that encoding the resulting locomotion reveals deformation of the cyclic signals in the latent space.

Methods

We posit that the oscillations which exist in latent space and in the motor cortex are a result of their structures. Since we are unable to artificially construct a motor cortex in the lab, we created a learnt latent-space. Therefore, we describe how the latent-space is created; how we inspect and discover the structure; and finally, how the models are lesioned in order to inspect how correlations are captured by the models.

A VAE is trained for the quadruped and for the manipulator. The inputs to the VAEs' encoders consist of a history of robot states (e.g. joint positions, torques, contact forces) and the decoder output are the predicted next states for a preview horizon. The VAEs are trained using short trajectories of both the manipulator and the quadruped operating in their environments. For the manipulator, the ELBO loss is minimised. The quadruped continually makes and breaks contact as it takes steps resulting in discontinuously changing dynamics between swing versus stance. Therefore, the latent space is constrained by learnt multi-layer perceptrons (MLP) called performance predictors (PP) that predict which feet are in contact. This adds an additional binary cross-entropy loss term to the locomotion training loss during optimisation.

The latent-space structure is investigated in two ways: Firstly, trajectories from the test dataset are encoded and the resulting trajectories inspected. Secondly, a short oscillatory trajectory is injected into latent space and decoded. The decoded trajectory is sent to a tracking controller and the robots' movements are visualised.

Finally, we artificially impair the locomotion VAE in order to investigate how the latent-space trajectories map to the resulting robot trajectories. This is achieved by applying a cascading dropout to the decoder layers. This filter either passes the output of neurons in the decoder through or zeros them. The probability of zeroing the neuron output increases with time and once the filter zeros a neuron, it continues to do so – akin to neural degradation.

Results

The learnt latent-spaces are straight-forward to inspect compared to the motor cortex. Therefore, we probe our latent-space structures to investigate the structure and compare the resulting trajectories to those observed in motor cortex. Additionally, we modulate the signals injected into the latent-space to discover how these affect the robot's movement. Finally, we impair the locomotion model and visualise how this causes the locomotion trajectories to degrade.

As seen in Figure 1, the locomotion trajectories in latent-space form a periodic cycle. We inject these trajectories into the latent space and find that the time period of the red signal in Figure 1 controls the cadence of the robot and the amplitude governs the footstep height. The blue signal is inferred from the robot states and controls the footstep length of the robot, and is $\pi/2$ out of phase with the red signal. The two signals plotted against each other form the limit cycle in Figure 1.

We repeat this experiment for manipulation and find that two sine wave oscillations where the latter is also $\pi/2$ out of phase with the first can solve reaching tasks. This resembles what Churchland et al., 2012 have found in Figure 1 (b), where we see a cyclic pattern constructed from two slightly out of phase periodic signals for both reaching and locomotion. We emphasise that this is a curious resemblance and it is not possible for us to show that the similarity is more fundamental.

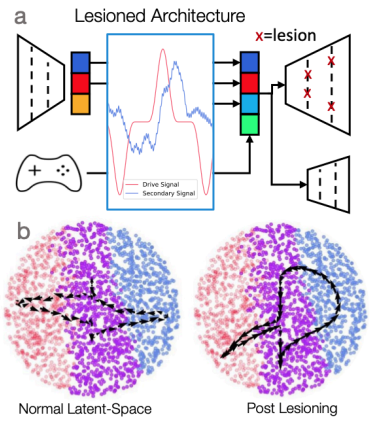


Figure 2: The VAE's decoder is lesioned and the trajectories decoded. These output trajectories are re-encoded and visualised. The latent-space trajectory has deformed after lesioning. The symmetrical structure has degraded.

Lastly, we extend the analogy between the motor cortex and the latent space by artificially impairing the locomotion decoder. This is something that should not be done to the motor cortex on both practical and ethical grounds, but is possible for our *in-silico* models. We utilise the same oscillatory drive signals as in Figure 1 (c) and decode the results. We find that two pairs of legs are dragged along the ground and no longer take swing steps. The same sets of legs fail together since the locomotion is conditioned on the trot gait where the left front and right hind move together. We also encode the resulting motion and notice how the oscillations in latent space change shape and are no longer symmetrical, see Figure 2.

Conclusion

There are visual similarities between the oscillations found in the primate's motor cortex and the robot's latent spaces. Unlike the motor cortex, we are able to interrogate the structure of the latent space. In doing so, we find that emergent properties of locomotion and the reaching workspace become embedded in the space. In the case of locomotion, cadence and footstep height are disentangled into separate orthogonal dimensions of the latent space. This facilitates the blending between dynamic trot gaits. Alternatively, for manipulation the workspace of the robot is embedded in the latent space. In essence the table top surface is mapped into the space. Finally, we are able to lesion our VAE models to observe how the latent-space trajectories degrade.

Acknowledgements

This work was supported by a UKRI/EPSRC Programme Grant [EP/V000748/1], the EPSRC grant 'Robust Legged Locomotion' [EP/S002383/1], the EPSRC CDT [EP/L015897/1], the UKRI/EPSRC RAIN [EP/R026084/1] and ORCA [EP/R026173/1] Hubs and the EU H2020 Project MEMMO (780684). It was conducted as part of ANYmal Research, a community to advance legged robotics. The authors would like to acknowledge the use of the SCAN facility.

References

Churchland, M. M., Cunningham, J. P., Kaufman, M. T., Foster, J. D., Nuyujukian, P., Ryu, S. I., & Shenoy, K. V. (2012). Neural population dynamics during reaching. *Nature*, *487*.

Kingma, D., & Welling, M. (2014). Auto-encoding variational bayes. In *Int. conf. on learn. repr. (iclr).*

Mitchell, A. L., Merkt, W., Geisert, M., Gangapurwala, S., Engelcke, M., Parker Jones, O., ... Posner, I. (2022). Next steps: Learning a disentangled gait representation for versatile quadruped locomotion. *IEEE International Conference on Robotics and Automation (ICRA)*.

Mitchell, A. L., Merkt, W. X., Geisert, M., Gangapurwala, S., Engelcke, M., Jones, O. P., ... Posner, I. (2023). Vae-loco: Versatile quadruped locomotion by learning a disentangled gait representation. *IEEE Transactions on Robotics*, 1-16.

Parker Jones, O., Mitchell, A. L., Jun, Y., Merkt, W., Geisert, M., Havoutis, I., & Posner, I. (2022). Brain-like latent dynamics emerge in robot systems during walking and reaching. *Under review*.

Rezende, D. J., Mohamed, S., & Wierstra, D. (2014). Stochastic backpropagation and approximate inference in deep generative models. In *Int. conf. on mach. learn. (icml).*

Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor **(only required where there isn't already a statement of contribution within the paper itself).**

Title of Paper	From Primates to Robots: Emerging Oscillatory Latent-Space Dynamics for Sensorimotor Control
Publication Status	Published
Publication Details	Alexander L. Mitchell, Oiwi Parker Jones, Jun Yamada, Wolfgang Merkt, Ioannis Havoutis, Ingmar Posner Published at Conference on Computational Neuroscience

Student Confirmation

Student Name:	Alexander Luis Mitchell		
Contribution to the Paper	Led following efforts: • Proposing the project and organising the team • All work on quadruped including training models and experiments • All quadruped figures and first figure • Writing all quadruped related sections Co-led efforts: • Writing the introduction		
Signature		Date	02/OCT/2023

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title: Professor Ioannis Havoutis		
Supervisor comments These statements are correct.		
Signature Ioannis Havoutis	Date	02/OCT/2023

Oscillating latent dynamics in robot systems during walking and reaching

Oiwi Parker Jones,[1*] Alexander L. Mitchell,[1,2*] Jun Yamada,[1] Wolfgang Merkt,[2] Mathieu Geisert,[2] Ioannis Havoutis,[2] Ingmar Posner[1]

[1]Applied AI Lab, [2]Dynamic Robot Systems Group,
Oxford Robotics Institute, University of Oxford
[*]Corresponding authors / equal contribution:
oiwi@robots.ox.ac.uk, mitch@robots.ox.ac.uk.

Sensorimotor control of complex, dynamic systems such as humanoids or quadrupedal robots is notoriously difficult. While artificial systems traditionally employ hierarchical optimisation approaches or black-box policies, recent results in systems neuroscience suggest that complex behaviours such as locomotion and reaching are correlated with limit cycles in the primate motor cortex. A recent result suggests that, when applied to a learned latent space, oscillating patterns of activation can be used to control locomotion in a physical robot. While reminiscent of limit cycles observed in primate motor cortex, these dynamics are unsurprising given the cyclic nature of the robot's behaviour (walking). In this preliminary investigation, we consider how a similar approach extends to a less obviously cyclic behaviour (reaching). This has been explored in prior work using computational simulations. But simulations necessarily make simplifying assumptions that do not necessarily correspond to reality, so do not trivially transfer to real robot platforms. Our primary contribution is to demonstrate that we can infer and control real robot states in a learnt representation using oscillatory dynamics during reaching tasks.

We further show that the learned latent representation encodes interpretable movements in the robot's workspace. Compared to robot locomotion, the dynamics that we observe for reaching are not fully cyclic, as they do not begin and end at the same position of latent space. However, they do begin to trace out the shape of a cycle, and, by construction, they are driven by the same underlying oscillatory mechanics.

1 Introduction

Biology has served as inspiration for artificial intelligence and robotics from the founding of these fields. Yet many engineering solutions for complex planning and control tasks make use of non-biologically plausible optimisation schemes and are customised for specific robot platforms. One reason for this divergence is that we do not fully understand, and therefore are not able to directly replicate, how brains solve complex planning and control problems in tasks such as manipulation and locomotion. Of course this does not mean that neuroscience has nothing to offer, especially as the frontiers of neuroscience are continuously expanding.

The present work takes inspiration from two recent developments, one in systems neuroscience (*2*) and another in robotics (*3*). An analogy between them is teased in Figure 1. On the robotics side, Mitchell et al. (*3*) learned two things by using a variational autoencoder (VAE) (*4, 5*) to encode the states (e.g. joint positions) of a legged robot. First, they found that different leg configurations correspond to different positions in the learned latent space. The second discovery was that the different latent states (which represent different foot configurations) are organised so that walking corresponds (roughly) to elliptical trajectories in a low-dimensional (2D) projection of the latent space. The existence of elliptical trajectories did not need to be the case: walking might have corresponded to more erratic wanderings through latent space.

The authors next showed that synthetic oscillations can be used to generate robot behaviour

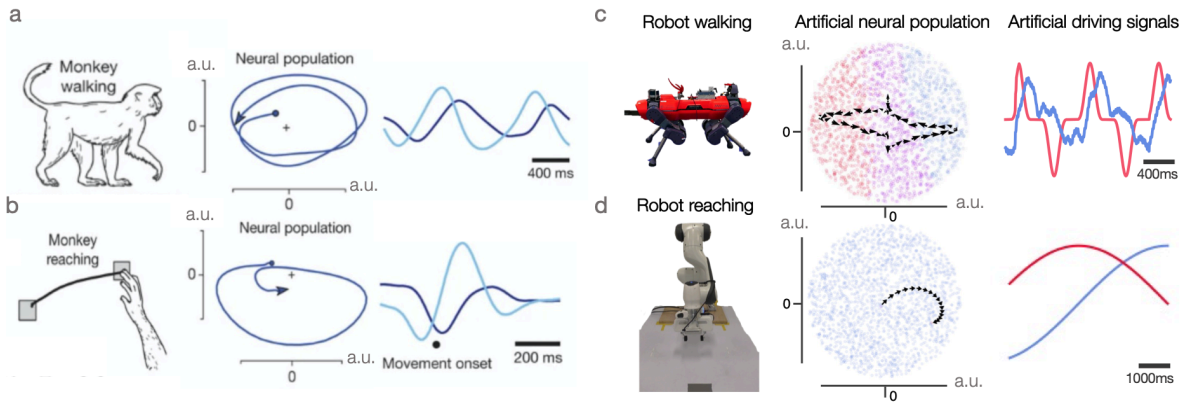


Figure 1: Rotational trajectories in monkeys and robots while walking and reaching. In the process of learning a structured latent-space for robot locomotion (*1*), we discover oscillatory dynamics that bear an intriguing resemblance to those observed in the monkey motor cortex (*2*). We show for the first time that these dynamics can be used to drive a robot arm during the non-periodic task of reaching. (a) Monkey walking. (b) Monkey reaching. (c) Robot walking. (d) Robot reaching. Note the broad analogy between corresponding monkey and robot panels. Each contains an image of the behaviour (walking or reaching), the associated rotational trajectories, and equivalent neural oscillations. Although reaching is a non-periodic behaviour, rotational trajectories are shown for both neural populations in the monkey motor cortex (axes represent principal components) and artificial neurons in the robot model (axes represent latent dimensions). Further details are explained in the text. Monkey panels adapted with permission from prior work (*2*).

(*1*). In an investigation of latent dynamics in the trained VAE during walking, one dimension of latent space stood out: it oscillated regularly and in synchrony with the robot's walking. The authors showed that by overwriting this latent dimension with a synthetic drive signal (imagine a parameterised sine wave), they could continuously control the robot's behaviour. For example, changing the amplitude of the sine wave corresponded to how high the robot raised its legs (*1,6*). The shape of the sine wave thus had interpretable effects on robot behaviour.

On the neuroscience side, Churchland et al. (*2*) report similar elliptical trajectories in neural activity recorded from the motor cortex of monkeys while performing tasks like walking and reaching (*2*). These elliptical cycles were observed in a low-dimensional projection (2D) of the population neural code. Cyclic patterns in the brain are less surprising for walking than for

reaching, as walking is a cyclic behaviour. But reaching is not obviously cyclic. Therefore, we ask whether similar oscillations drive reaching behaviour in a robot. Motivated by the example from robot locomotion (*1, 3, 6*), our aim in this study is to see if we can train a VAE for a robot arm and then control it with oscillating drive signals (see Figure 1).

Our primary contribution is to demonstrate that the state of the robot arm can be both *inferred* and *controlled* in the latent space, using synthetic drive signals which are rotational by construction, being out of phase by $\pi/2$ radians (Figure 1d). Consequently, the general approach used for legged robots (*1, 3, 6*) (Figure 1c) can be extended to reaching. When plotted in 2D, the drive signals for both reaching and walking in robots bear an intriguing resemblance to those observed in living brains (*2*) (Figure 1b,a). We are inspired by prior work such as (*7, 8*) which finds that rotational dynamics in the hidden layers of recurrent neural-networks (RNNs) (*9, 10*) can solve reaching tasks *in-simulation*. In contrast to prior works, we choose an alternative architecture to an RNN and apply rotational dynamics in a learnt representation to solve control tasks on a *real* robot platform, the Frank Emika Panda arm (*11*). As a result, our research focuses on the real-world control for both locomotion and manipulation. Indeed, simulations make simplifying assumptions. For example, physical implementations expose systems to variables like sensor noise, mechanical wear, and unmodelled environmental interactions. Moreover, demonstrating that an idea works on a physical platform is important in fields like robotics where the end-goal often includes deployment in real-world situations. In addition, we show that oscillatory movements in separate dimensions of the manipulator's latent space control movements both longitudinal (forward/backward) and lateral (left/right) movements independently. This mirrors discoveries in the robot locomotion domain where key gait characteristics such as footswing lengths and heights are disentangled within the representation (*3, 6*).

Our secondary contributions are minor: we suggest a few ways of investigating the learned representations in our models. It would be easy to exaggerate the importance of these inves-

tigations by making too much of the analogy between rotational dynamics in biological and artificial neural networks. Shared computational principles may underlie both systems. But we do not yet understand these principles, so it is too soon to draw strong conclusions (e.g. infer things about brains based on interpretations of robot systems). We therefore present the present work as preliminary.

The remainder of the paper is organised as follows. Section 2 describes the methods for training VAEs on legged robots and robot arms. It describes the methods for inferring latent states with rotational dynamics and for injecting synthetic drive signals into the latent states to control robot behaviour. Section 3 presents the results. We see that training latent space models for robot locomotion and manipulation tasks results in intriguing similarities to biological systems. Nonetheless, our primary contribution is to demonstrate that these models can be used for robot control. We further suggest a few ways of investigating the resulting systems that would be difficult to replicate in living organisms. Section 4 ends the paper with a discussion of the results in the context of robot control and systems neuroscience.

2 Materials and methods

In this section we detail how to utilise VAEs (*4, 5*) to create latent spaces in which cyclic dynamics emerge for the locomotion and reaching. Our method draws together previous work (*1, 6*) while extending it with a view to highlighting the biological inspiration of a latent-space approach to robot control. We first present common foundations for both the locomotion and reaching systems (Section 2.1) before presenting the locomotion-specific architecture (Section 2.2) and manipulator-specific architecture (Section 2.3). Next we describe the construction of drive signals for locomotion (Section 2.4), followed by drive signals for manipulation (Section 2.5). Finally, we detail the data collection for locomotion (Section 2.6) and manipulation (Section 2.7), followed by a short note on close-loop control (Section 2.8). For this study, we used

the highly-dynamic ANYmal quadruped, both B and C variants, for the walking system (see Figure 1c). For reaching, we employed a Franka Panda arm (Figure 1d). The Franka Panda arm is a 7 degree of freedom (DOF) manipulator.

2.1 A general framework for locomotion and reaching

As input to the VAE models (*4, 5*), we use a concatenation of measurable robot states sampled in time. These are quantities that are either required for inference or for variables we wish to control. For example, we encode the quadruped's end-effector contact forces in order to infer the contact state, as well as the joint configurations as these are decoded and sent to the tracking controller. *End-effectors* are contact points with the environment such as grippers for robot arms, or the points at which legs contact the ground. A *join configuration* in robotics refers to the specific angles (e.g. in degrees or radians) of all the joints in a robotic arm or leg at a given moment.

Let the input to the VAE $\mathbf{X}_k$ be a history of the previous N robot states $\mathbf{x}_k$ stacked on top of each other. A history of previous states is used as it is necessary to encode how the motion of the robot changes over time to infer the momentum of the robot at the current time step k. The control frequency f_c is the frequency the VAE-planner operates at, providing updated predictions of future robot motion. For the quadruped, this frequency is $400\,\text{Hz}$. For the manipulator, it is $20\,\text{Hz}$. To understand the different control frequencies, note that the quadruped may need to make frequent adjustments to stay upright. By contrast, the manipulator operates in a more controlled environment, so its planner does not need to sample as frequently. The states which make up the encoder's input are downsampled to a lower encoder-frequency f_enc compared to the control frequency. We thus introduce the ratio $r = f_\text{enc}/f_\text{c}$ to create the input:

$$\mathbf{X}_k = [\mathbf{x}_{k-r(N-1)}^\top, \ldots, \mathbf{x}_{k-r}^\top, \mathbf{x}_k^\top]. \tag{1}$$

An alternative formulation would be to use the derivative of the robot states instead (e.g. velocity

and acceleration). But these are very noisy quantities in practise, and implicitly inferring them helps with the domain transfer from simulation to real robot.

Once the input $\mathbf{X}_k$ is constructed, we encode both a mean latent-variable μ_k and a variance $\mathbf{\Sigma}_k^2$. The mean and variance serve to parameterise the prior distribution, in this case a zero-mean and unit-variance Gaussian. We can sample the latent variable $\mathbf{z}_k \sim N(\mu_k, \mathbf{\Sigma}_k^2)$. Together with the input action $\mathbf{a}_k$, this forms the input to the decoder. This is also the input to the performance predictor g_{pp}, which is optional and can be used to enforce semantic constraints (*1, 12–14*). In the locomotion case a performance predictor is used to predict which of the robot's feet are in contact with the ground. The robotic arm did not require any constraints in our reaching experiments. The decoder output $\hat{\mathbf{X}}_k^+$ is formed as

$$\hat{\mathbf{X}}_k^+ = [\hat{\mathbf{x}}_k^\top, \hat{\mathbf{x}}_{k+1}^\top, \ldots, \hat{\mathbf{x}}_{k+M}^\top]. \tag{2}$$

The VAE is a generative model. To train it, we maximise the evidence lower bound (ELBO) (*4, 5*). This is lower bound on the evidence of the input distribution, and is constructed of two terms. One of these is the reconstruction loss between the target values $\mathbf{X}_k^+$ and the predicted values $\hat{\mathbf{X}}_k^+$. The other term is the is the KL-divergence between the predicted distribution, $q(\mathbf{z}_k|\mathbf{X}_k)$, and the prior distribution, $p(\mathbf{z}_k)$. We use Guassian distributions for the prior, posterior, and likelihood functions. The loss function thus takes the form

$$\mathcal{L}_{\text{ELBO}} = \text{MSE}(\mathbf{X}_k^+, \hat{\mathbf{X}}_k^+) + \beta \mathcal{D}_{\text{KL}}[q(\mathbf{z}_k|\mathbf{X}_k)||p(\mathbf{z}_k)], \tag{3}$$

where the mean-squared error (MSE) is used for the reconstruction loss, and β is a hyperparameter that trades off between how well the VAE reconstructs the input data and how much it regularises the latent space to encourage disentanglement of latent factors (*15*).

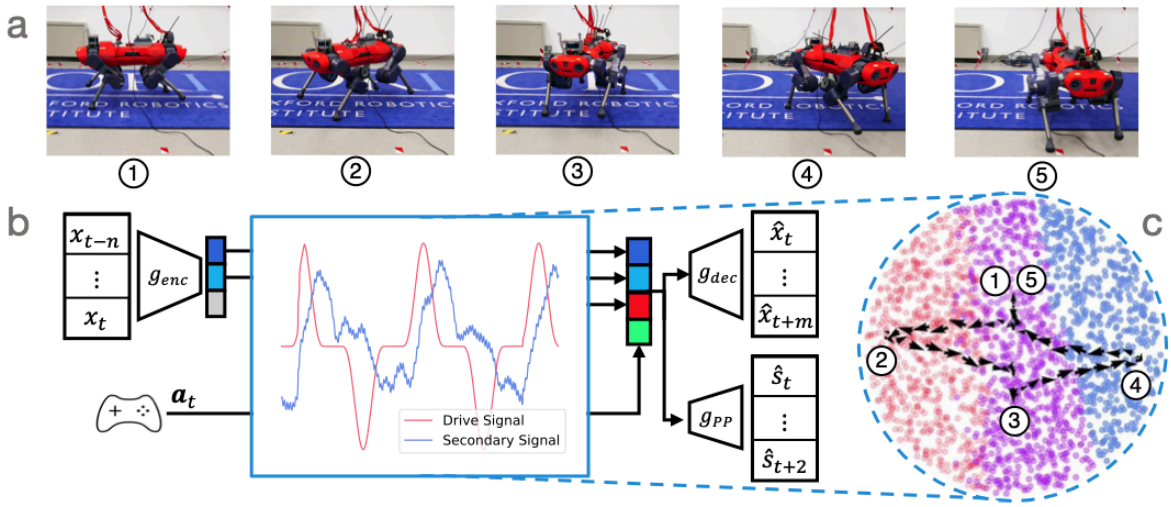


Figure 2: Robot locomotion induced by artificial drive signals. (a) Quadruped locomotion. Numbered snapshots correspond to points in the learned latent space. (b) Injection of drive signal into learned latent space. Raw sensor data is encoded into latent space prior to the injection of an oscillatory drive-signal (red). During closed-loop operation, we encode the robot states and can infer a secondary signal (blue), which is $\pi/2$ out of phase with the drive signal. Decoding the updated latent variable along with the desired base-twist action $\mathbf{a}_k$ results in locomotion trajectories which are tracked by a whole-body controller. (c) Rotational trajectory in learned latent space. Plotting the drive signal against the secondary signal reveals that locomotion follows a limit cycle in latent space (black arrows). Numbered points in latent space correspond to robot configurations seen in quadruped locomotion, as described earlier in this caption. Colour-coding of points in latent space correspond to foot configurations, to be described in Figure 7.

2.2 Locomotion-specific architecture

A few domain-specific modifications are required to learn a model for robot locomotion. Building on the general framework described above, Figure 2 provides an overview of our specific approach to the locomotion problem. We will first discuss the entire set of inputs required for locomotion planning before explaining how we structure the latent space using performance predictors.

The input robot-state $\mathbf{x}_k$ is constructed using the joint angles, end-effector positions in the base frame, joint torques, contact forces, gravity body vector, base velocity, and finally the base

pose relative to a control frame. In our case, we choose the control frame to the base pose in the world at time step $k - N + 1$. Inputs are encoded by the VAE g_{enc} and then decoded by the VAE g_{dec}. Note that the drive signal α_t is concatenated to the latent vector before decoding.

To predict which of the feet are in contact $\mathbf{s}_k$, we employ a performance predictor g_{pp} (*1, 12–14*). This performance predictor g_{pp} is a multi-layer perceptron which takes the latent variable $\mathbf{z}_k$ as input. Similar to the decoder, we predict the current time-step as well as $J - 1$ future steps:

$$\hat{\mathbf{S}}_k = [\hat{\mathbf{s}}_k^\top, \hat{\mathbf{s}}_{k+1}^\top, \ldots, \hat{\mathbf{s}}_{k+J-1}^\top]. \tag{4}$$

We train the VAE and the performance predictor g_{pp} in conjunction with each other to encourage structure that is informed by both in the latent space. We used binary cross-entropy (BCE) for the performance predictor loss. The total loss for locomotion is therefore

$$\mathcal{L} = \mathcal{L}_{\text{ELBO}} + \gamma \text{BCE}(\mathbf{S}_{\text{k}}, g_{\text{pp}}(\text{z}_{\text{k}})), \tag{5}$$

where γ is a weighting between the $\mathcal{L}_{\text{ELBO}}$ loss and the BCE.

2.3 Manipulator-specific architecture

Figure 3 provides an overview of our approach to reaching. Compared to the locomotion problem, training the VAE on reaching data is relatively simple. For instance, we do not require a performance predictor, as we do not have any contact constraints in this domain.

For reaching, the input robot-state $\mathbf{x}_k$ consists of joint angles, the end-effector position, and the goal end-effector position. Again, inputs are encoded by the VAE g_{enc} and latent vectors are decoded by the VAE g_{dec}. The synthetic drive signal α_t is concatenated to the latent vector before decoding. To achieve rotations in 2D, the drive signal is composed of two components: a sinusoid and a secondary signal that has a phase lag of $\pi/2$. This creates rotations in 2D (Figure 3c).

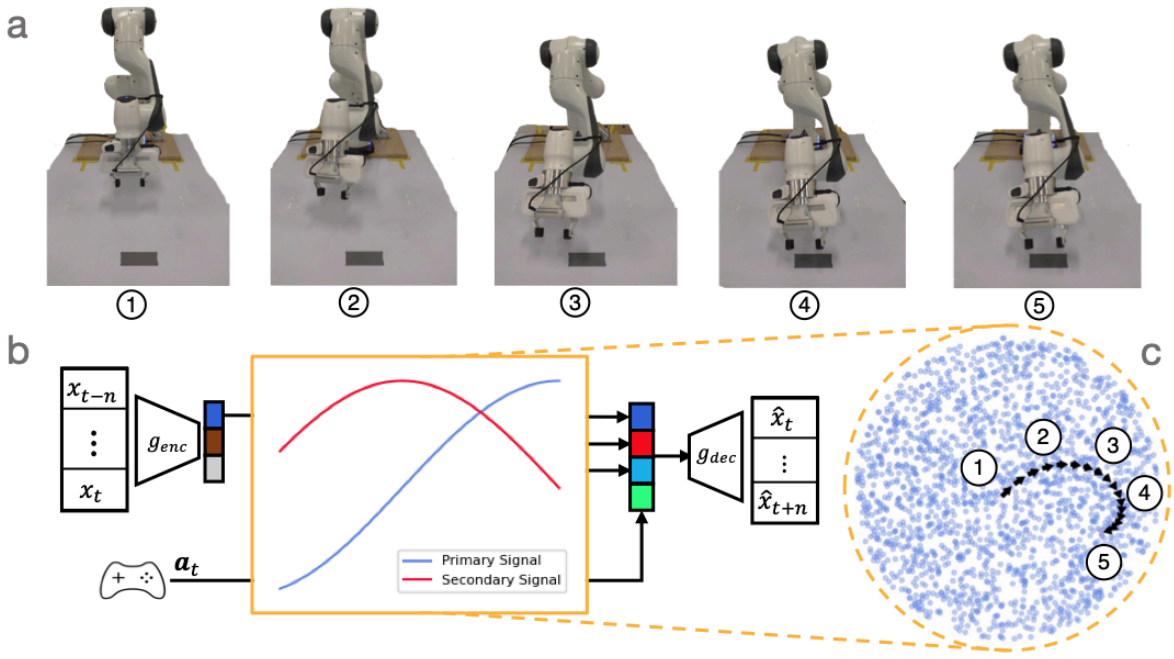


Figure 3: Robotic arm reaching induced by artificial drive signals. In contrast to locomotion, training the VAE for reaching does not require the performance predictor. (a) The outcome of the robotic arm reaching. The arm end-effector moves towards the gray tape on the table. (b) Two artificial drive signals are injected into the latent space trained with the VAE. The drive signals consist of two components where the secondary signal has a phase lag of $\pi/2$ relative to the first. (c) The latent space with the injected drive signals.

To realise robotic reaching behaviour via the injection of drive signals, which will be formulated in Equation 9, we need to form a goal robot-state $\mathbf{x}_T$ that requires goal joint angles. Goal joint angles can be obtained by inverse kinematics or manual specification by a user.

2.4 Drive signals for locomotion

To construct a drive signal for locomotion, we employ a cubed sine ($\sin^3$) drive signal with parameters for amplitude A_k and phase ϕ_k:

$$\mathbf{z}_{k,d_z} = A_k \sin^3(\phi_k). \tag{6}$$

The choice of a cubed sine is based on performance. The phase is parameterised using the time period T_k and the stance counter ϵ_k. The time period controls the quadruped's swing time, while we utilise the stance counter to artificially extend the duration that the robot has four feet in contact with the ground. Therefore, the phase dynamics update as

$$\phi_{k+1} = \begin{cases} \phi_k & \text{if } \phi_k \bmod \pi = 0 \text{ and } k_\epsilon < \epsilon_k \\ \phi_k + 2\pi/T_k & \text{otherwise} \end{cases} \tag{7}$$

and, contemporaneously, the counter ϵ_k propagates:

$$k_\epsilon \leftarrow \begin{cases} k_\epsilon + 1 & \text{if } \phi_k \bmod \pi = 0 \text{ and } k_\epsilon < \epsilon_k \\ 0 & \text{otherwise.} \end{cases} \tag{8}$$

Example drive signals and their corresponding behaviours are presented in Figure 7b. We utilise a sine cubed function since its gradient is zero when the phase is equal to a multiple of π. Therefore, when the phase is artificially held constant for ϵ_k time steps during the stance phase, the drive signal remains smooth. A smooth drive signal, in turn, yields smooth robot trajectories which are easier for the whole-body controller to track.

Given this formalisation, we can vary the drive signal's amplitude A_k, and parameterise the phase ϕ_k using T_k and ϵ_k. This results in observable correspondences between the drive signal and robot behaviour, which we describe in the results.

2.5 Drive signals for manipulation

Once the manipulation specific latent-space has been created, we inject a drive-signal into latent space which enables the robot to reach a desired goal position. To solve the reaching problem, we encode the start position $z_{\text{start}} = \begin{bmatrix} z_{k=0,d_z=0} \\ z_{k=0,d_z=1} \end{bmatrix}$ and goal position $z_{\text{goal}} = \begin{bmatrix} z_{k=T,d_z=0} \\ z_{k=T,d_z=1} \end{bmatrix}$ into latent space using VAE's encoder. With these latent variables and given the time horizon T, we introduce the following primary and secondary driving signals at timestep k for the robotic arm reaching

$$R(t) = \begin{bmatrix} z_{k,0} \\ z_{k,1} \end{bmatrix} = \begin{bmatrix} A\cos(\phi_t) + z_{T,0} \\ A\sin(\phi_t) + z_{T,1} \end{bmatrix},$$

where:

$$\begin{aligned} A &= ||(z_{\text{goal}} - z_{\text{start}})||(1 - \tfrac{k}{T}) \\ \phi_k &= \text{atan2}(z_{k=0,d_z=1} - z_{k=T,d_z=1}, z_{k=0,d_z=0} - z_{k=T,d_z=0})(1 - \tfrac{k}{T}). \end{aligned} \tag{9}$$

From this, the latent-space trajectory is decoded and the sequence of robot states are sent to a tracking controller. Control is performed in an open-loop for reaching, as the manipulator is statically stable (its base does not move). This is different than in the legged robot.

2.6 Data collection for locomotion

To train the legged robot model, we create a synthetic dataset of the quadruped walking on flat ground in simulation. The dataset contains $30\,\text{min}$ of locomotion sampled at $400\,\text{Hz}$. To vary the locomotion, we randomly sampled base-twist actions which were fed through the planning pipeline resulting in trot locomotion. The locomotion planning pipeline consists primarily of two modules, the first being *Dynamic Gaits* (DG) (*16*). This solves for the base-pose and end-effector trajectories. The second module is a whole-body controller (WBC) (*17*), which finally converts the base-pose and end-effector trajectory to joint positions, velocities, and torques.

DG solves for the base and feet plans utilising a hierarchical framework, which employs a number of kinematic and dynamic constraints. Firstly, the base-pose optimisation utilises a centroidal dynamics model (*18*), and is constrained utilising a zero moment point (ZMP) model (*19*). The WBC, which we also utilise to track the VAE trajectories consists of another hierarchical optimisation. For completeness, the constraints enforced are: base-pose tracking, contact creation, friction constraints, and torque limits.

For the simulations, we utilise the *Raisim* (*20*) simulator and an actuator network (*21*) in order to increase the fidelity of the trajectories. The actuator network uses a learned network to approximate the robot's series elastic actuators (SEA) (*22*). After training, which is performed solely using synthetic data from a single quadruped model, we deploy the same VAE models on

two different quadruped robots (ANYmal B and C) in real-world hardware experiments.

2.7 Data collection for manipulation

To train the reaching model, we collect 2000 reaching trajectories from the Robosuite simulator (*23*). The resulting trajectories are the output of an expert policy trained with a soft actor-critic (*24*). The input to the VAE consists of joint angles, end-effector position, and the goal position in Cartesian coordinates. After training the VAE on the simulated datasets, we deploy the VAE on a real physical Franka Panda arm robot.

2.8 Closed-loop control

For completeness, let us briefly note the use of open and closed-loop controllers. *Open-loop controllers* operate without feedback. Their actions are pre-programmed and not adjusted based on sensor data. *Closed-loop controllers*, on the other hand, require the system to perform constant monitoring and feedback, to adjust based on interactions with the environment.

For closed-loop control, the trained VAEs are used as planners and deployed on the quadrupeds and on the manipulator. This has a number of benefits, such as mitigating external disturbances and counteracting unmodelled dynamics. For example, the ANYmal actuators are SEAs (*22*), which are complex systems themselves and are difficult to model analytically, exhibiting dynamic differences between units.

To plan in latent space, we first estimate the current latent variable. Following Equation 1, we create the encoder input $\mathbf{X}_k$. We then pass $\mathbf{X}_k$ through the VAE's encoder to predict $\mathbf{z}_k$. The predicted value $\mathbf{z}_k$ is concatenated with $\mathbf{a}_k$, which is the desired base-pose twist. Next, we inject our drive signals into latent space, as described above, for the quadruped (Section 2.4) and manipulator (Section 2.5). This produces the predicted trajectory snippet, which we decode and send to our tracking controller. For locomotion, the controller is the WBC described

in Section 2.6. The manipulation trajectories are instead tracked using a joint-space position controller. This entire process is repeated at a frequency of $400\,\mathrm{Hz}$ on the quadruped robot, and at $20\,\mathrm{Hz}$ for the manipulator.

3 Results

Our main result is a proof of concept: we find that the oscillating drive signals, which produce circular motion in latent space, can lead to smooth and directed reaching behaviour in a physical robot. This was already glimpsed in the model architecture (Figure 3) but is highlighted again here in Figure 4. This Figure 4a shows the shape of the oscillatory drive signals, a primary sine wave and a secondary sine wave with its phase shifted by $\pi/2$. These oscillators correspond to a partial cycle in the model's learned latent space (Figure 4b), which in turn corresponds to a smooth trajectory in physical space from the arm's initial position to its goal position. Grey tape appears on the table over the goal position.

As a secondary aim, we would like to understand the model's learned latent representations. To investigate how consistent they are, we trained the reaching model with 4 different random seeds, observing the robot arm trajectories and plotting the latent spaces. These results are presented in Figure 5. The results reveal that the learned latent spaces differ between random seeds: the latent trajectories do not begin or end in the same places between random seeds. However, each learned latent space reliably enables the arm to reach the same target when controlled with oscillatory drive signals. In effect, the robot arm trajectories were the same across random seeds but we can see variations in the paths traced in latent space.

To explore what the dimensions in the learned latent space correspond to, we can sample trajectories in latent space along a line and then visualise the corresponding robot arm configurations. Figure 6a shows five points sampled along the horizontal axis, from left-to-right. Figure 6b shows five points sampled along the vertical axis, from bottom-to-top. The corresponding

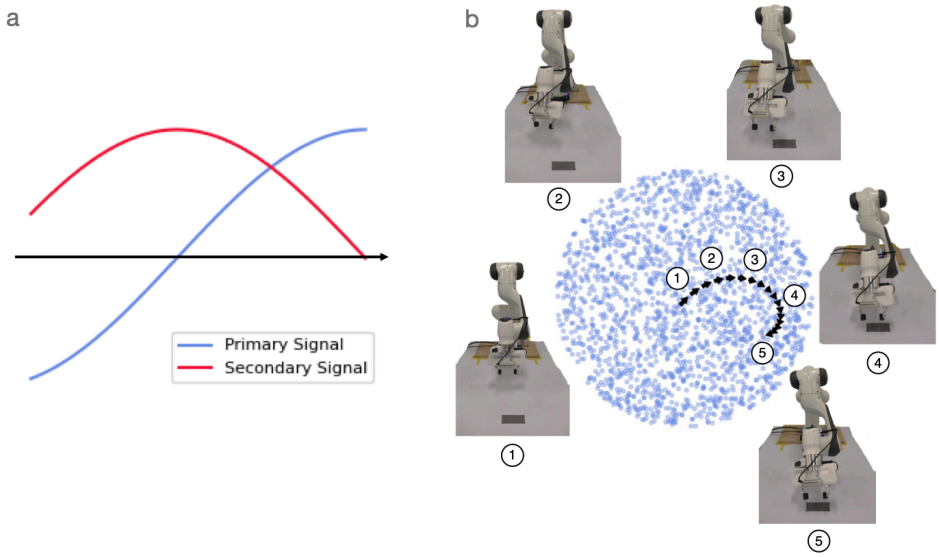


Figure 4: Oscillatory drive signals in the learnt latent-space decode to non-periodic reaching trajectories. (a) The two oscillatory drive signals. Both are sinusoidal, with the secondary drive signal being $\pi/2$ out of phase with the primary drive signal. (b) Plotting the drive signals against each other reveals a rotational trajectory. The circled numbers along the latent trajectory (black arrows) correspond to the robot arm configurations, which together depict a smooth trajectory from an initial end-effector position to one where it has reached the target (grey square on the table).

robot arm configurations are shown in Figure 6c and d, respectively.

For the horizontal traversal of the latent space, we see the arm initially move 'backward' (from the robot's point of view) before moving 'forward' (Figure 6c). For the vertical traversal, the arm movement is more linear (Figure 6d). One general observation, from the horizontal traversal, is that the latent dimensions do not necessarily correspond to linear movements in observation space. It would be interesting if this level of analysis generalised from robots to brains, as we argue in the discussion (Section 4). However, before continuing to the discussion, let us briefly consider an analysis of the robot locomotion system.

Prior work shows that similar oscillations can drive robot locomotion (*1, 3, 6*). As presented in the introduction, Mitchell et al. (*3*) originally trained a VAE on the states of a walking robot

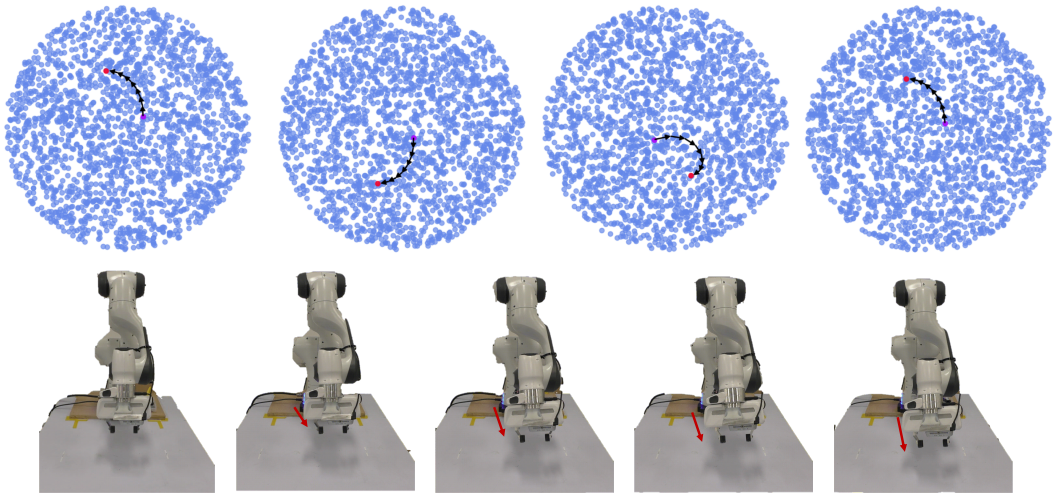


Figure 5: Consistency of latent space structure in robotic arm reaching. A VAE is trained with 4 different seeds. Then its latent space is evaluated by adding drive signals given the same start and goal position for reaching behaviour. All of the latent spaces with the drive signals induces the same successful reaching behavior. This result indicates that the latent space consistently captures the smooth geometry information in the robot workspace, but the latent space represents this workspace differently depending on the seeds.

and noticed periodic signals in the inferred latent space. We visualise that here in Figure 7a. Replacing the primary periodic signal which emerged naturally with a synthetic drive signal that was hand crafted (cf. Figures 7a,b), the authors found they could control the behaviour of the legged robot. In this case, the secondary signal was inferred by the system (cf. Figure 3). Plotting the drive signal and inferred secondary signal in 2D, the resulting trajectory in latent space could be related to different robot states (Figure 7c). Perhaps the deepest understanding of this relationship can be seen from the colour-coding of the latent space, to indicate which regions of latent space correspond to which robot configurations. For example, in 7c, the latent states that are coloured purple correspond to robot configurations in which all four of its feet

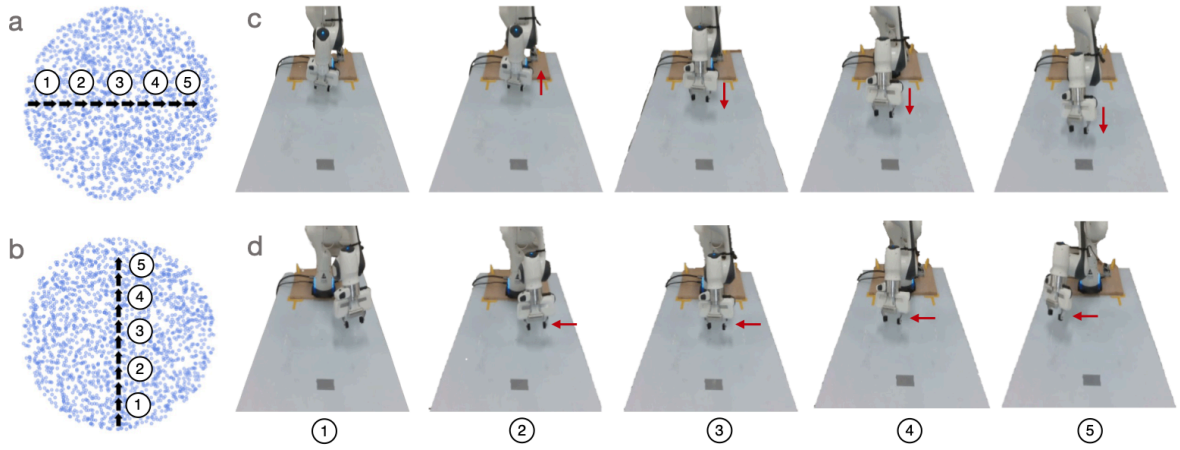


Figure 6: Moving in straight lines across each dimension of the latent space. To explore the learned latent space, we traverse both of its dimensions and decode the corresponding robot arm configurations. (a) Traversing the 'horizontal' latent dimension. (b) Traversing the 'vertical' latent dimension. (c) Over five snapshots of behaviour for the 'horizontal' traversal in latent space, we see the arm in its initial configuration move 'backward' before moving 'forward' (from the robot's point of view; see red arrows). (d) In the 'vertical' traversal of latent space, the robot arm moves more consistently in the same direction (left-to-right from the robot's point of view; red arrows).

are on the ground. As the trajectory in latent space cycles in and out of these purple regions, the robot transitions from having all four feet on the ground, to configurations in which different feet are not touching the ground.

Finally, it is possible to control the robot's behaviour by modulating the drive signal. This can be seen in Figure 8. For example, 8d shows what happens when the amplitude of the drive signal is reduced over time. Note that the amplitude of the drive signal controls the step height of the robot. The height of the robot's steps thus gets smaller as the amplitude of the drive signal gets smaller. Ultimately the steps are too small to overcome friction and the robot halts.

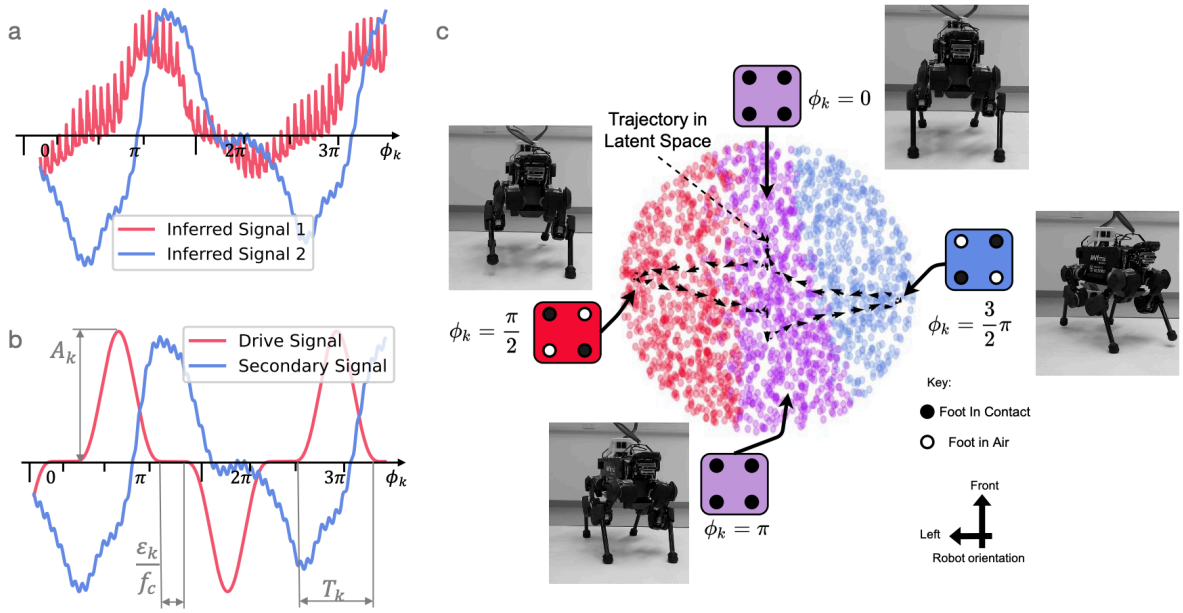


Figure 7: Latent space oscillations, rotation trajectories, and corresponding robot configurations. (a) Inferred oscillations. These emerge in an unsupervised generative model trained to observe a set of state space quantities (e.g. joint angles) of an ANYmal B quadruped as it moves. (b) Synthetic drive signal. To test our understanding of the primary inferred signal, we replace it with a single sinusoid. This can be shaped by varying a set of parameters at each time step k: amplitude A_k, phase ϕ_k, time-period T_k, and full-support duration ϵ_k / f_c. We use a cubed sine wave for locomotion to better model the full-support duration which is the period in which all four feet contact the ground. (c) Rotational trajectories and corresponding robot configurations. Plotting the synthetic drive signal against the inferred secondary signal results in rotational trajectories in latent space (black arrows). To aid interpretation, each point in latent space can be decoded to a robot configuration, which we colour code according to one of three cases: all four feet are on the ground (purple); the Left Front and Right Hind feet are on the ground (red); or Right Front and Left Hind feet are on the ground (blue). Points were randomly generated in the latent space and then propagated through the decoder to assign a colour (corresponding to a foot configuration). See text for further details. Figure adapted from prior work (*6*).

4 Discussion

In this paper, we demonstrate that a real-robot manipulator can be controlled using oscillatory signals in a structured latent-representation. We also observe that the properties of these signals (namely the $\pi/2$ phase lag) appear to resemble those found in the brain's of primates during

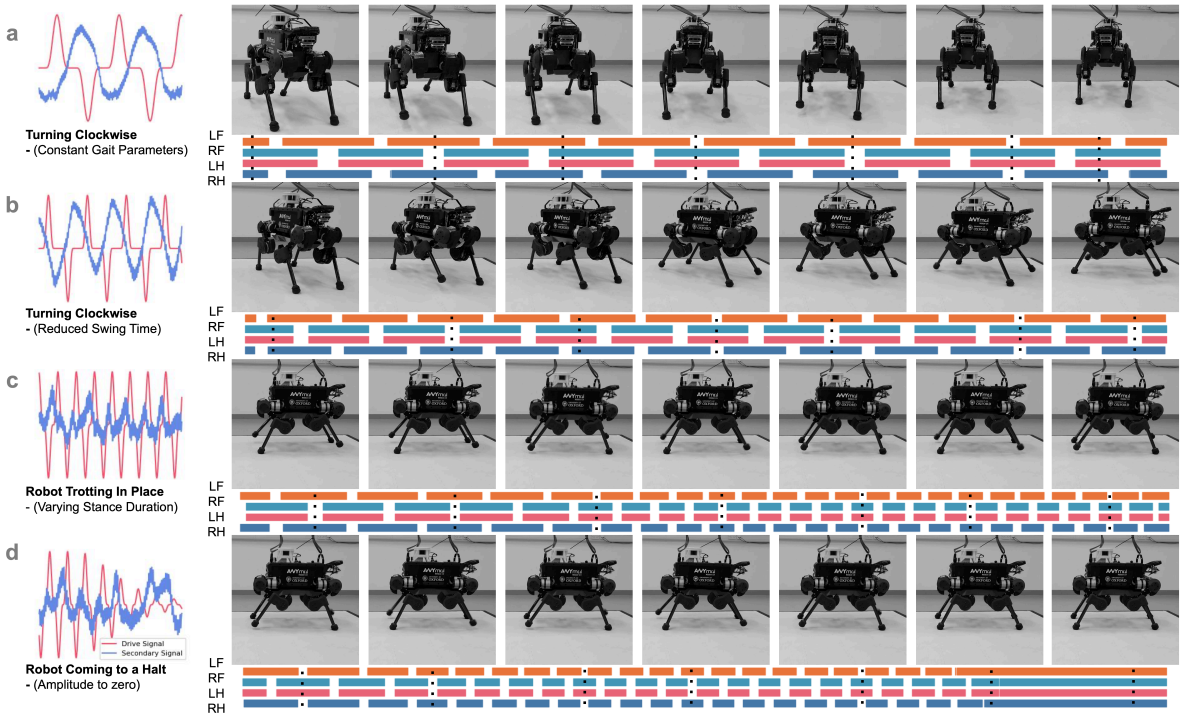


Figure 8: Modulating the amplitude and frequency of the drive signal leads to continuous and interpretable variation in robot gaits. Feet are abbreviated in the contact schedule: LF (Left Front), RF (Right Front), LH (Left Hind), and RH (Right Hind). (a) Robot turning clockwise. (b) Faster clockwise turning with a higher frequency drive signal. (c) Change in drive signal and swing duration over time. (d) Reducing the amplitude of the primary drive signal over time. This ultimately results in the robot halting. Figure adapted from (*1*).

reaching (see Figure 1). In learning a latent representation for the control of real robot systems, we find that we can solve robotics tasks using *explicit* rotations (i.e. $\pi/2$ oscillatory drive signals). Prior art has argued for the use of rotational signals in recurrent neural networks (*8*), where the internal state (hidden activations) evolves over time and rotations emerge from (or are *implicit* in) the choice of network architecture. We further find structure in the learned latent spaces of our models. For example, inspection of the manipulation latent representation reveals that when oscillations are injected into each dimension of a 2D slice of latent space the robot end effector moves in interpretable patterns, either up and down or side to side (Figure 6). The consistent emergence of this learned latent space for reaching is verified by training multiple VAEs with four different random seeds and observing the resulting latent structures (Figure 5). This result is similar to the locomotion case where individual latent dimensions encode swing lengths and heights independently as found in prior work (*1, 6*). Together, these results illustrate the kinds of interpretable structures that emerge in such models' learned latent spaces.

To compare the legged robot, we followed prior work (*1, 3, 6*) in colour-coding the latent space based on the robot's foot configurations (Figure 7c). We also varied the shape of the drive signal and observed how this affects the robot's behaviour (Figure 8). Additional exploratory analyses are included in the Supplementary Materials, where we observe the latent space when the legged robot is pushed while walking, deviating from its expected path (Supplementary Figure S1). We also explore how behaviour in the legged robot responds to virtual lesions (Supplementary Figure S2).

In this paper, we extend the VAE-based approach of Mitchell et al. (*3*) from walking to reaching. While this approach shares foundational assumptions with theoretical neuroscience, it diverges as well in many details. The shared assumptions are very general: for example, current theories of neuroscience view the brain as a generative model that optimises an objective (free energy) which is equivalent to optimising the ELBO for VAEs (*25, 26*). In the details, however,

stronger links have been forged by work in computational neuroscience (*7, 8*). Our contribution to this line of work is to extend it from computational simulations to real-world robot platforms: a proof of concept that rotating dynamics in latent space work to drive walking and reaching in robots. The representational learning approach allows one to investigate the latent spaces learned by instances of reaching and walking robots. An important limitations is that we do not claim the learned latent spaces for robots are the same ones learned by animal brains. Rather, we suggest that they represent possible solutions for similar behaviours. Our results build on the recent exploration of VAEs for robot control, which includes the work of Mitchell et al. (*1, 3, 6*) as well as prior work on reaching with VAEs (*13*). Prior work on reaching (*13*) did not use oscillatory drive signals. The present work therefore helps to unify both behaviours, walking and reaching, under a common biologically-inspired framework.

Data availability

The datasets used and/or analysed during the current study available from the corresponding authors on reasonable request.

References

1. A. L. Mitchell, *et al.*, Next steps: Learning a disentangled gait representation for versatile quadruped locomotion, *IEEE International Conference on Robotics and Automation (ICRA)* (2022).

2. M. M. Churchland, *et al.*, Neural population dynamics during reaching, *Nature* **487** (2012).

3. A. L. Mitchell, *et al.*, First steps: Latent-space control with semantic constraints for quadruped locomotion, *IEEE/RSJ Int. Conf. Intell. Rob. Sys. (IROS)* (2020), pp. 5343–5350.

4. D. Kingma, M. Welling, Auto-encoding variational bayes, *Int. Conf. on Learn. Repr. (ICLR)* (2014).

5. D. J. Rezende, S. Mohamed, D. Wierstra, Stochastic backpropagation and approximate inference in deep generative models, *Int. Conf. on Mach. Learn. (ICML)* (2014).

6. A. L. Mitchell, *et al.*, Vae-loco: Versatile quadruped locomotion by learning a disentangled gait representation, *IEEE Transactions on Robotics* **39**, 3805 (2023).

7. D. Sussillo, M. M. Churchland, M. T. Kaufman, K. V. Shenoy, A neural network that finds a naturalistic solution for the production of muscle activity, *Nature Neuroscience* **18**, 1025 (2015).

8. H. T. Kalidindi, *et al.*, Rotational dynamics in motor cortex are consistent with a feedback controller, *eLife* **10**, e67256 (2021).

9. S. Hochreiter, J. Schmidhuber, Long short-term memory, *Neural computation* **9**, 1735 (1997).

10. R. M. Schmidt, Recurrent neural networks (rnns): A gentle introduction and overview (2019).

11. S. Haddadin, *et al.*, The franka emika robot: A reference platform for robotics research and education, *IEEE Robotics Automation Magazine* **29**, 2 (2022).

12. Y. Wu, *et al.*, Imagine that! leveraging emergent affordances for 3d tool synthesis, *arXiv preprint arXiv:1909.13561* (2020).

13. C.-M. Hung, *et al.*, Reaching through latent space: From joint statistics to path planning in manipulation, *IEEE Robotics and Automation Letters (RA-L)* (2022).

14. Y. Wu, *et al.*, Learning affordances in object-centric generative models, *Int. Conf. on Mach. Learn. (ICML)* (2020).

15. I. Higgins, *et al.*, β-VAE: Learning basic visual concepts with a constrained variational framework., *Int. Conf. on Learn. Repr. (ICLR)* (2017).

16. C. D. Bellicoso, F. Jenelten, C. Gehring, M. Hutter, Dynamic locomotion through online nonlinear motion optimization for quadrupedal robots, *IEEE Robot. Automat. Lett. (RA-L)* **3**, 2261 (2018).

17. C. Dario Bellicoso, *et al.*, Dynamic locomotion and whole-body control for quadrupedal robots, *IEEE/RSJ Int. Conf. Intell. Rob. Sys. (IROS)* (2017), pp. 3359–3365.

18. D. Orin, A. Goswami, S.-H. Lee, Centroidal dynamics of a humanoid robot, *Autonomous Robots* **35** (2013).

19. M. Vukobratovic, B. Borovac, Zero-moment point - thirty five years of its life., *Int. J. Hum. Rob. (IJHR)* **1**, 157 (2004).

20. J. Hwangbo, J. Lee, M. Hutter, Per-contact iteration method for solving contact dynamics, *IEEE Robot. Automat. Lett. (RA-L)* **3**, 895 (2018).

21. S. Gangapurwala, M. Geisert, R. Orsolino, M. Fallon, I. Havoutis, RLOC: Terrain-aware legged locomotion using reinforcement learning and optimal control, *arXiv preprint arXiv:2012.03094* (2020).

22. G. A. Pratt, M. M. Williamson, Series elastic actuators, *IEEE/RSJ Int. Conf. Intell. Rob. Sys. (IROS)* (1995).

23. L. Fan, *et al.*, Surreal: Open-source reinforcement learning framework and robot manipulation benchmark, *Conf. on Rob. Learn. (CoRL)* (2018), pp. 767–782.

24. T. Haarnoja, A. Zhou, P. Abbeel, S. Levine, Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor, *Int. Conf. on Mach. Learn. (ICML)* (2018).

25. K. Friston, A theory of cortical responses, *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences* **360**, 815 (2005).

26. P. Mazzaglia, T. Verbelen, O. Çatal, B. Dhoedt, The free energy principle for perception and action: A deep learning perspective, *Entropy* **24**, 301 (2022).

Acknowledgments

This work was supported by a UKRI/EPSRC Programme Grant [EP/V000748/1], the EPSRC grant 'Robust Legged Locomotion' [EP/S002383/1], the EPSRC CDT[EP/L015897/1], the UKRI/EPSRC RAIN [EP/R026084/1] and ORCA [EP/R026173/1] Hubs and the EU H2020 Project MEMMO (780684). It was conducted as part of ANYmal Research, a community to advance legged robotics. The authors would like to acknowledge the use of the SCAN facility, and thank Martin Engelcke, Siddhant Gangapurwala, and Jack Collins for their valuable feedback. Please contact Nature for permission to reproduce sub-Figures 1a,b which come from prior work (*2*).

Supplementary Materials

Supplementary Text

Figures S1 and S2

S1 Pushing experiments / The robustness of the VAE-planner

As a supplementary analysis, we show that the VAE-planner is able to detect and reject external disturbances. Figure S1 adapts two figures from our previous work (*6*) that illustrate what

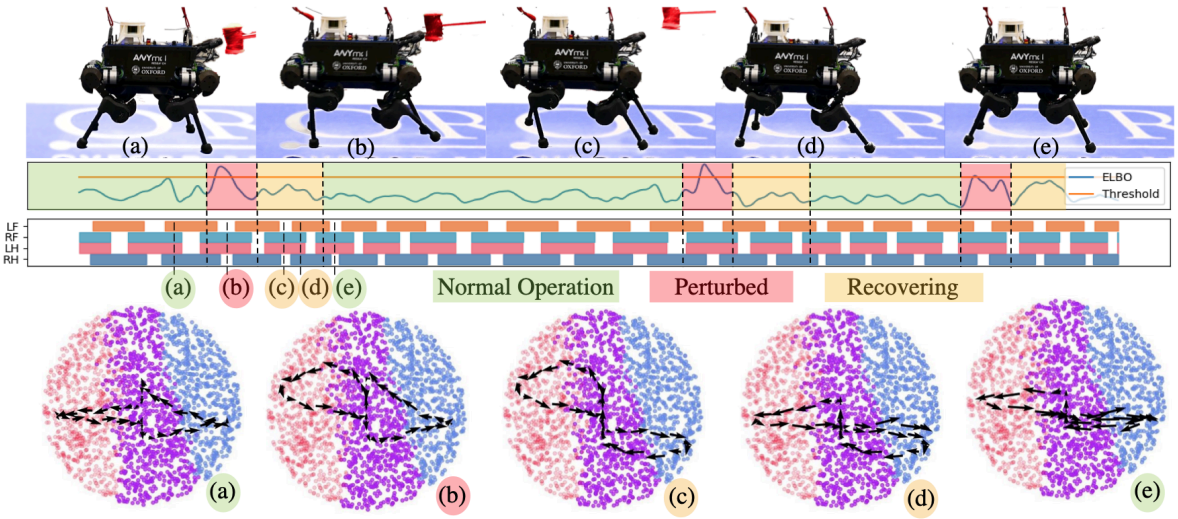


Figure S1: The VAE approach is capable of both detecting and reacting to external disturbances. Here, the external disturbance is a large push from the red broom (top panels). The ELBO corresponding to the encoded input is monitored and if this surpasses a pre-determined threshold, the VAE-planner increases the robot's cadence. Initially, the robot is walking normally and the ELBO is below the threshold (a). At point (b), the robot is pushed. The ELBO rises above the threshold triggering the swing duration to halve. This increase in cadence is captured at point (c) in the contact schedule. After roughly three or four steps, the robot recovers and the swing duration returns to nominal (d) and (e). The bottom row shows the corresponding latent-space trajectories starting with the nominal mode at (a). The external disturbance is clearly visible at point (b) as a large red object (a broom). Adapted from prior work (*6*).

happens to the quadruped's latent space when normal operation is perturbed by an external disturbance (pushing the robot with a broom). Figure S1 shows the robot first detecting the disturbance and then reacting by increasing its cadence to counteract the push. The corresponding latent space plots show how the rotational trajectories are pushed out of orbit but then return towards the original limit cycle. This internal recovery of latent dynamics contrasts with the virtual lesion example, in which the dynamics never recover (Figure S2).

S2 Virtual lesions

Another way to explore the relationship between latent dynamics and behaviour in our VAE approach is to employ virtual lesions. How does progressively damaging the underlying network affect both dynamics and behaviour? To illustrate this, we focus again here on the ANYmal B quadruped robot. By 'lesion' or 'damage', we specifically mean zeroing artificial neurons in the VAE's decoder (illustrated schematically in Figure S2a). In brief, we mask the hidden units in the decoder, randomly setting a subset of them to zero. In the implementation we use cascade failure dropout masks: masks are initialised with ones; mask units are then changed to zero with random probability; finally, zeroed mask units remain zeroed for the remainder of the experiment. The last point means that the virtual lesions are 'progressive', expanding over time without 'recovery'.

Three phases stand out in the exploratory lesion analyses. The first phase is 'behaviour as usual': we inject a drive signal into the latent space and observe the robot walking. In the second phase, we apply progressive lesions to the VAE, using the cascade failure dropout mask approach. Finally, in the third phase we pause lesioning the model and simply observe the latent dynamics and robot behaviour. What happens to the dynamics of the model and behaviour of the robot when the damage stops?

In the latent space, these three phases correspond to distinct dynamics (Figure S2b). In the

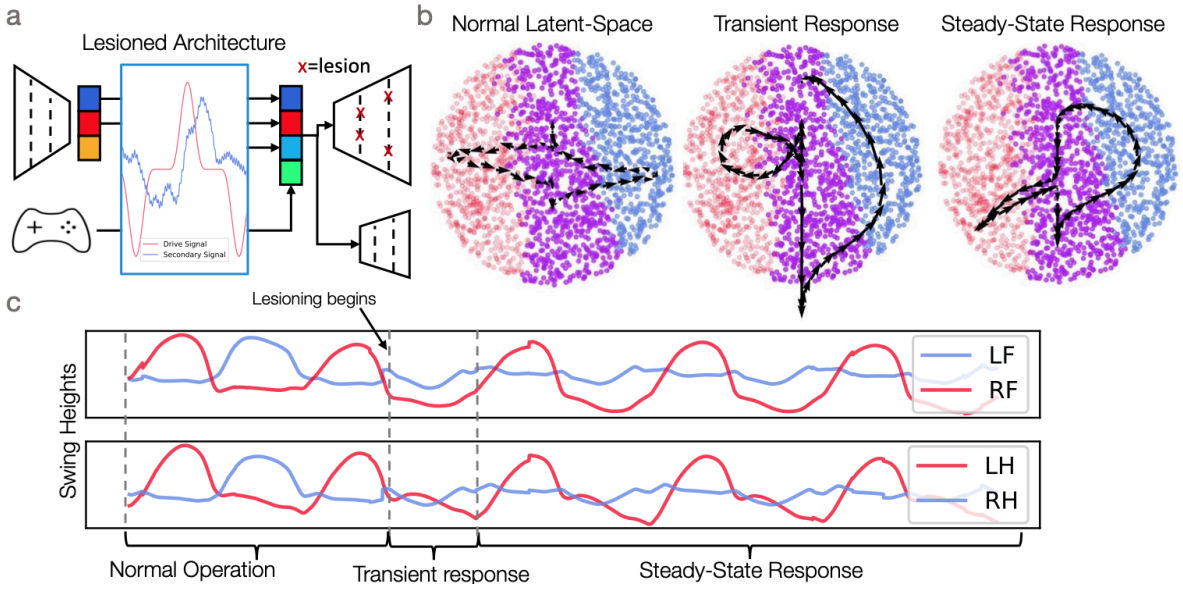


Figure S2: Virtual lesion analysis. (a) We mask hidden units in the VAE's decoder while feeding the drive signal in as usual. Decoder outputs contribute to encoder inputs and thus to the latent space dynamics. Lesions in the decoder are marked with red Xs. (b) Plotting the rotational trajectories in the latent space. We observe three phases in the latent-space dynamics. The first corresponds to normal operation. The second is the transient response while the network is being lesioned. The third is the steady-state response after lesioning has ended. Colour coding is the same as in Figure 7. (c) The robot behaviour is displayed. We plot the swing height of the robot legs over time across the same three phases observed in the latent space. Lesioning is progressive and continues through the transient response period. Feet are abbreviated: LF (Left Front), RF (Right Front), LH (Left Hind), RH (Right Hind).

first phase, no lesioning has yet been applied and we observe normal dynamics. In the second phase, when the network is virtually lesioned, we see a transient response in the dynamics, where the latent trajectories become aberrant. We can represent the robot's behaviour by plotting the swing height of each foot (Figure S2c). Finally, the second phase (lesioning) dynamics persist into the third phase (steady-state response). In this phase we see that the Left Front and Right Hind legs lose their swing height. In other words, the robot begins to drag its legs on the ground as it tries to walk.

Although hidden units are zeroed in regular increments through the second phase of the

experiment, behaviour carries on for a while, until it fails rather suddenly. One hypothesis is that significant function is concentrated in specific hidden units: thus, zeroing these units causes shift in behaviour. Another hypothesis is that there is a capacity threshold: if the right number of neurons are damaged, then behaviour changes suddenly. But if we re-ran the experiment different 'critical neurons' would emerge.

To explore these hypotheses, we repeat the lesion experiment 200 times with different random seeds, the specific question being whether the critical change in behaviour corresponds across random seeds to the same number of hidden units being zeroed or not. We find that the number of hidden units that need to be zeroed before we observe the change in behaviour varies across the 200 times we repeated the experiment. In addition, once a critical neuron is zeroed, we find that normal function resumes if we restore that one unit by changing the mask from zero to one (i.e. 'rewinding' the damage). Together, these results suggest that damage to specific neurons accounts for the critical changes in behaviour.

If this is correct, what do these critical neurons do? To present one suggestive example, we observe that, for each repeat of the experiment, opposite front and back legs fail in behaviour together. The Left Front and Right Hind legs failed together more often (54.9%) than the Right Front and Left Hind legs (29.4%). In the remaining 15.7% cases, all four legs appear to fail at the same time. This suggest that the the lesioned neurons have learned shared representations for leg pairs. This makes sense given that the models are trained on trot gaits in which the movement of these legs are always correlated. If the models were trained with a wider variety of gaits, these correlations could be broken. In this case, we conjecture that lesioned neurons would show different patterns.

Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor **(only required where there isn't already a statement of contribution within the paper itself).**

Title of Paper	Oscillating Latent Dynamics In Robot Systems During Walking And Reaching
Publication Status	Published
Publication Details	Oiwi Parker Jones*, Alexander L. Mitchell*, Jun Yamada*, Wolfgang Merkt, Mathieu Giesert, Ioannis Havoutis, Ingmar Posner Published in the journal of Nature Scientific Reports

Student Confirmation

Student Name:	Alexander Luis Mitchell		
Contribution to the Paper	Led following efforts: • Proposing the project and organising the team • All work on quadruped including training models and experiments • All quadruped figures and first figure • Writing all quadruped related sections Co-led efforts: • Writing the introduction Efforts by other co-authors: • Manipulation work		
Signature		Date	02/OCT/2023

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title: Professor Ioannis Havoutis		
Supervisor comments These statements are correct.		
Signature Ioannis Havoutis	Date	02/OCT/2023

7
Discussion and Limitations

This thesis investigates the creation of a unified representation across multiple gaits in order to find smooth transitions between them. Chapter 5 addresses this directly and we show a VAE-based planner, which produces terrain-aware locomotion over uneven terrain with continuous gait switching. In the preceding chapters, we answer a series of intermediate questions to gain insight which helps overcome the limitations of each development.

In Chapter 3, we develop methods for structuring a latent space for locomotion as well as introduce a gradient-based planning method for generating static trajectories. In particular, we investigate how to tackle the non-linearities introduced by the contact dynamics and cluster samples by contact state in the latent space. The development that constraints such as the contact-switching dynamics can be expressed as traversing crisp boundaries in latent space is fundamental to all research in this thesis going forward. Indeed, these boundaries are analysed in Chapter 4.

The limitations of Chapter 3 are addressed in the next two works found in Chapter A and Chapter 4. The first limitation is the gait generated is statically stable only and the second is that the gradient-based planner is too slow for deployment on the robot in an MPC fashion. The former limitation is addressed by training with dynamic trot gaits and creating an encoding which estimates the robot's pose and momentum. In doing this, we discover that the dynamic locomotion is encoded as periodic signals in the

robot's latent space. Moreover, we find that the latent space is disentangled as well as structured. We find that oscillations in the vertical axis decode control the robot's footstep length and those in the horizontal govern the swing length. A novel oscillatory planner in the space leverages these properties to create trot locomotion where the in-swing characteristics of the robot's gait can be continuously modified during the flight phase. The latent space structure in the space is a product of estimating the robot's contact state as found in Chapter 3. Furthermore, we investigate the creation of the crisp decision boundaries between contact states and confirm that these are created by back-propagating the gradients from the performance predictors back through the VAE's encoder. Without the crisp boundaries, contact switching would not be deterministic and the subsequent trajectories would not be suitable for locomotion.

Inspired by the work in Chapter 4, which finds a representation for a set of trot gaits, we develop methodologies to create a unified manifold for multiple gaits and terrain traversal. In Chapter 5, we train a VAE using examples of distinct gaits and automatically discover smooth interpolations between these skills. Additionally, we condition the latent space using a rudimentary terrain encoder and using a learnt planner in the space produce terrain-aware and reactive locomotion. We train the planner in latent space using behavioural cloning and see that the resulting behaviour exploits the structure of the latent space to modulate the swing length and heights in response to the terrain. In particular, we see that footstep length significantly increases resulting in a stable foothold. This unified latent space is expressive enough to produce multiple skills and structured to exploit correlations between different gaits leading to seamless transitions between the learnt skills.

Finally, we discover that there exist similar oscillatory signals found in *VAE-Loco* in the brains of primates during locomotion and reaching. We compare the periodic signals found in the latent space to those in the motor cortex and note that the signals have similar time periods and phase lags. Furthermore, we utilise the same locomotion oscillations to solve reaching problems in a manipulation-specific latent space. This mirrors Churland et al.'s findings in [25] which report similar oscillations present in the motor cortices of primates during reaching. Finally, we stretch the analogy between the VAE and the

motor cortex and lesion the VAE's decoder. This means that the VAE is no longer able to produce the locomotion trajectories accurately. In particular, we observe a lack of control over half the robot's limbs and note that this occurs when a set of specific neurons are cut. This work in Chapter 6 hints at the possibility that there is an intrinsic representation for locomotion in nature similar to those found in the latent space for locomotion.

7.1 Limitations

There are some limitations common to all the latent space traversal planning methods presented in this thesis. The first is kinematic consistency. We have shown in multiple chapters that the joint-space trajectories generated by the VAE are dynamically feasible. However, one persisting issue is the requirement for the joint-space trajectories to match the task-space ones. For the ANYmal robot, the task-space quantities (e.g. base and feet positions) are analytically related to the joint-space trajectories (e.g. joint angles, torques) via forward kinematics. As mentioned, the VAE tracks the robot's complete pose in the world by integrating base-pose deltas and using forward kinematics to calculate the positions of the robot's feet. The base-pose deltas will be biased due to precision errors and over time these errors will be integrated to drift, meaning that feet in contact will slide slightly. We see this phenomenon in *VAE-Loco*, where the base leans over the front feet when the robot is walking forward. To alleviate this, we close the base-loop so that we replan from the measured base-pose periodically. Nevertheless, there remains drift over time.

This issue worsens when climbing uneven terrain since the joint-space trajectories adapt in response to the terrain. Therefore, the VAE must produce world-space trajectories where the base rises above the terrain and the feet are placed in stable locations on the terrain. During training, there is no explicit loss for inaccuracies in this prediction apart from the reconstruction loss. As such there is no signal during training from the environment to correct inaccuracies between the joint and task spaces.

In Chapter 5, we introduce a learnt planner in latent space. This is a rudimentary planner trained with behavioural cloning. There is scope for a more sophisticated planner which has access to a depth image of the terrain around the robot and has access to

the base velocity and gait commands. In doing so, a more sophisticated planner could plan for long-horizon tasks as well as switch gaits in response to terrain to aid in the traversal of very steep or tall obstacles.

Another potential limitation of the work in its current form is the requirement of expert trajectories. However, a key contribution and strength of this approach is its ability to represent a diverse range of skills/gaits in a unified representation. As seen in the previous chapter, this facilitates continuous transition between multiple gaits and fusion with sensor data for terrain traversal. Therefore, expert trajectories from multiple different sources can be represented in the latent space. Thus as long it is possible to create examples of motion using alternative methods, the VAE can embed these into a structured latent space and if feasible find continuous paths between them.

All the locomotion trajectories produced by the VAEs are tracked using the whole-body controller (WBC). This is a capable, optimisation-based tracking controller, but it is likely that the trajectories are nearing the limits of its performance. In literature, we see that the WBC is capable of tracking the majority of useful motions on uneven terrains both in the lab and in the wild. However, this does not include very steep staircases or significant gradients. Instead, we see that the state-of-the-art dynamic motions such as stair climbing or jumps are solved using either differential dynamic programming (DDP) [2, 4] or RL [13, 46]. The former set of methods utilises the complete dynamics equation as constraints and optimises the joint-actuator gains as well as the joint position, velocities and torques. This results in motions optimised for the complete dynamic range of the robot and the response of the actuators can be continuously optimised during the dynamic manoeuvres. Similarly, RL can exploit the full dynamics range of the robot as approximated by the simulation software. As mentioned, the WBC optimises joint torques using simplified centroidal dynamics and inertia-less legs, which limits the controller's ceiling of performance.

7.2 Future Work

There are two streams for the future work of latent space planning. The first stream focuses on improving the performance of the locomotion resulting in even more dynamic motions.

For example, emerging literature is shifting focus to long-horizon dynamic planning such as robot parkour. The second stream includes the application of this work to complex and highly under-actuated platforms. Candidate platforms are loco-manipulators as well as tendon-driven or even soft-robot manipulators or hands. The methods presented in this thesis are highly effective at creating low-dimensional representations for quadruped robots. For example, *Gaitor* is capable of producing multiple gaits for a 18 -dimensional robot from a two-dimensional representation. By extension, planning for highly coordinated movements in a latent space for other under-actuated systems seems achievable.

As mentioned in the limitations section, there tends to be a non-zero consistency error between the task space and the joint-space trajectories. Alleviating this during training of the VAE will unlock further performance from the current system. To do this, we require an additional loss between the task-space quantities and the estimated task-space poses. The estimated task-space quantities are generated using the predicted joint configurations and integrating the base-pose deltas. Since the robot's kinematics are required to generate and backpropagate this loss, the simplest method would be to integrate a differential simulator into the training process. The differential simulator would facilitate rolling out the trajectories through time to convert the joint-space motions to task space. These task-space predictions could then be compared with the expert trajectories. By reducing the loss between the predicted task space motion the expert would reduce the consistency error.

The learnt latent spaces have an intuitive and low-dimensional structure. As mentioned in the limitations section, the learnt planner introduced in Chapter 5 is rudimentary. This could easily be replaced by a more sophisticated planner with access to not only the latent space but also the robot's base velocity and gait command. With this, the planner could create trajectories with long horizons and optimise the correct gait given the terrain ahead. For example, a planner trained using RL could plan a trajectory using trot as a default gait until reaching some very tall terrain where crawl would be more appropriate. In addition, an RL planner operating in latent space would have a semantically meaningful and low-dimensional latent space which would simplify the training process improving its tractability.

The most recent work in the field of quadruped locomotion focuses on highly dynamic motion over obstacles much larger than the quadruped itself such as $1.5\,\mathrm{m}$ blocks. This requires coordinated jumps as well as integrated perception. As mentioned in the limitations section, the WBC may not be able to track these types of movements. Indeed the WBC could be replaced with an RL tracking controller permitting full in simulation training. The current WBC cannot be easily parallelised due to name-spacing issues. This makes RL in simulation with the WBC prohibitive. Finally, with the VAE and its accessories deployed in a parallelisable simulator, RL could train both the tracking controller and the latent space planner.

One of the advantages of this work is the interpretable latent space. This compact representation accurately captures quadruped dynamics in a compact representation. In theory, this approach should generalise well to other underactuated or complex coordinated systems. For example, a shared latent representation for loco-manipulation is an obvious choice. Research for loco-manipulation is a growing field as these platforms can interact with the world making them extremely useful. The challenges for such systems arise from the additional manipulator dynamics which if unmodelled could destabilise the quadruped. Additionally, there are two planning tasks here: a locomotion problem and a manipulation one. Ideally, we require the quadruped to react to the manipulator dynamics and stabilise the system as the manipulator is operating. As we see in this thesis, latent space planning is an effective method for capturing highly correlated and dynamic movements in a compact planning space. Furthermore, we show a plethora of diverse planning techniques which would be the inspiration for solving loco-manipulation tasks in a suitable latent space.

8
Conclusions

This thesis finds a representation for quadruped locomotion which permits continuous and natural-looking transitions between gaits and perceptive terrain traversal. We first created a structured latent space for static crawl gaits and developed methods for handling complex contact dynamics. Locomotion trajectories were solved in this space using a gradient-based planner. The limitations of this work are addressed in the following chapter leading to dynamic trot gaits with continuously variable characteristics. This latent space is structured and disentangled, and this results from embedding the contact dynamics in the latent space. Furthermore, we discover that trot trajectories are oscillatory signals in space. We inject periodic signals with tunable amplitudes and frequencies to produce trot trajectories. This, in effect, trivialises the contact scheduling as crossing between the clusters which encode different stances. Inspired by the ability to represent a set of trot gaits from a single example, we train a model with multiple gaits. By exploring this new latent space, we discover unseen intermediate gaits. We discover that the latent space encodes the phases between legs across all the gaits and utilises these correlations to create a shared representation across all gaits. We exploit this shared representation to create the gait transitions. This latent space is conditioned using a terrain encoder and a learnt planner is applied in latent space to traverse uneven terrain. The planner modulates the gait's characteristics and elongates the swing lengths aiding the robot when climbing terrain. Lastly, we notice that the periodic oscillations look strikingly similar

to those found in primates' brains during both reaching and locomotion. We investigate this further and lesion our models to understand how information from the compact representation propagates out to create the locomotion trajectories.

References

[1] M. Hutter et al. “ANYmal - A Highly Mobile and Dynamic Quadrupedal Robot”. In: *IEEE/RSJ Int. Conf. Intell. Rob. Sys. (IROS)*. 2016.

[2] Carlos Mastalli et al. *Agile Maneuvers in Legged Robots: a predictive control approach*. 2022.

[3] A. W. Winkler et al. “Gait and Trajectory Optimization for Legged Systems Through Phase-Based End-Effector Parameterization”. In: *IEEE Robot. Automat. Lett. (RA-L)* 3.3 (July 2018), pp. 1560–1567.

[4] Carlos Mastalli et al. “A Feasibility-Driven Approach to Control-Limited DDP”. In: *Autonomous Robots* (2022).

[5] Yuval Tassa, Nicolas Mansard, and Emo Todorov. “Control-limited differential dynamic programming”. In: *2014 IEEE International Conference on Robotics and Automation (ICRA)*. 2014, pp. 1168–1175.

[6] Miomir Vukobratovic and Branislav Borovac. “Zero-Moment Point - Thirty Five Years of its Life.” In: *Int. J. Hum. Rob. (IJHR)* 1 (2004), pp. 157–173.

[7] David Orin, Ambarish Goswami, and Sung-Hee Lee. “Centroidal Dynamics of a Humanoid Robot”. In: *Autonomous Robots* 35 (Oct. 2013).

[8] C Dario Bellicoso et al. “Dynamic locomotion through online nonlinear motion optimization for quadrupedal robots”. In: *IEEE Robot. Automat. Lett. (RA-L)* 3.3 (2018), pp. 2261–2268.

[9] Jemin Hwangbo et al. “Learning agile and dynamic motor skills for legged robots”. In: *Science Robotics* 4.26 (2019).

[10] Siddhant Gangapurwala, Alexander Mitchell, and Ioannis Havoutis. “Guided Constrained Policy Optimization for Dynamic Quadrupedal Robot Locomotion”. In: *IEEE Robot. Automat. Lett. (RA-L)* 5.2 (2020), pp. 3642–3649.

[11] Joonho Lee et al. “Learning quadrupedal locomotion over challenging terrain”. In: *Science Robotics* 5.47 (Oct. 2020).

[12] Takahiro Miki et al. “Learning robust perceptive locomotion for quadrupedal robots in the wild”. In: *Science Robotics* 7.62 (2022), eabk2822.

[13] David Hoeller et al. *ANYmal Parkour: Learning Agile Navigation for Quadrupedal Robots*. 2023.

[14] Ken Caluwaerts et al. *Barkour: Benchmarking Animal-level Agility with Quadruped Robots*. 2023.

[15] Ashish Vaswani et al. *Attention Is All You Need*. 2023.

[16] Maximilian Karl et al. “Deep Variational Bayes Filters: Unsupervised Learning of State Space Models from Raw Data”. In: *Int. Conf. on Learn. Repr. (ICLR)*. 2016.

[17] Manuel Watter et al. “Embed to Control: A Locally Linear Latent Dynamics Model for Control from Raw Images”. In: *NeurIPS*. 2015.

[18] A. L. Mitchell et al. “First Steps: Latent-Space Control with Semantic Constraints for Quadruped Locomotion”. In: *IEEE/RSJ Int. Conf. Intell. Rob. Sys. (IROS)*. 2020, pp. 5343–5350.

[19] Diederik Kingma and Max Welling. “Auto-Encoding Variational Bayes”. In: *Int. Conf. on Learn. Repr. (ICLR)*. 2014.

[20] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. “Stochastic Backpropagation and Approximate Inference in Deep Generative Models”. In: *Int. Conf. on Mach. Learn. (ICML)*. 2014.

[21] Irina Higgins et al. “β-VAE: Learning basic visual concepts with a constrained variational framework.” In: *Int. Conf. on Learn. Repr. (ICLR)*. 2017.

[22] Chia-Man Hung et al. “Reaching Through Latent Space: From Joint Statistics to Path Planning in Manipulation”. In: *IEEE Robotics and Automation Letters* 7.2 (2022), pp. 5334–5341.

[23] Jun Yamada et al. *Leveraging Scene Embeddings for Gradient-Based Motion Planning in Latent Space*. 2023.

[24] Alexander Luis Mitchell et al. “VAE-Loco: Versatile Quadruped Locomotion by Learning a Disentangled Gait Representation”. In: *IEEE Transactions on Robotics* (2023), pp. 1–16.

[25] Mark M. Churchland et al. “Neural population dynamics during reaching”. In: *Nature* 487 (2012).

[26] Alexander L. Mitchell et al. “Next Steps: Learning a Disentangled Gait Representation for Versatile Quadruped Locomotion”. In: *IEEE Int. Conf. Rob. Autom. (ICRA)*. 2022, pp. 10564–10570.

[27] G. A. Pratt and M. M. Williamson. “Series Elastic Actuators”. In: *IEEE/RSJ Int. Conf. Intell. Rob. Sys. (IROS)*. 1995.

[28] Gerardo Bledt et al. “MIT Cheetah 3: Design and Control of a Robust, Dynamic Quadruped Robot”. In: Madrid, Spain: IEEE Press, 2018, pp. 2245–2252.

[29] Xue Bin Peng et al. “DeepLoco: Dynamic Locomotion Skills Using Hierarchical Deep Reinforcement Learning”. In: 36.4 (2017).

[30] Nguyen Thanh Phuong et al. “An optimal control method for biped robot with stable walking gait”. In: *IEEE Int. Conf. Hum. Rob. (ICHR)*. 2008, pp. 211–218.

[31] A. W. Winkler et al. “Fast Trajectory Optimization for Legged Robots Using Vertex-Based ZMP Constraints”. In: *IEEE Robot. Automat. Lett. (RA-L)* 2.4 (Oct. 2017), pp. 2201–2208.

[32] M. Mistry, J. Buchli, and S. Schaal. “Inverse Dynamics Control of Floating Base Systems using Orthogonal Decomposition”. In: *IEEE Int. Conf. Rob. Autom. (ICRA)*. 2010.

[33] C. Dario Bellicoso et al. “Dynamic locomotion and whole-body control for quadrupedal robots”. In: *IEEE/RSJ Int. Conf. Intell. Rob. Sys. (IROS)*. 2017, pp. 3359–3365.

[34] William Uther. “Markov Decision Processes”. In: *Encyclopedia of Machine Learning*. Ed. by Claude Sammut and Geoffrey I. Webb. Boston, MA: Springer US, 2010, pp. 642–646.

[35] Thomas M. Moerland et al. 2023.

[36] John Schulman et al. *Proximal Policy Optimization Algorithms*. 2017.

[37] John Schulman et al. "Trust Region Policy Optimization". In: *CoRR* (2015).

[38] Sergey Levine and Vladlen Koltun. "Learning Complex Neural Network Policies with Trajectory Optimization". In: *Proceedings of the 31st International Conference on Machine Learning*. Ed. by Eric P. Xing and Tony Jebara. Vol. 32. Proceedings of Machine Learning Research 2. Bejing, China: PMLR, 22–24 Jun 2014, pp. 829–837.

[39] Jemin Hwangbo, Joonho Lee, and Marco Hutter. "Per-contact iteration method for solving contact dynamics". In: *IEEE Robot. Automat. Lett. (RA-L)* 3.2 (2018), pp. 895–902.

[40] Rika Antonova et al. "Bayesian Optimization in Variational Latent Spaces with Dynamic Compression". In: *Proceedings of the Conference on Robot Learning*. Ed. by Leslie Pack Kaelbling, Danica Kragic, and Komei Sugiura. Vol. 100. Proceedings of Machine Learning Research. PMLR, 30 Oct–01 Nov 2020, pp. 456–465.

[41] Siddhant Gangapurwala et al. "RLOC: Terrain-Aware Legged Locomotion Using Reinforcement Learning and Optimal Control". In: *IEEE Transactions on Robotics* 38.5 (2022), pp. 2908–2927.

[42] Yuxiang Yang et al. "Fast and Efficient Locomotion via Learned Gait Transitions". In: *Conf. on Rob. Learn. (CoRL)*. 2021.

[43] Jinze Wu, Yufei Xue, and Chenkun Qi. *Learning Multiple Gaits within Latent Space for Quadruped Robots*. 2023.

[44] Oliwier Melon et al. "Reliable Trajectories for Dynamic Quadrupeds using Analytical Costs and Learned Initializations". In: *IEEE Int. Conf. Rob. Autom. (ICRA)*. 2020.

[45] Hari Teja Kalidindi et al. "Rotational dynamics in motor cortex are consistent with a feedback controller". In: *eLife* 10 (2021), e67256.

[46] Nikita Rudin et al. "Advanced Skills by Learning Locomotion and Local Navigation End-to-End". In: *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2022, pp. 2497–2503.

Appendices

A

Next Steps: Learning a Disentangled Gait Representation for Versatile Quadruped Locomotion

Next Steps: Learning a Disentangled Gait Representation for Versatile Quadruped Locomotion

Alexander L. Mitchell, Wolfgang Merkt, Mathieu Geisert, Siddhant Gangapurwala, Martin Engelcke, Oiwi Parker Jones, Ioannis Havoutis, and Ingmar Posner

***Abstract*— Quadruped locomotion is rapidly maturing to a degree where robots now routinely traverse a variety of unstructured terrains. However, while gaits can be varied typically by selecting from a range of pre-computed styles, current planners are unable to vary key gait parameters *continuously* while the robot is in motion. The synthesis, on-the-fly, of gaits with unexpected operational characteristics or even the blending of dynamic manoeuvres lies beyond the capabilities of the current state-of-the-art. In this work we address this limitation by learning a latent space capturing the key stance phases constituting a particular gait. This is achieved via a generative model trained on a single trot style, which encourages disentanglement such that application of a *drive signal* to a single dimension of the latent state induces holistic plans synthesising a continuous variety of trot styles. We demonstrate that specific properties of the drive signal map directly to gait parameters such as cadence, footstep height and full stance duration. Due to the nature of our approach these synthesised gaits are continuously variable online during robot operation and robustly capture a richness of movement significantly exceeding the relatively narrow behaviour seen during training. In addition, the use of a generative model facilitates the detection and mitigation of disturbances to provide a versatile and robust planning framework. We evaluate our approach on a real ANYmal quadruped robot and demonstrate that our method achieves a continuous blend of dynamic trot styles whilst being robust and reactive to external perturbations.**

I. Introduction

Quadruped locomotion has advanced significantly in recent years, extending their capability towards applications of significant value to industry and the public domain. Driven primarily by advances in optimisation-based [1]–[4] and reinforcement learning-based methods [5]–[7], quadrupeds are now able to robustly plan traversals over a wide variety of terrains, which makes them an increasingly popular choice for tasks such as inspection, monitoring, search and rescue or goods delivery in difficult, unstructured environments. However, despite recent advances, important limitations remain. Due to the complexity of the system, models used for gait planning and control are often overly simplified and handcrafted for particular gait types (e.g. crawl, trot, gallop, e.g. [1], [8]). In the worst case, this can limit the versatility of the robot as the models deployed are failing to exploit the full capability of the underlying hardware (e.g. [1], [9], [10]). Furthermore, current gait planners are unable to vary key parameters (such as step height) of a gait continuously and online [3], [4]. Instead, contact schedules for discrete gaits are often pre-computed and selected as needed using hierarchical approaches that rely upon a sequence of optimisations, yielding only a narrow operating window which severely limits their response to external disturbances [3], [4]. The synthesis of gaits which fall outside of this set is typically cumbersome, as approaches that optimise dynamics over gait schedules and footstep lengths or heights are computationally expensive [3], [4], and often require manual intervention [1]. In contrast, the ability to control key gait parameters – such as cadence, swing height, and full-support duration – *on-the-fly* would enable a smooth interpolation between dynamic manoeuvres allowing for swift reaction to external stimuli. This leads to significantly more versatile locomotion.

All authors are with the Oxford Robotics Institute, University of Oxford
Email: mitch@robots.ox.ac.uk

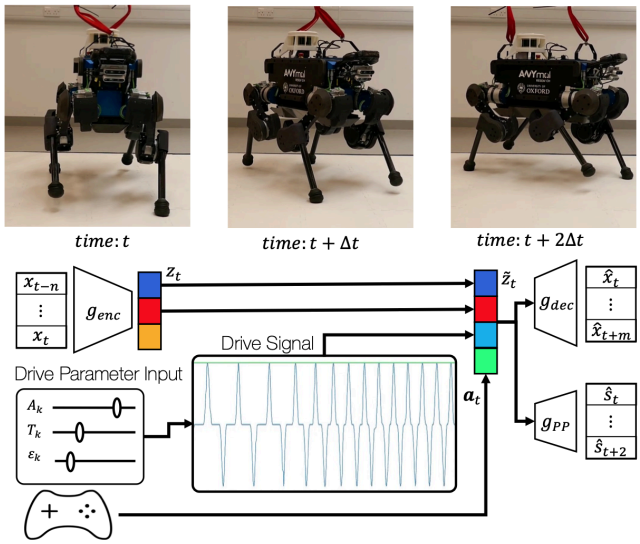


Fig. 1: Using a variational auto-encoder (VAE), our approach learns a structured latent space capturing key stance phases constituting a particular gait. The space is disentangled to a degree such that application of a *drive signal* to a single dimension of the latent variable induces gait styles which can be seamlessly interpolated between. The drive signal's amplitude and phase provide continuous control over the gait parameters such as cadence, full-support duration and foot swing height. To instantiate closed-loop control, we encode raw sensor information to infer the robot's gait phase using g_{enc} and append an action vector, which expresses the desired base twist, to the augmented latent variable before decoding via g_{dec}, and predicting the feet in contact using g_{pp}. Consequently, planning in latent space leads to user-defined and continuously variable trot locomotion.

Inspired by recent work on a quadruped that achieves a crawl gait via the traversal of a learned latent space [11], we approach the challenge of continuous contact schedule variation from the perspective of learning and traversing a structured latent-space. This is enabled by learning a *generative model* of locomotion data which, in addition to capturing relevant structure in the latent space, also enables the detection and mitigation of disturbances to provide a versatile and robust planning framework. In particular, we train a variational auto-encoder (VAE) [12], [13] on short sequences of state-space trajectories taken from a single gait type (trot), and predict a set of future states. We show that the resulting latent space has an interpretable structure, lending itself to the generation of a variety of trot styles, depending on how the latent space is traversed. In fact, examining trot trajectories in latent space reveals an oscillatory drive signal which controls fundamental aspects of the gait. We subsequently find that by overwriting this trajectory with a synthetic drive signal, we can *continuously* control the robot's gait properties whilst the robot is executing the motion. Parameters of this drive signal can be mapped explicitly to gait parameters such as cadence, footstep height, and full-stance support duration. We emphasise that this ability to generalise over gait styles emerges from training on a single gait type: a trot gait with constant parameters.

We illustrate the efficacy of our approach by generating a range of continuously blended trajectories on a real ANYmal quadruped robot – a medium-sized platform ($35\,\mathrm{kg}$) standing $0.5\,\mathrm{m}$ tall. While the latent space is learnt using examples only from a specific gait style, our approach is able to synthesise behaviours significantly beyond this training distribution.

In addition, we leverage our generative approach to both characterise and react to external perturbations. A large impulse applied to the robot's base triggers a spike in the Evidence Lower Bound (ELBO) which clearly identifies the disturbance as out of the distribution seen during training. Inspired by [14], which states that an increase in cadence is both a response to slip and a form of push recovery in humans, our planner automatically increases the robot's cadence to aid in counteracting the disturbance and demonstrate a marked improvement in robustness.

To the best of our knowledge, our method is the first which allows continuous variation of the robot's gait characteristics whilst the robot is walking. It provides a versatile and data-driven approach to quadruped locomotion which additionally allows for disturbance detection and recovery. Beyond these contributions, our results are tantalising as the notion of the drive signal is reminiscent of central pattern generators (CPGs) located in the spinal cord and brain stem of vertebrates [15]. Both CPGs and our synthetic drive signal generate cyclic patterns that effect holistic motor activity and stereotyped rhythmic behaviours such as locomotion.

II. Related Work

Planning and control for quadruped locomotion has advanced in leaps and bounds in recent years. A seminal work in these areas is *Dynamic Gaits* (DG) [1]. DG enables a quadrupedal robot (ANYmal) to execute a wide variety of dynamic gaits (e.g. trot, pace, lateral walk, jump) with real-time motion planning and control. However, to achieve this impressive range of behaviours, DG provides each gait type with its own contact schedule and utilises an environment-specific footstep planner, ultimately limiting its capability.

Latent space approaches for planning and control learn useful and typically low-dimensional representations that can be used to control complex dynamics, without relying on known system models. Classic examples include *Deep Variational Bayes Filters* (DVBF) [16] and *Embed to Control* (E2C) [17]. DVBF produces dynamically consistent trajectories by traversing continuous paths in latent space whilst E2C learns a linear system model in which control problems can be solved. *Conditional Neural Movement Primitives* (CNMP) [18] is a more recent latent space approach for robotic arms that generalises between a variety of tasks, such as pick-and-place and obstacle avoidance. Other recent works like UPN and PlaNet [19], [20] show impressive capabilities in simulation but are yet to be applied to real-world systems, including floating-base robots.

In the quadruped domain, *First Steps* [11] learns a structured latent space based on feasible robot *configurations* and then defines a set of performance predictors that can be used in an optimisation framework to control the robot. In practice, these performance predictors can be viewed as symbolic inputs (e.g. 'left front leg up') but drive the robot in continuous space. However, because *First Steps* is trained on static snapshots of robot configurations, it does not learn from observable dynamics and may thus require more explicit structuring of the system than is necessary. Our work addresses this shortcoming and significantly extends this framework to effective and robust closed-loop planning and control.

The *Motion VAE* (MVAE) [21] learns to represent dynamic trajectories in a structured latent space for the locomotion of computer-animated humanoids. This is similar to our emphasis here on learning representations for dynamic trajectories in the context of locomotion. However, moving from simulated to real physical systems, as is required for robotic applications, necessitates tackling additional complexities like latency, hard real-time requirements, and actuator dynamics. In this work, we tackle these challenges and demonstrate that a single gait style contains sufficient richness to learn a structured latent space that can be exploited to manipulate gait characteristics that generalise beyond the range seen during training. Thus, unlike MVAE, our approach does not train on multiple gait styles, despite succeeding in producing them.

Finally, a study conducted concurrently to our own [22] yields variation between gait types (walk and trot). It utilises a reinforcement learning (RL) approach which employs a phase iterator similar to our drive signal. However, this phase iterator is enforced, whilst our gait dynamics are discovered automatically purely from exposure to trot trajectories with constant parameters.

III. METHOD

Our aim is to use unsupervised learning to infer a structured latent-space which facilitates real-time and smooth variation of key gait parameters. We conjecture that structure can emerge from the exposure of a suitable generative model to a gait with predetermined and constant characteristics such as cadence, swing height, and full-support duration. Specifically, we propose that due to this structure, continuous latent trajectories result in robot locomotion. By inspecting this structure, we discover a disentangled latent-space where gait parameters are axis-aligned within this space. Periodic trajectories in latent space can then be decoded back to smooth robot locomotion as depicted in Fig. 1. Subsequently, the VAE is deployed as a planner in a real-time control loop.

VAE Architecture: We train a VAE [12], [13] to create a structured latent-space using observed dynamic data. The input to the VAE $\mathbf{X}_k$ at time step k consists of N robot states sampled from simulated trot gaits with constant parameters (e.g. cadence and foot-step height). These state-space quantities are values we wish either to control or are required to infer the gait phase. These are joint angles; end-effector positions in the base frame; joint torques; contact forces; the base velocity; and the base pose evolution relative to a control frame, which is updated periodically. These quantities are denoted as $\mathbf{x}_k = [\mathbf{q}_k, \mathbf{ee}_k, \tau_k, \lambda_k, \dot{\mathbf{c}}_k, \Delta\mathbf{c}_k]$, where k is the time step. Note that velocities and accelerations do not form part of the VAE's input, but are inferred from the input history. This has dual benefits: first, it yields a lower-dimensional input space; and, second, it prevents sensitivity to fast-changing quantities such as the recorded joint accelerations during inference.

To deploy the VAE-planner in a closed-loop framework, we encode the input history at the control frequency f_c. However, due to restrictions on the VAE's size caused by the tight computation bounds required for real-time control, the encoder input $\mathbf{X}_k$ is constructed using states sub-sampled at a lower frequency f_{enc}:

$$\mathbf{X}_k = [\mathbf{x}_{k-r(N-1)}^\top, \ldots, \mathbf{x}_{k-r}^\top, \mathbf{x}_k^\top], \tag{1}$$

where $r = f_c/f_{\text{enc}}$ is the ratio between the control and encoder frequencies. The input $\mathbf{X}_k$ is created every time step by sampling from an input buffer which stores every robot state $\mathbf{x}$ from time step k to $k - r(N-1)$ at the control frequency.

The VAE's decoder output $\hat{\mathbf{X}}_k^+$ predicts the current robot state $\mathbf{x}_k$ as well as M future ones sampled at a frequency of $f_{\text{dec}} = f_c$:

$$\hat{\mathbf{X}}_k^+ = [\hat{\mathbf{x}}_k^\top, \hat{\mathbf{x}}_{k+1}^\top, \ldots, \hat{\mathbf{x}}_{k+M}^\top] \tag{2}$$

As the *desired-feet-in-contact* is an input to the tracking controller, we also want to predict which feet are in contact, $\mathbf{s}_k$, at the current time step, k, as well as J steps in the future. Inspired by *First Steps* [11], we therefore utilise a feet-in-contact performance predictor $g_{pp}(\mathbf{z}_k)$. This is attached to the latent space, which estimates the probability of each foot being in contact:

$$\hat{\mathbf{S}}_k = [\hat{\mathbf{s}}_k^\top, ..., \hat{\mathbf{s}}_{k+J-1}^\top]^\top \tag{3}$$

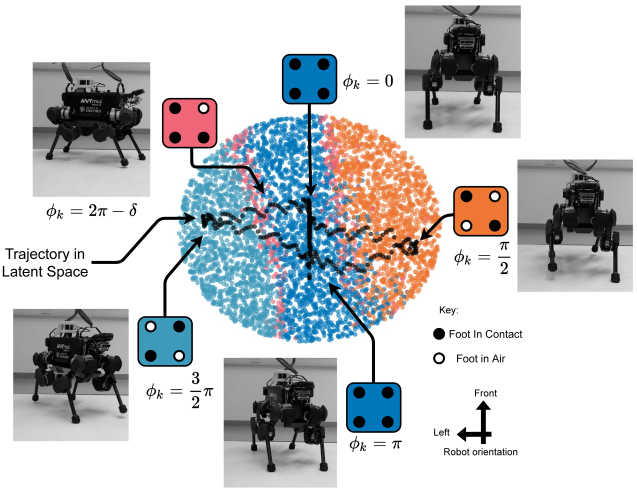


Fig. 2: A slice through the structured latent space, colour coded to illustrate the ordering and clustering of the distinct stances which make up the trot gait. The component of the latent-space trajectory (black) along the horizontal axis contributes to the robot's footstep height, whilst the vertical component gives rise to the footstep length. Snapshots of the robot controlled using the VAE-planner illustrates the interplay between these two latent dimensions.

To command the base twist of the robot, a high-level action command $\mathbf{a}_k$ is utilised. This represents longitudinal (x), lateral (y), and yaw (θ) twist in the robot's base frame. The latent state $\mathbf{z}_k$ and the action $\mathbf{a}_k$ form the input to the decoder.

Training the VAE: We train the VAE and performance predictor together. The VAE's training loss is the standard ELBO formulation consisting of a reconstruction loss (mean-squared error) plus the Kullback–Leibler (KL) divergence D_{KL} between the inferred posterior $q(\mathbf{z}|\mathbf{X}_k)$ and the prior $p(\mathbf{z})$, multiplied by a hyper-parameter β:

$$\mathcal{L}_{\text{ELBO}} = \text{MSE}(\mathbf{X}_k^+, \hat{\mathbf{X}}_k^+) + \beta D_{\text{KL}}[q(\mathbf{z}|\mathbf{X}_k)||p(\mathbf{z})]. \tag{4}$$

These ELBO terms are then then summed with the scaled binary cross-entropy loss between the predicted feet in contact and the recorded ones, such that the overall loss is computed as

$$\mathcal{L} = \mathcal{L}_{\text{ELBO}} + \gamma \text{BCE}(\mathbf{S}_k, \hat{\mathbf{S}}_k) \tag{5}$$

The VAE training loss (Eq. 4), as seen in prior work [23], is responsible for any subsequent disentanglement found in the latent space. The reconstruction error is weighed against the decomposition of the latent space using the hyper-parameter β. This constraint encourages an efficient latent representation, containing only the required information for reconstruction, whilst also reducing the channel capacity. As shown in [23], the D_{KL} term used with an isotropic unit Gaussian ($p(\mathbf{z}) = \mathcal{N}(\mathbf{0}, \mathbf{I})$) encourages conditional independence within $\mathbf{z}$.

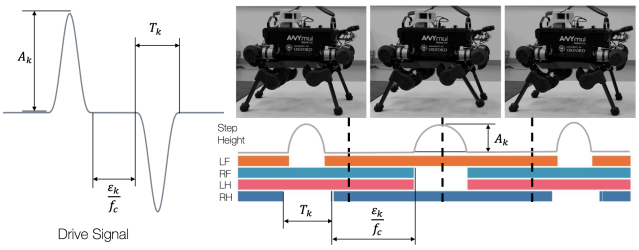


Fig. 3: An oscillatory *drive signal* overwrites the dimension with the smallest variance in the structured latent-space. The amplitude A_k, time-period T_k and stance duration counter ϵ_k of this signal control the robot's foot swing height, cadence and full-support duration in real time.

In our approach, as well as that of *First Steps* [11], a structured latent space is encouraged by backpropagating gradients from the performance predictor's loss through to the encoder input. We hypothesise that useful structuring of this space is inferred from the continuous trajectories used for training, and, in contrast to *First Steps*, no explicit labelling for each stance is required.

A. Investigating the Latent Space

Before solving for locomotion trajectories by planning in latent space, we examine the space to see what structure exists within. Disentangled features are discovered inside.

Latent Space Structure: Fig. 2 shows samples from the latent space colour-coded by their predicted stance. For trot there are four stances: the full-support phase, left front and right hind in contact, another support phase and finally right front plus left hind in contact. Fig. 2 reveals that the latent space has emerged clustered by stance and that, due to the ordering of these stances, a periodic trajectory decodes to a trot gait. This favourable structure is inferred from the continuous trot trajectory input during training.

The Effect of Disentanglement: By examining the latent variables, we discover that oscillations injected into just two dimensions in the latent space decode to continuously varying trot trajectories. This result stems from a latent space where variation in footstep length and cadence is aligned along one dimension, while variation in footstep distance lies along another. Specifically, the time period of what we denote the *drive signal* oscillation controls the robot's cadence, whilst its amplitude is proportionate to the footstep height. In addition, the amplitude of the second signal, which is $\pi/2$ out of phase with the drive signal, controls foot swing length and as such is denoted as the *trot signal*. Given that other work has tried to explicitly build this structure into locomotion systems [22], it is important to emphasise that this disentanglement *emerges* in our study as a result of the training paradigm and data.

B. Control Over the Gait Parameters

The disentangled latent space is exploited such that we can control the robot's gait parameters, specifically the robot's cadence, footstep height, and full-support duration. We therefore design a specific *drive signal*. In particular, we utilise a modified $\sin^3$ oscillation as depicted in Fig. 3 with commandable amplitude A_k, and phase ϕ_k:

$$\mathbf{z}_{k,d_z} = A_k \sin^3(\phi_k). \tag{6}$$

The amplitude A_k controls the foot swing height, and the phase ϕ_k governs cadence and support duration.

To control the robot's swing and stance duration separately, we set the drive signal's time-period T_k and we employ a stance counter ϵ_k. The time-period T_k is equal to the swing duration, whilst the time that the drive signal is equal to zero is extended by ϵ_k time steps to introduce a full-support duration of $(\epsilon_k/f_c)\,\mathrm{s}$. Hence, once both T_k and ϵ_k are used together, the phase dynamics are:

$$\phi_{k+1} = \begin{cases} \phi_k & \text{if } \phi_k \bmod \pi = 0 \text{ and } k_\epsilon < \epsilon_k \\ \phi_k + 2\pi/T_k & \text{otherwise} \end{cases} \tag{7}$$

and, in tandem, the counter ϵ_k is updated as:

$$k_\epsilon \leftarrow \begin{cases} k_\epsilon + 1 & \text{if } \phi_k \bmod \pi = 0 \text{ and } k_\epsilon < \epsilon_k \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

C. Planning for Closed-Loop Control

The VAE is fast enough to act as a planner in a closed-loop controller. Therefore, our approach can react to external disturbances and mitigate against real-world effects such as unmodelled dynamics and hardware latency.

For closed-loop control, we begin by encoding a history of robot states from the raw sensor measurements to infer the current gait phase. Since the encoder and decoder operate at different frequencies, we store a buffer of robot states sampled at the control frequency. This is sub-sampled at f_{enc} to create the encoder's input.

Next, once we have an estimate of the current latent variable, we overwrite latent dimension d_z with the drive signal (see Sec. III-B). In addition, we employ a second-order Butterworth filter to smooth the latent trajectory and further smooth the locomotion plan. In essence, the drive signal encourages the decoder to output the next open-loop prediction while the other latent variables infer the gait phase from the raw sensor input. This process is repeated every control step i.e. at $400\,\mathrm{Hz}$.

The latent variable $\mathbf{z}_k$ and a desired base twist $\mathbf{a}_k$ are decoded to produce $\hat{\mathbf{X}}_k^+ = g_{\mathrm{dec}}(\mathbf{z}_k, \mathbf{a}_k)$. From this, the joint-space trajectory $\hat{\mathbf{Q}}_k$, and local base velocity $\hat{\dot{\mathbf{C}}}_k$ are extracted and derived or integrated to produce the base and joint positions, velocities and accelerations. These quantities along with the predicted contact schedule are sent to the Whole-Body Controller (WBC) [24].

The WBC solves a hierarchical optimisation problem to calculate the joint torques which are sent to the actuators. At the same time, the WBC enforces a series of constraints: contact creation, friction constraints and torque limits. Additionally, the WBC applies forward kinematics to the VAE's trajectory to track it in task-space. However, the WBC does not compensate for infeasible plans.

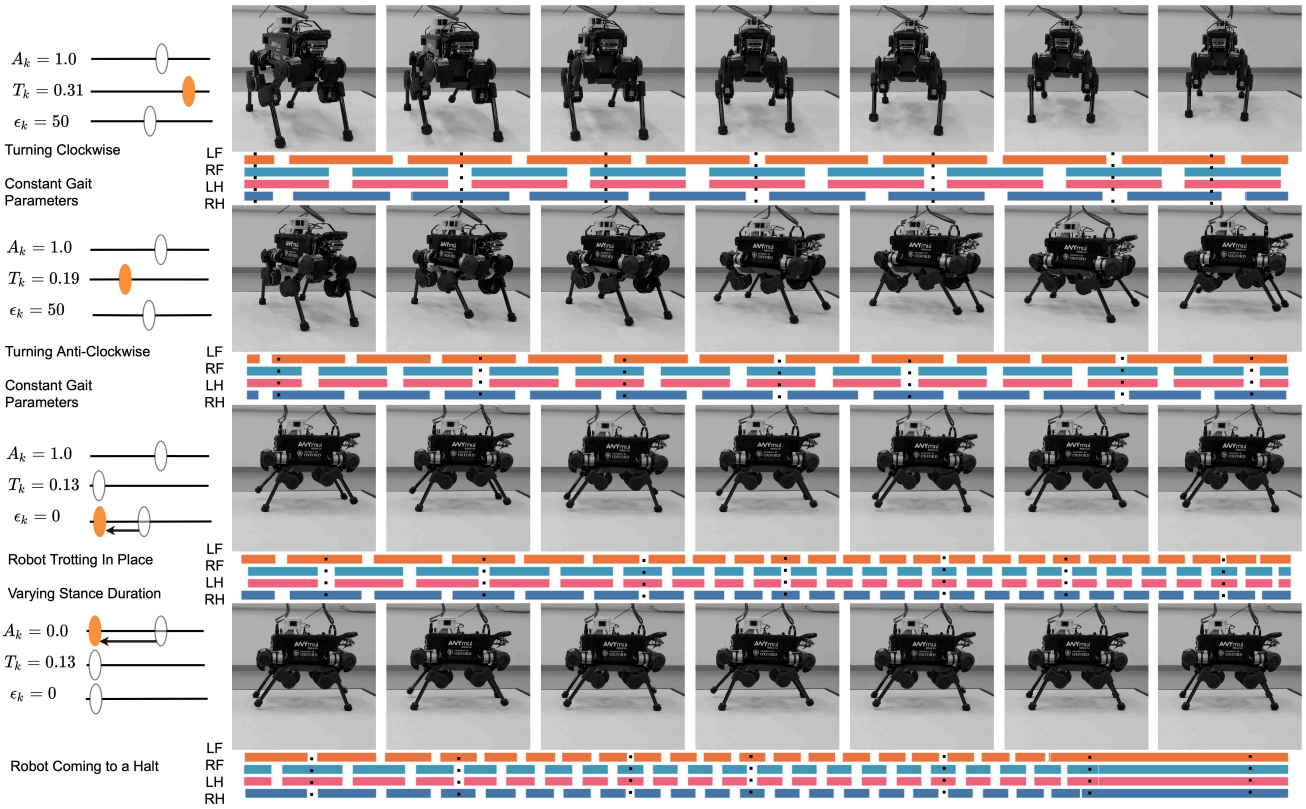


Fig. 4: Closed-loop control of the real ANYmal quadruped using our VAE-planner. This demonstrates user-controlled variation of gait parameters on the fly. Here, coloured rectangles represent the full-stance phase, whilst white space denotes the swing duration. The top row shows a trot gait with an introduced quadrupedal stance phase (gait cycle of $0.75\,\mathrm{s}$ – swing $312.5\,\mathrm{ms}$, stance $62.5\,\mathrm{ms}$; a gait cycle consists of a swing phase for each of the leg pairs). Next, the swing duration T_k is reduced using the time-period slider (gait cycle of $0.5\,\mathrm{s}$ – swing $188\,\mathrm{ms}$, stance $62.5\,\mathrm{ms}$). The third row illustrates the effect of reducing the stance duration counter ϵ_k to produce trot with reducing full-stance phases (gait cycle of $250\,\mathrm{ms}$ – swing $125\,\mathrm{ms}$, stance $0.0\,\mathrm{s}$). Finally, transition into standing occurs when the drive signal amplitude A_k is reduced to zero.

Disturbance Detection and Response: Our approach is able to both detect and react to disturbances. The VAE is trained using canonical feasible trajectories. Therefore, any disturbances are characterised as out of distribution with respect to the training set. Given the generative nature of our approach, this discrepancy is quantified during operation by the trained model via the Evidence Lower-Bound (ELBO, Eq. 4 where β is set to one). We will show in the evaluation (Sec. IV-F) that even a rudimentary response strategy serves to increase the range of disturbance the system can reject.

IV. Experimental Results

In this section, we demonstrate the capabilities of the VAE as a flexible and robust locomotion-planner deployed for closed-loop control on a real robot. In doing so, we investigate (i) the structure induced in the latent space (Sec. IV-B), (ii) the sensitivity of our approach to variations in key hyper parameters (Sec. IV-C), (iii) to what extent the locomotion parameters can be varied online (Sec. IV-D), (iv) the feasibility of the locomotion plans produced (Sec. IV-E); and, finally, (v) the degree to which disturbance detection, coupled with a rudimentary recovery strategy, further increases the robustness of our approach (Sec. IV-F). Please see the following video for an extended set of experiments along with a brief description of our approach (`https://youtu.be/2XMyAAe7yDs`).

A. Details on Training and Deployment

Before incorporating our VAE-planner into a closed-loop controller and evaluating the results as deployed on the real ANYmal robot, we discuss the following: firstly, how the dataset is constructed; secondly, how the VAE is trained; and lastly, implementation details for transfer to the real robot.

Dataset Generation: To train the VAE and create the structured latent space, we require a set of continuous trot trajectories. Therefore, we employ the gait generation framework *Dynamic Gaits* (DG) [1] and solve for trot gaits which track a user-defined base twist command. DG is a hierarchical planning and control framework which is used with a *fixed* contact schedule and predefined footstep heights. Specifically, swing, full-stance durations and footstep height remain constant and are set to $0.5\,\mathrm{s}$, $75\,\mathrm{ms}$ and $0.10\,\mathrm{m}$, respectively. This is the only gait style used for training. The main components of DG are a footstep planner, a base motion planner and a Whole-Body Controller. The footstep planner computes the next four steps over the

gait period using an inverted-pendulum model. Using the footstep positions and schedule, the base motion planner computes the base trajectory over the gait period using a centroidal dynamics model [10] constrained with a Zero-Moment Point (ZMP) [9] criterion. The Whole-Body Controller [24] described in Sec. III-C then converts the task space trajectories to joint feedforward torques, reference positions and velocities, which are sent to the actuators.

The dataset is generated by uniformly sampling desired base twist and executing DG in the *RaiSim* physics simulator [25]. To improve the fidelity of the simulation, the dynamics of the Series-Elastic Actuators (SEA) [26] in the ANYmal's joints are modelled using an *actuator network* [5]. This is essential for good performance as the input response of SEAs depends on a history of states, inputs, and the low-level control law. The specific network used here is found in [7], and takes into account the commanded positions, velocities, feed-forward torques, and low-level PD gains.

Architecture Details: In all our experiments, the VAE used has encoder, decoder and stance performance predictor networks with two hidden layers and widths of 256 units, as well as ELU non-linearities [27]. The encoder operates on data sampled at a frequency of $f_{\text{enc}} = 200\,\text{Hz}$, whilst the decoder's output is fed directly to the tracking controller at the control frequency of $f_{\text{dec}} = 400\,\text{Hz}$. The encoder input is created using $N = 80$ robot states sampled at $200\,\text{Hz}$ – representing a history of $0.4\,\text{s}$ – from the encoder input, which is of size 5120 units. The input is compressed via a latent space of 125 units which is concatenated with an action of 3 units. Only a few future states are required: the decoder outputs the current state and the next $M = 19$ robot states (preview horizon $47.5\,\text{ms}$, output size: 1216 units). The performance predictor predicts the current feet in contact and two future states. Finally, hyper-parameters used for training are $\beta = 1.0$, $\gamma = 0.5$, with a learning rate of 1×10^{-3} using the Adam optimiser. Training is terminated after 1×10^{6} gradient steps.

Domain Transfer: To achieve domain transfer and deployment on the real robot, some modifications are required. The contact forces are artificially set to zero when the robot measures no contact (as determined by a probabilistic contact estimator). This is necessary since the real-world ANYmal robot measures large contact forces even during swing motion, as these forces are inferred from torque residuals and are affected by model error. In contrast, simulators estimate no contact force during swing. As mentioned in Sec. III-C, a Butterworth filter with a cutoff frequency of $10\,\text{Hz}$ is employed to smooth the latent trajectory. Due to the strict computation budget for real-time control, inference times for the VAE is restricted to at most $1\,\text{ms}$, which imposes constraints on the capacity of the model (see Sec. IV-C for an ablation of model hyperparameters). Our largest model takes approximately $1\,\text{ms}$ for the VAE computation, which is roughly equal to the computation time of the WBC.

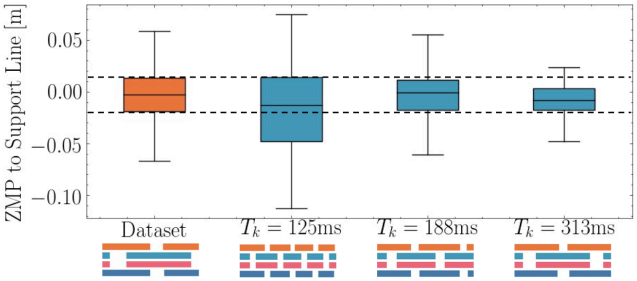


Fig. 5: Distribution of the signed distance between the ZMP and the support line during the 2-stance phases (i.e., one pair of legs was in swing). In orange, the distribution with upper and lower bounds (dashed lines) from the dataset which uses a swing duration of $T_k = 0.5\,\text{s}$. In blue, the distribution of the VAE trajectories shown over a range of modes deployed on the real robot.

B. Structure Induced In The Latent Space

The latent space is inspected to discover what structure exists and if any locomotion properties are disentangled within. Fig. 2 depicts the latent space and is plotted such that the drive signal dimension is along the horizontal, and trot signal dimension is aligned with the vertical. In addition, the stances (coloured areas) are distinctly clustered in sequence of the trot gait, whilst the user-controlled drive-signal, and the inferred trot-signal (black trajectory) move through latent space in a figure of eight visiting each stance cluster sequentially. See Sec. III-A for detailed discussions.

C. Sensitivity To Hyper Parameters

The size of the latent dimension is reduced from 125 to a minimum of 6 in strides of 32 units. The latent dimension of 6 units deployed in simulation and the real robot produces trajectories which are stable and well tracked by the WBC. Next, the input history is reduced from $0.4\,\text{s}$ to $0.3\,\text{s}$ sampled at $200\,\text{Hz}$, resulting in poor open-loop performance. Since the gait's swing duration in training lasts $0.45\,\text{s}$, an input history long enough to capture this is necessary for the model to infer the gait phase. Next, we investigated reducing the encoder's sampling frequency f_{enc} by halving it to $100\,\text{Hz}$ whilst the history remains sampled over $0.4\,\text{s}$. Though this speeds up inference, the resulting trajectories are less smooth than the $200\,\text{Hz}$ encoder and the robot is not stable. Finally, the VAE's width is reduced in 32 unit increments to reduce the channel capacity until the VAE's trajectories cause the robot to become unstable in simulation. This limit is reached at a width of 128 units, corresponding to a reduction in channel capacity by $52.8\,\%$. We verified these models on the robot showcasing that a minimal model requires a latent space of 6 units, an input history of $0.4\,\text{s}$ sampled at $200\,\text{Hz}$, and a hidden layer size of 128 neurons.

D. Varying Locomotion Parameters Online

We leverage the disentangled latent-space to smoothly transition between gait parameters whilst the robot is walking. Crucially, cadence, stance duration, and footstep height can be varied during any phase of the gait by modulating

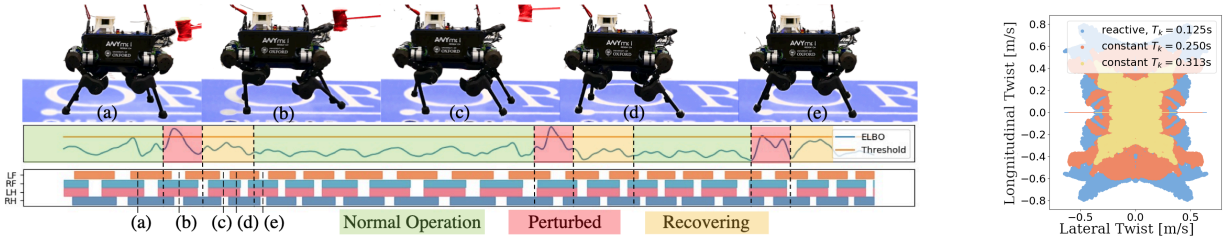


Fig. 6: Push recovery following a shove to the base. A disturbance to the robot causes the ELBO of the form in Eq. 4 to rise above a predetermined threshold (red areas). This identifies a disturbance and triggers an increase in cadence from 250 ms to 125 ms as a rudimentary response. This reduction in the swing duration is visible in the contact schedule. The right sub-plot shows how the range of rejectable base-velocity disturbances increases when automatically modulating the cadence. This plot is symmetrical for ease of view. The range of swing durations plotted is 313 ms, 250 ms, and 125 ms. To appreciate the results fully, please see the following video `https://youtu.be/2XMyAAe7yDs`.

the drive signal's parameters (see Sec. III-B). This results in operating modes which vary from those seen during training. Examples of walking motion using the VAE-planner on the real robot are shown in Fig. 4.

Swing Duration: The swing duration is varied over a large operating window on the ANYmal robot. This begins with a swing time-period starting at 312.5 ms, and is smoothly varied until the swing duration reaches 125 ms, effecting a faster step rate. In parallel, we alter the robot's heading, demonstrating the independence of the action and the latent-space dynamics. Specifically, the top row of Fig. 4 shows the nominal swing duration of 312.5 ms as the robot turns clockwise, tracking a constant angular velocity command. Following this, we demand a slightly faster swing 188 ms and a constant angular velocity anti-clockwise, before transitioning to the fastest swing (125 ms) in the third row. Here, the coloured contact schedule captures the changes in swing duration as it occurs in real-time.

Stance Duration: Following a successful reduction in cadence, the stance duration is reduced and the robot transitions into a trot with negligible full-stance phase: $\epsilon_k = 0$. Trot gaits with little to no full-support phase are particularly challenging manoeuvres for the system in general, as there is reduced control authority to correct for accrued base pose error. During the swing phase of this gait style, only feet across the diagonal are in contact resulting in a line contact limiting the robot's ability to steady its base. The transition to a negligible full-stance phase is captured in the third row, where the coloured stance duration reduces in length.

Footstep Height: We vary the amplitude of the drive signal smoothly to zero as is seen in the bottom row (Fig. 4): The footstep height reduces to zero as the white-space in the contact schedule disappears and the robot remains standing. Beyond versatility, e.g. to increase swing heights to overcome irregular ground height, this capability further enables a safe, smooth and natural transition into and out of the VAE control mode (i.e. to start and come to a halt).

E. Evaluating Dynamic Feasibility Of Planned Motions

Although we have shown that we can smoothly interpolate between a variety of different trot gait parameters continuously, we inspect the VAE's trajectories post experiment to see if they are dynamically feasible. To evaluate the dynamic feasibility of trajectories synthesised by our VAE, we measure the distance of the Zero Moment Point (ZMP) [9] to the support line (i.e. during 2-stance phases) and then compare this distance with that in the dataset. This is a commonly used criterion in model-based legged robot control [1], [9].

The comparison of the ZMP from trajectories generated by the VAE-planner and DG is summarised in Fig. 5. The distribution the ZMP positions is similar for both DG (dataset) and a range of the VAE-planner's operating modes. Crucially, this remains true despite the maximum swing duration generated by the VAE-planner being up to 3.2 times faster than in the training data. We conclude that the representation is good enough to generalise to robot's dynamics shown here. Empirically, we have further been able to steer the robot with arbitrary and fast changing input actions for x, y, and yaw rates, issued from a remote control while being able to interpolate the gait style.

F. Disturbance Detection and Recovery

The ELBO can be used to detect disturbances as it is a lower bound for the evidence of a sample given a particular distribution. The distribution in question is that learned over the training data from DG. Therefore, any motions which deviate from this due to a perturbation cause a large negative spike in the ELBO value. For simplicity we use ELBO in the form of Eq. 4 meaning that large positive values are a result of a perturbation (see red areas in Fig. 6).

The VAE-planner is able to reject a wide range of impulses applied to the robot's base. However, this operating window is enlarged by increasing the robot's cadence as soon as a disturbance is detected, see Fig. 6. This is a rudimentary response inspired by [14], which reports that humans increase their cadence to recover from both trips and pushes. The cadence is only increased if the ELBO value rises above a pre-selected threshold, below which is considered normal conditions. This value is chosen to be 11.0 and is found by operating the robot for a few minutes and recording the ELBO. The nominal cadence is set to 250 ms and halves to 125 ms for 1.50 s when the threshold is surpassed.

Fig. 6 depicts the ELBO trace for three push events along with the robot's contact schedule. The widths of the white

spaces in the contact schedule half as the cadence increases to mitigate the disturbance. The robot images above this are snapshots taken from the first push and show the robot's recovery. The robot successfully recovers and this usually requires between three and four steps.

V. Conclusion

In this paper, we present a robust and flexible approach for locomotion planning from the perspective of traversing a structured latent-space. This is achieved utilising a deep generative model to capture relevant structure from locomotion data and enables the detection and mitigation of disturbances. The latent space is disentangled to a degree that key salient locomotion features are automatically discovered from a single style of trot gait. An investigation of this latent space reveals a two-dimensional representation which encapsulates the underlying dynamics of the system. This disentanglement is exploited using a drive signal with which dynamically consistent locomotion is generated. Crucially, the amplitude and phase of the drive signal directly control the gait characteristics, namely the cadence, swing height, and full-support duration. Once deployed, the ease with which modulation of the drive signal gives rise to seamless interpolation between gait parameters is demonstrated. Despite generalising to remarkably distinct trot styles compared to the training distribution, the entire range of VAE trajectories remains dynamically consistent. Additionally, utilising a generative model affords the ability to characterise disturbances as out of the distribution seen during training. Though the VAE-planner is able to reject a broad range of impulses applied to the robot's base, this window is broadened by increasing the cadence as soon as the disturbance is detected. This is a rudimentary response inspired by analysis of human locomotion during trip and push recovery [14]. As well as providing a novel alternative to traditional control for quadruped locomotion, the emergence of an oscillatory drive signal governing key gait characteristics has intriguing parallels with the oscillatory signals found in central pattern generators (CPG). These exist in the spinal column and brain stem of vertebrates and regulate motor control and drive rhythmic behaviours such as locomotion in biological systems [15].

Acknowledgements

This work was supported by a UKRI/EPSRC Programme Grant [EP/V000748/1], the EPSRC grant 'Robust Legged Locomotion' [EP/S002383/1], the EPSRC CDT [EP/L015897/1], the UKRI/EPSRC RAIN [EP/R026084/1] and ORCA [EP/R026173/1] Hubs and the EU H2020 Project MEMMO (780684). It was conducted as part of ANYmal Research, a community to advance legged robotics. The authors would like to acknowledge the use of the SCAN facility, and thank Oliver Groth for useful discussions.

References

[1] C. D. Bellicoso, F. Jenelten, C. Gehring, and M. Hutter, "Dynamic locomotion through online nonlinear motion optimization for quadrupedal robots," *IEEE Robot. Automat. Lett. (RA-L)*, vol. 3, no. 3, pp. 2261–2268, 2018.

[2] C. Mastalli, W. Merkt, J. Marti-Saumell, H. Ferrolho *et al.*, "A direct-indirect hybridization approach to control-limited DDP," *arXiv:2010.00411*, 2021.

[3] O. Melon, R. Orsolino, D. Surovik, M. Geisert *et al.*, "Receding-horizon perceptive trajectory optimization for dynamic legged locomotion with learned initialization," in *IEEE Int. Conf. Rob. Autom. (ICRA)*, 2021.

[4] A. W. Winkler, C. D. Bellicoso, M. Hutter, and J. Buchli, "Gait and trajectory optimization for legged systems through phase-based end-effector parameterization," *IEEE Robot. Automat. Lett. (RA-L)*, vol. 3, no. 3, pp. 1560–1567, July 2018.

[5] J. Hwangbo, J. Lee, A. Dosovitskiy, D. Bellicoso *et al.*, "Learning agile and dynamic motor skills for legged robots," *Science Robotics*, vol. 4, no. 26, 2019.

[6] S. Gangapurwala, A. Mitchell, and I. Havoutis, "Guided constrained policy optimization for dynamic quadrupedal robot locomotion," *IEEE Robot. Automat. Lett. (RA-L)*, vol. 5, no. 2, pp. 3642–3649, 2020.

[7] S. Gangapurwala, M. Geisert, R. Orsolino, M. Fallon, and I. Havoutis, "RLOC: Terrain-aware legged locomotion using reinforcement learning and optimal control," *arXiv preprint arXiv:2012.03094*, 2020.

[8] A. W. Winkler, F. Farshidian, D. Pardo, M. Neunert, and J. Buchli, "Fast trajectory optimization for legged robots using vertex-based ZMP constraints," *IEEE Robot. Automat. Lett. (RA-L)*, vol. 2, no. 4, pp. 2201–2208, Oct 2017.

[9] M. Vukobratovic and B. Borovac, "Zero-moment point - thirty five years of its life." *Int. J. Hum. Rob. (IJHR)*, vol. 1, pp. 157–173, 2004.

[10] D. Orin, A. Goswami, and S.-H. Lee, "Centroidal dynamics of a humanoid robot," *Autonomous Robots*, vol. 35, 10 2013.

[11] A. L. Mitchell, M. Engelcke, O. Parker Jones, D. Surovik *et al.*, "First steps: Latent-space control with semantic constraints for quadruped locomotion," in *IEEE/RSJ Int. Conf. Intell. Rob. Sys. (IROS)*, 2020, pp. 5343–5350.

[12] D. Kingma and M. Welling, "Auto-encoding variational bayes," in *Int. Conf. on Learn. Repr. (ICLR)*, 2014.

[13] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," in *Int. Conf. on Mach. Learn. (ICML)*, 2014.

[14] B. E. Moyer, A. J. Chambers, M. S. Redfern, and R. Cham, "Gait parameters as predictors of slip severity in younger and older adults," *Ergonomics*, vol. 49, pp. 329–343, 2006.

[15] I. Steuer and P. A. Guertin, "Central pattern generators in the brainstem and spinal cord: an overview of basic principles, similarities and differences," *Brain Research Reviews*, no. 2, pp. 107–164, 2019.

[16] M. Karl, M. Soelch, J. Bayer, and P. van der Smagt, "Deep variational bayes filters: Unsupervised learning of state space models from raw data," in *Int. Conf. on Learn. Repr. (ICLR)*, 2016.

[17] M. Watter, J. T. Springenberg, J. Boedecker, and M. A. Riedmiller, "Embed to control: A locally linear latent dynamics model for control from raw images," in *NeurIPS*, 2015.

[18] M. Y. Seker, M. Imre, J. H. Piater, and E. Ugur, "Conditional neural movement primitives," in *Rob.: Sci. Sys. (RSS)*, 2019.

[19] A. Srinivas, A. Jabri, P. Abbeel, S. Levine, and C. Finn, "Universal planning networks," in *Int. Conf. on Mach. Learn. (ICML)*, 2018.

[20] D. Hafner, T. Lillicrap, I. Fischer, R. Villegas *et al.*, "Learning latent dynamics for planning from pixels," in *Int. Conf. on Mach. Learn. (ICML)*, vol. 97, 2019, pp. 2555–2565.

[21] H. Ling, F. Zinno, G. Cheng, and M. Panne, "Character controllers using motion VAEs," *ACM Trans. Graph.*, vol. 39, 07 2020.

[22] Y. Yang, T. Zhang, E. Coumans, J. Tan, and B. Boots, "Fast and efficient locomotion via learned gait transitions," in *Conf. on Rob. Learn. (CoRL)*, 2021.

[23] I. Higgins, L. Matthey, A. Pal, C. Burgess *et al.*, "β-VAE: Learning basic visual concepts with a constrained variational framework." in *Int. Conf. on Learn. Repr. (ICLR)*, 2017.

[24] C. Dario Bellicoso, F. Jenelten, P. Fankhauser, C. Gehring *et al.*, "Dynamic locomotion and whole-body control for quadrupedal robots," in *IEEE/RSJ Int. Conf. Intell. Rob. Sys. (IROS)*, 2017, pp. 3359–3365.

[25] J. Hwangbo, J. Lee, and M. Hutter, "Per-contact iteration method for solving contact dynamics," *IEEE Robot. Automat. Lett. (RA-L)*, vol. 3, no. 2, pp. 895–902, 2018.

[26] G. A. Pratt and M. M. Williamson, "Series elastic actuators," in *IEEE/RSJ Int. Conf. Intell. Rob. Sys. (IROS)*, 1995.

[27] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (elus)," in *Int. Conf. on Learn. Repr. (ICLR)*, 2016.

Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor **(only required where there isn't already a statement of contribution within the paper itself).**

Title of Paper	Next Steps: Learning a Disentangled Gait Representation for Versatile Quadruped Locomotion
Publication Status	Published
Publication Details	Alexander L. Mitchell, Wolfgang Merkt, Mathieu Geisert, Siddhant Gangapurwala, Martin Engelcke, Oiwi Parker Jones, Ioannis Havoutis, Ingmar Posner Published at ICRA 2022

Student Confirmation

Student Name:	Alexander Luis Mitchell		
Contribution to the Paper	Led efforts on: • Proposed work and created the team • Created and developed methodologies • Ran experiments • Created all figures Co-led efforts: • Paper writing • Experiment support • Optimising deployment code		
Signature		Date	02/OCT/2023

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title: Professor Ioannis Havoutis		
Supervisor comments These statements are correct.		
Signature Ioannis Havoutis	Date	02/OCT/2023