Automatic Gait Pattern Selection for Legged Robots

Jiayi Wang [1] Iordanis Chatzinikolaidis [1] Carlos Mastalli [1,2] Wouter Wolfslag [1]
Guiyang Xin [1] Steve Tonneau [1] Sethu Vijayakumar [1,2]

***Abstract*— An important issue when synthesizing legged locomotion plans is the combinatorial complexity that arises from gait pattern selection. Though it can be defined manually, the gait pattern plays an important role in the feasibility and optimality of a motion with respect to a task. Replacing human intuition with an automatic and efficient approach for gait pattern selection would allow for more autonomous robots, responsive to task and environment changes. To this end, we propose the idea of building a map from task to gait pattern selection for given environment and performance objective. Indeed, we show that for a 2D half-cheetah model and a quadruped robot, a direct mapping between a given task and an optimal gait pattern can be established. We use supervised learning to capture the structure of this map in a form of gait regions. Furthermore, we propose to construct a warm-starting trajectory for each gait region. We empirically show that these warm-starting trajectories improve the convergence speed of trajectory optimization problems up to 60 times when compared with random initial guesses. Finally, we conduct experimental trials on the ANYmal robot to validate our method.**

I. Introduction

Motion and contact planning for legged locomotion in arbitrary environments is an open problem. In particular, optimizing the gait pattern adds combinatorial complexity to this high-dimensional problem. This complexity can be interpreted as a wide range of possible motions for each gait pattern. However, an appropriate gait pattern selection might be crucial to find a feasible motion [1], [2]. To avoid this challenge, most of the traditional approaches require to prespecify the gait pattern, e.g. [3], [4], [5], [6], [7], [8], [9], [10]. However, biological and computational studies have identified that the choice of gait pattern has large impacts on the performance of legged locomotion [11], [12], [13], [14]. These studies show that gait pattern selection highly depends on the locomotion speed. Furthermore, some of the gait patterns might not be suitable choices in a given terrain condition (e.g. climbing [15]), and manually defining appropriate contact sequence is tedious and difficult.

To enhance autonomous operation of legged robots, our research aims to build a map from task conditions to optimal gait patterns. The main idea is to encode this map such that we can quickly retrieve the optimal gait pattern during online planning. In this exploratory work, we represent robots using single rigid body models. The scalability of the approach towards more complex robot models is discussed in the last section.

[1] Authors are with the School of Informatics, The University of Edinburgh, Edinburgh, United Kingdom.
[2] Authors are with Artificial Intelligence Programme, The Alan Turing Institute, London, United Kingdom.
e-mail: jiayi.wang@ed.ac.uk

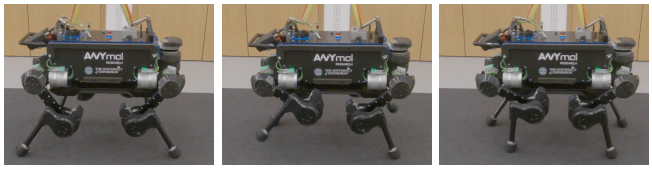


Fig. 1: Snapshots of an experimental trial on the ANYmal robot. In this scenario, for a given performance objective, our proposed framework discovers that the walking-trot gait is preferred for the locomotion task with a forward speed of $0.3^{m}/_{s}$ and cycle duration of $0.8s$ (Video: `https://youtu.be/BHfPymX6Hhs`).

Contributions and method overview

We establish optimal[1] gait pattern maps for a 2D half-cheetah model in a defined task descriptor space given different environment models and performance objectives. The maps typically form several contiguous regions for which a particular gait pattern is optimal. This property allows us to use supervised learning to capture the structure of the map efficiently. Also, we observed that the optimal trajectories within each gait region are qualitatively similar, which implies that they are in the same basin of attraction. We empirically find that the mean of these trajectories can be used to initialise optimization-based planning algorithms. By utilizing the optimal gait pattern maps along with the obtained initial seeds, we achieve significant gains in computation time for our trajectory optimization method. We also present preliminary results on the extension to the 3D quadruped robot ANYmal [16] (Fig. 1).

To generate the map, we propose a nonlinear optimal control formulation based on Mixed-Integer Nonlinear Programming (MINLP). With our formulation, we compute a dataset of optimal gait patterns and state trajectories for a predefined set of tasks and environments *offline* – where the task is described by the locomotion speed and cycle duration, and the environment is defined by its slope angle. Then, we employ a neural network to learn and encode this map from the generated dataset.

The rest of the paper is organised as follows. Section II presents the related work. Section III introduces the proposed method for building the gait pattern map. Our results are detailed in Section IV, while we discuss our findings and the possibility of extending the method to more complex models and tasks in Section V.

[1] Across the paper we use the term optimal to refer to the locally optimal solutions enhanced using random restarts.

II. Related Work

Recent advances in Trajectory Optimization (TO) has enabled us to explore the modulation of gait patterns through a continuous optimization processes. These formulations owe their success to techniques such as: (i) enforcing complementary conditions [17], [18], (ii) smoothing of discontinuous contact events [19], [20], [21], (iii) phase-based parameterization of individual limbs [19], [21], and (iv) resolving system dynamics through contacts in the inner loop of a bilevel optimization framework [22], [23]. Generally speaking, these methods require to smooth the contact dynamics in order to numerically compute a search direction[2] between different contact modes. However, Toussaint et al. [24], [25] argue that the landscape of this optimization problem is highly non-convex, i.e., different contact sequences imply different local minima. It renders the exploration of a wide range of contact sequences impossible, and the optimizer can get stuck in infeasible local minima and fail to generate valid motions. This is the reason why many methods avoid this combinatorial complexity by assuming a given gait and step timings sequence, e.g. [1], [6], [10].

An alternative for gait pattern optimization would be Mixed-Integer Programming (MIP). MIP can automatically explore the entire search space in a systematic manner. This is done by a Branch and Bound (B&B) algorithm [26] that performs a tree search for each integer variable which are used to describe the contact sequences [1], [2], [27]. In comparison to exhaustive search, this method increases search efficiency because the B&B algorithm keeps pruning branches with high cost, restricting searches to promising ones.

To achieve fast computation, Aceituno et al. [2] use (i) a single binary value for representing the entire foot-swing phase, (ii) a convex model of the centroidal dynamics, and (iii) a fixed phase duration. The first assumption cannot capture unsynchronized breaking of contacts, and thus it models only symmetrical gaits (e.g. walking and trotting). The second assumption does not properly encode the typical changes of the angular momentum needed in highly dynamic motions (i.e. bounding). The last assumption restricts the search space of possible trajectories being considered. Our method follows the same paradigm as in [2] but can consider both synchronized and unsynchronized gait patterns. Additionally, the angular momentum is accurately described by a full nonlinear model and the phase duration is considered as decision variable. Due to the combinatorial complexity, MIP are computationally expensive, which prohibits online usage. Instead, this work explores the possibility to extract a gait pattern selection policy by solving the MIP offline. Then machine learning techniques are used to find a direct map between robot specifications, task objectives and environmental models to the optimal gait patterns – with optimal representative trajectories within the extracted gait pattern region used to bootstrap local optimisation.

[2]The search direction is computed locally from the derivatives of the problem.

III. Automatic Gait Pattern Selection

In this section, we first describe the problem of finding the optimal gait pattern map. Next, we introduce a novel MINLP optimal control formulation that enables us to simultaneously compute the optimal motion and gait pattern for a given task and environment. The MINLP is used to discover the gait pattern map for a set of task and environment conditions. Lastly, the extracted optimal gait pattern map is encoded by a neural network that can be used as a gait selection policy in an online fashion.

A. Problem description

We model the robot as a single rigid body along with n_l mass-less legs – while this assumption may sound limiting for hardware execution, this is less likely to affect gait selection decisions. Note that we could still rely on instantaneous or predictive controllers [28], [29], [30] to locally adapt the motions while considering the leg inertia. We define $\boldsymbol{\Gamma} = [\boldsymbol{r}, \boldsymbol{\Theta}]^T$ as the base state, where $\boldsymbol{r} \in \mathbb{R}^3$ is the Center of Mass (CoM) position and $\boldsymbol{\Theta} \in \mathbb{R}^3$ defines the yaw, pitch and roll angles, and $\boldsymbol{p}_l \in \mathbb{R}^3$, $\boldsymbol{f}_l \in \mathbb{R}^3$ represent the foot location and contact forces of leg l expressed in the world frame, respectively. For simplicity, we stack vectors $\boldsymbol{P} = [\boldsymbol{p}_1, \cdots, \boldsymbol{p}_{n_l}]^T$ and $\boldsymbol{F} = [\boldsymbol{f}_1, \cdots, \boldsymbol{f}_{n_l}]^T$ to collect the foot location and contact force of each leg.

Let us consider a scenario specification described by $\boldsymbol{S} = [\boldsymbol{\kappa}_{task}, \Omega, J]^T$, where $\boldsymbol{\kappa}_{task}$ denotes the task descriptors, Ω is a model of the environment, and J is the performance objective. We consider a periodic locomotion task where the task descriptor $\boldsymbol{\kappa}_{task} = [v, T]$ defines the forward speed v of the CoM along the tangential direction of the environment and cycle duration T, while Ω represents slope terrains whose inclination is defined by γ.

Given a scenario specification $\boldsymbol{S}$, we aim to compute a motion plan

$$\boldsymbol{\pi}(t) = [\boldsymbol{\Gamma}(t), \boldsymbol{P}(t), \boldsymbol{F}(t)], \tag{1}$$

for a finite time horizon $t \in [0, T]$.

The swing trajectories $\boldsymbol{P}(t)$ and contact force profiles $\boldsymbol{F}(t)$ are discontinuous at the instant of contact transition and they exhibit an underlying structure determined by the gait pattern $\boldsymbol{\Lambda}$. The motion can be described through binary values Λ_i, used in each phase, along with continuous motions $\boldsymbol{\pi}(t)$ as depicted in Fig. 2. Below we describe how to compute and extract a direct mapping from task descriptor space $\boldsymbol{\kappa}_{task}$ to the optimal gait pattern for a given environment Ω and performance objective J, i.e. $\boldsymbol{\Lambda}^*_{\Omega,J} = \mathbf{g}_{\Omega,J}(\boldsymbol{\kappa}_{task})$.

B. MINLP optimal control formulation

Our formulation computes the optimal gait pattern $\boldsymbol{\Lambda}^*$ by searching for an optimal N-phase motion plan. The gait pattern matrix $\boldsymbol{\Lambda}$ is included as a decision variable, along with the base trajectory $\boldsymbol{\Gamma}(t)$, feet trajectories $\boldsymbol{P}(t)$, contact force profiles $\boldsymbol{F}(t)$, and the *phase-time* vector $\tilde{\boldsymbol{t}}$. Note that $\tilde{t}_i \in \mathbb{R}$ denotes the terminal time of the i-th phase (Fig. 2).

We formulate the hybrid optimal control problem as

$$\min_{\boldsymbol{\pi},\boldsymbol{\Lambda},\tilde{\boldsymbol{t}}} \quad \sum_{i=1}^{N} \int_{\tilde{t}_{i-1}}^{\tilde{t}_i} J(\boldsymbol{\pi},\boldsymbol{\Lambda})\, dt \tag{2a}$$

$$\text{s.t.} \quad \boldsymbol{h}(\boldsymbol{\pi}_0, \boldsymbol{\pi}_T, \boldsymbol{\kappa}_{\text{task}}) \leq 0, \quad \text{(task)} \tag{2b}$$

$$0 = \tilde{t}_0 \leq \tilde{t}_1 \leq \cdots \leq \tilde{t}_N = T, \quad \text{(timings)} \tag{2c}$$

$$\ddot{\boldsymbol{\Gamma}} = \boldsymbol{f}_d(\boldsymbol{\Gamma}, \boldsymbol{P}, \boldsymbol{F}), \quad \text{(dynamics)} \tag{2d}$$

$$\boldsymbol{P} \in \boldsymbol{B}(\boldsymbol{\Gamma}), \quad \text{(kinematics)} \tag{2e}$$

$$\boldsymbol{F} \in \boldsymbol{f}_\mu(\boldsymbol{P}, \Omega), \quad \text{(friction)} \tag{2f}$$

$$\text{For } t \in [\tilde{t}_{i-1}, \tilde{t}_i]\text{:}$$

$$\boldsymbol{h}_{\Lambda_i}(\boldsymbol{F}, \boldsymbol{P}, \Omega) \leq 0, \quad \text{(contacts)} \tag{2g}$$

where, for simplicity, we do not explicitly write the time dependency for the decision variables. To ensure the motion plan is physically consistent, we define different constraints in (2b)-(2g). They describe the desired task, phase timings, system dynamics, friction cone, and contact condition. Specifically, we can instantiate these terms as follows.

1) Task: We impose the periodic task constraints defined by $\boldsymbol{\kappa}_{task}$ as:

$$\boldsymbol{r}_T = \boldsymbol{r}_0 + vT, \tag{3a}$$

$$\boldsymbol{\Theta}_T = \boldsymbol{\Theta}_0, \tag{3b}$$

$$\boldsymbol{P}_T = \boldsymbol{P}_0. \tag{3c}$$

2) Timings: We ensure that the switching times $\tilde{t}_i$ increase monotonically in (2c). Additionally, we impose a final cycle duration defined by T.

3) Dynamics: In (2d), we describe the dynamics using the Newton-Euler equations for a single rigid body,

$$\begin{aligned} m\ddot{\boldsymbol{r}} &= m\boldsymbol{g} + \sum_{l=1}^{N_l} \boldsymbol{f}_l \\ \mathcal{I}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times (\mathcal{I}\boldsymbol{\omega}) &= \sum_{l=1}^{N_l} \boldsymbol{f}_l \times (\boldsymbol{r} - \boldsymbol{p}_l), \end{aligned} \tag{4}$$

where m is the total mass, $\mathcal{I}$ is the inertia tensor of the robot base, $\boldsymbol{\omega}$ is the angular velocity, $\boldsymbol{g}$ is the gravitational acceleration.

4) Kinematics: In (2e), we constrain each leg to stay inside a box with length $2\boldsymbol{b}$ as the single rigid body model does not consider the limbs' configuration. The box origin is placed at the default foot position $\bar{\boldsymbol{p}}_l$, i.e.

$$|R\,(\boldsymbol{p}_l - \boldsymbol{r}) - \bar{\boldsymbol{p}}_l| \leq \boldsymbol{b}, \tag{5}$$

where R is the rotation matrix from the world frame to the base frame.

5) Friction: (2f) approximates the friction cone using a pyramid:

$$-\mu \boldsymbol{f}_l^{\hat{n}} \leq \boldsymbol{f}_l^{\hat{t}_1, \hat{t}_2} \leq \mu \boldsymbol{f}_l^{\hat{n}} \tag{6}$$

where μ is the friction coefficient, $\boldsymbol{f}_l^{\hat{n}}$ is the normal component of the contact force, and $\boldsymbol{f}_l^{\hat{t}_1}$, $\boldsymbol{f}_l^{\hat{t}_2}$ are its tangential components.

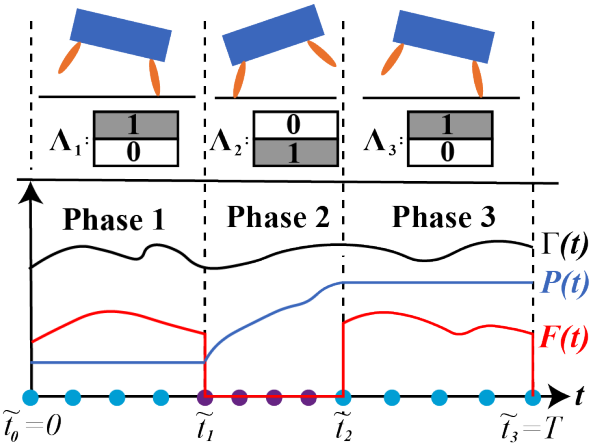


Fig. 2: Continuous and binary decision variables in our optimal control formulation. $\boldsymbol{\Gamma}(t)$, $\boldsymbol{P}(t)$ and $\boldsymbol{F}(t)$ refer to the CoM trajectory, feet trajectories, and contact force profiles, respectively. For illustration purposes, we use scalars to represent these vectors. The motion is described using N phases. Each phase has an associated contact configuration Λ_i and switching time $\tilde{\boldsymbol{t}}_i$. Here we show an example gait pattern where 0 and 1 represents active and inactive contact, respectively. Different gait patterns can be represented by toggling the binary values of Λ_i. The time t is discretized using a variable time-step parameterization [32], where each phase has the same number of integration nodes (blue and purple dots) that scale proportionally with respect to the switching time $\tilde{\boldsymbol{t}}_i$.

6) Contacts: The contact dynamics constraint (2g) is defined as follows:

$$\boldsymbol{f}_{l,n} \geq 0, \quad \text{(unilaterality)} \tag{7a}$$

$$\boldsymbol{\phi}(\boldsymbol{p}_l) \geq 0, \quad \text{(non-penetration)} \tag{7b}$$

$$\Lambda_{i,l} = 0 \Rightarrow \quad \boldsymbol{f}_l = 0, \quad \text{(inactive contact)} \tag{7c}$$

$$\Lambda_{i,l} = 1 \Rightarrow \quad \boldsymbol{\phi}(\boldsymbol{p}_l) = 0,\ \dot{\boldsymbol{p}}_l = 0, \quad \text{(active contact)} \tag{7d}$$

where $\boldsymbol{\phi}(\boldsymbol{p}_l)$ is the signed distance between the foot and the environment, and $\Lambda_{i,l}$ denotes the contact configuration of a leg l at phase i. These set of constraints enforce (i) unilateral forces since the robot can only push against the environment (7a), (ii) the foot cannot penetrate the ground (7b), (iii) the contact force or velocity are zero if the contact state is 0 or 1, respectively, (7c), (7d). Note that the latter constraints describe the complementary condition of contacts through an integer representation, and we use the standard big-M formulation to model them [31].

To solve the presented optimal control problem numerically, we transcribe it into a Mixed-Integer Nonlinear Programming (MINLP) problem by using the variable time-step parameterization presented in [32]. As Fig. 2 illustrates, each phase has same number of uniformly distributed knots whose time interval scales proportionally to the change of switching time $\tilde{\boldsymbol{t}}$. The integrals (i.e. system dynamics and cost function) are approximated by a forward Euler integration scheme.

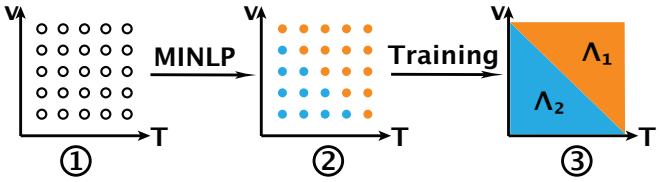


Fig. 3: Procedure to extract the optimal gait pattern map. We firstly uniformly sample the task descriptor space. We then use our MINLP approach to discover optimal gait patterns for these samples. Eventually, we use the pairs of task samples and discovered optimal gait patterns as a training set to classify the regions using a neural network.

C. Discovery and encoding of the gait pattern regions

Given an environment model Ω and a performance objective J, we first pick D tasks $\{\boldsymbol{\kappa}_{task,d} = [v_d, T_d]^T\}$ that are uniformly sampled in the task descriptor space as illustrated in Fig. 3. Then, we compute the optimal gait patterns $\boldsymbol{\Lambda}_d^*$ of these selected task samples $\boldsymbol{\kappa}_{task,d}$ from the MINLP formulation. We can only guarantee local optimality because our problem is nonlinear. Thus, we run the optimization between 15 to 50 times for each sampled task $\boldsymbol{\kappa}_{task,d}$ with randomized initial seeds to increase the chances of finding the global minimum. We use the pairs of task samples and optimal gaits $\{\boldsymbol{\kappa}_{task,d}, \Lambda_d^*\}$ as a training set for a neural network classifier. Our neural network has 3 hidden layers with a total number of 230 neurons. The hyper-parameter of the neural network can be decided by the standard cross-validation process [33].

IV. Results & Discussion

We present next the optimal gait pattern maps for different terrains and performance objectives, while highlighting their properties and how they can be integrated in existing TO frameworks to gain performance improvements. A video is available at `https://youtu.be/BHfPymX6Hhs`, that shows different computed gait patterns and an experimental validation on the ANYmal robot. In all cases, we model the optimization problems using `MATLAB`, while using the interior-point direct algorithm by `KNITRO` as the solver [34]. Furthermore, we compute the gradient and Hessian using the automatic differentiation framework `CasADi` [35]. The dataset for training the neural network is computed by the cluster Eddie provided by the Edinburgh Compute and Data Facility (ECDF). For online evaluations, we use a computer with an Intel Xeon E3-1535M v6 (Maximum 4.20 GHz) and 32 GB memory.

A. Gait Pattern Discovery for a 2D Half-Cheetah Model

First, we consider a 2D half-cheetah model. We compare the results for different terrain inclinations γ and performance objectives J. For all cases, we set the number of phases to 4, with 10 knots per phase. Even though we use a simplified model and a small number of phases, there are a total of $4^4 = 256$ combinations for the optimizer to consider. To avoid undesirably fast phases, we constrain each phase duration to at least 10% of the cycle duration T.

1) Contact forces minimization: We start by considering a flat terrain ($\gamma = 0$) and a performance objective of minimizing contact forces,

$$J_1 = \int_0^T \sum_{l=1}^{N_l} \boldsymbol{f}_l^2 \, dt. \tag{8}$$

This cost can be interpreted as the minimum effort to achieve the desired task.

The optimized gait patterns are summarized in Fig. 4a, while all discovered gait patterns are described in Fig. 5. At very low speeds (up until $0.3^m/_s$ approximately), the model favors a walking gait. If we slightly increase the speed (0.5 to $0.7^m/_s$), the model switches to a gallop-walk gait. If we further increase the speed ($0.9^m/_s$ and onward), galloping is preferred. For large cycle durations (1.2 to $1.6s$) and speeds near the maximum (1.7 to $2.3^m/_s$), bounding is preferred over galloping. Finally, pronking and gallop-bounding gaits cover two small regions only, the former during the transition from gallop-walk to gallop, while the latter during the transition from galloping to bounding.

2) Trunk vibration minimization: We switch the performance objective to

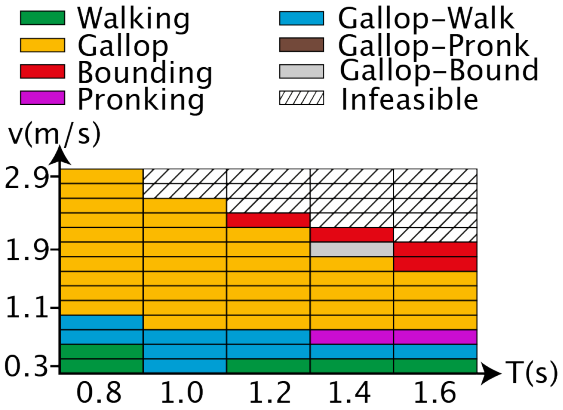
$$J_2 = \int_0^T ((v_t - v)^2 + v_n^2 + (\theta - \gamma)^2 + \dot{\theta}^2) \, dt, \tag{9}$$

where v_t and v_n are the tangential and normal CoM velocities with respect to the terrain, and θ is the pitch angle. This cost encourages to minimize robot base vibration. This is a common requirement during inspection missions, i.e. the camera on the robot needs to remain fixed on a point of interest. Since the terms in (9) can be infinitesimally small (approach 0), we scale all of them by 10^6 times to increase the numerical stability.

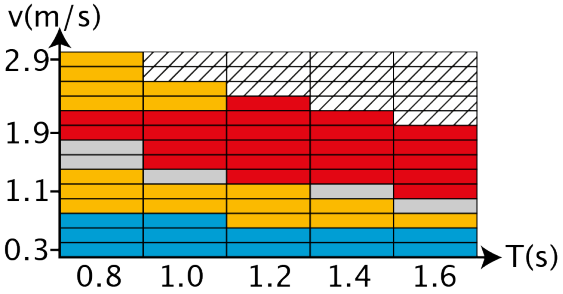
The optimized gait patterns are shown in Fig. 4b. We observe that some regions of optimal gaits remain the same. However, there are some salient changes worth discussing. For medium velocities and across the entire cycle duration range, the bounding gait is preferred over the galloping one. This leads also to an increase of the gallop-bound transient gait at the borders of the two regions. Additionally, the gallop-walk gait becomes prevalent at the low speed regime.

3) Trunk vibration minimization on a 20° slope: We set the terrain slope to 20° and use the performance objective J_2. Fig. 4c summarizes the results for this case. We observe that bounding gaits are replaced by galloping gaits for low cycle duration in the medium velocity regime. Furthermore, pronking gaits become more prominent on the boundary between these two gaits.

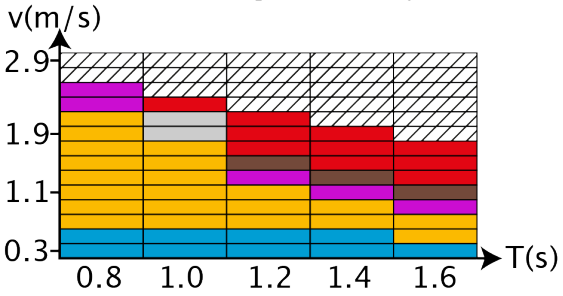
4) Discussion: Depending on the optimized objective and environment characteristic, optimal gait pattern selection can have sufficient differences. As a result, it is unreasonable to expect that a human operator will have the intuition to select proper gait patterns online during motion planning. Our approach provides a precise and systematic way to capture these differences, making automatic gait selection possible without needing to compute gaits every time from scratch – albeit for a particular robot dynamics.



(a) Flat terrain with performance objective J_1.



(b) Flat terrain with performance objective J_2.



(c) 20° sloped terrain with performance objective J_2.

Fig. 4: Optimal gait patterns for a 2D half-cheetah model under different performance objectives and terrain inclinations. The results are sufficiently distinct, demonstrating that heuristic selection can be tedious. Note that most of the regions can be clustered according to the computed gait patterns.

Another important observation is that *optimal gait patterns exhibit structural properties*; for example, distinctive gait patterns tend to occupy similar regions, as shown on the maps in Fig. 4. This observation motivates the use of machine learning techniques to capture the relationships among similar task descriptors, without having to resort to dense indexing. A simple strategy to obtain the optimal gait patterns using a neural network was described in Section III-C.

B. Fast Computation of Locomotion Plans

Based on the last observation, we address the following question: are there substantial differences between state trajectories within similarly clustered task descriptor samples?

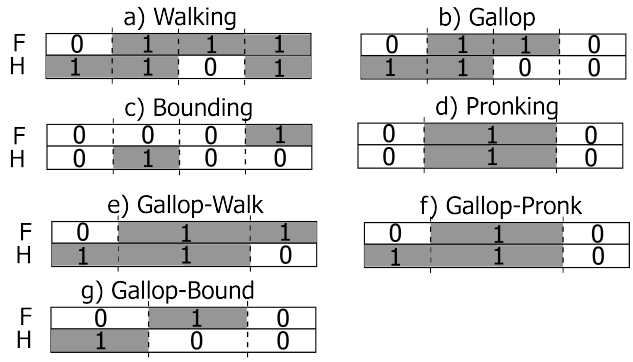
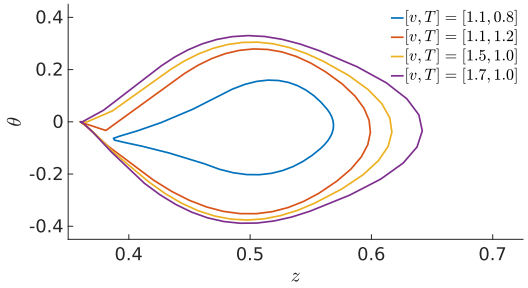
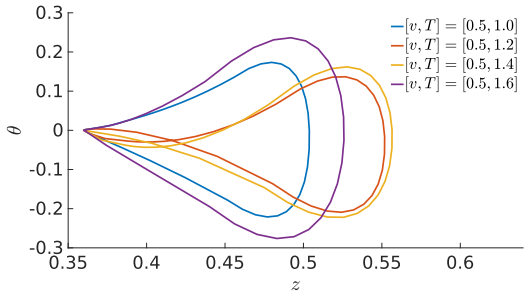


Fig. 5: Discovered gait patterns for the 2D half-cheetah model. F refers to front and H to hind leg, 0 to inactive and 1 to active contacts.



(a) Gallop gait.



(b) Gallop-walk gait.

Fig. 6: Sampled state-space trajectories in a connected region for two distinct optimal gait patterns.

And if not, how can we exploit this property? In our findings we encounter that the trajectories do not exhibit large differences. In Fig. 6, we plot the state-space trajectories for the CoM height z and body orientation θ for two different gait patterns. The resulting trajectories look qualitative similar. It seems that each gait region has its own basin of attraction. Using this, we can build simple strategies to warm-start TO problem with predefined gait patterns. Concretely, we use a neural network to classify the optimal gait pattern. Then, we estimate a nominal initial trajectory, e.g. the mean of the sampled trajectories, based on the gait pattern region that the task descriptor falls in.

Using this strategy, we perform a comparison between two TO frameworks with pre-specified gait patterns. For both methods, we use the neural network to select the appropriate gait pattern given the task descriptor. For the first method,

TABLE I: Differences between two TO methods with pre-specified optimal gait pattern, with and without nominal state, for the 2D half-cheetah model.

Method	Computation time (s)	Successful runs	Consistency
MINLP	93.66±53.7	30	N/A
TO: Gait pattern only	3.05±3.26	25	13
TO: Gait pattern & state	**0.115±0.02**	30	30

we randomly initialize the state variables. For the second method, we identify the region where the gait pattern belongs to, and use the nominal solution as initial seed. We solve the resulting TO problems for 10 random task descriptors for all 3 cases described earlier. Thus, we run each method 30 times. For both MINLP and TO we set 40 knots, which results in 1779 constraints for both methods. There are 742 variables for the MINLP, while TO has 734.

We summarize in Table I the results of this comparison. First, we report the results for the MINLP approach. We use them as a baseline to compute the consistency of the compared methods. Consistency is defined as the number of cases the optimizer converged to the same solution as in the MINLP case. Additionally, we report the successful runs, which correspond to the number of cases that the optimizer was able to output a valid locally optimal solution. There are two reasons why the optimizer might fail: either by exceeding the maximum number of iterations or by failing to output a feasible solution.

By using the nominal state information to warm-start the optimizer, we obtained approximately a 30× computation time decrease with respect to the one using only the optimal gait pattern. Furthermore, using the nominal state provides additional improvements: the number of times the optimizer is successful increases, while being able to discover more consistent local minima with respect to MINLP results.

C. Optimal Gait Pattern Discovery for the 3D Quadrupedal Robot ANYmal

We repeat the previous analysis for the 3D quadruped robot ANYmal. Our aim here is to demonstrate scalability to more complex models, and investigate empirically whether the previous conclusions can be extended to a 3D case. To demonstrate a wide range of motions, we allow large contact forces (max. $500N$) and foot velocity (max. $2.5^{m}/_{s}$).

The problem is described as follows: we select flat terrain and a performance objective that penalizes simultaneously contact forces, trunk vibration, and lateral feet displacements with respect to the nominal position:

$$J_3 = \int_0^T \big(w_1 \boldsymbol{F}^2 + w_2(v_t - v)^2 + w_2 v_n^2 + w_2 \boldsymbol{\Theta}^2 + w_2 \dot{\boldsymbol{\Theta}}^2 + w_2(\boldsymbol{P}_y - \bar{\boldsymbol{P}}_y)^2\big)\, dt, \quad (10)$$

where $\boldsymbol{P}_y$ is the lateral feet displacements, $\bar{\boldsymbol{P}}_y$ is the nominal feet displacements, $w_1 = 10^{-2}$ and $w_2 = 10^6$. Additionally, we define the lower bound on each phase duration as 20% of the cycle duration T.

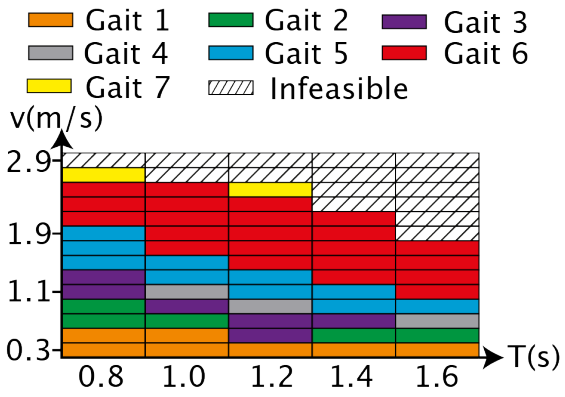


Fig. 7: Optimal gait pattern map for the 3D quadruped.

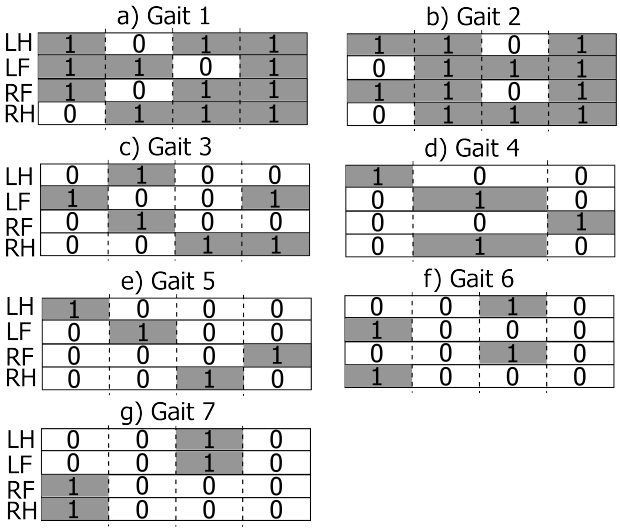


Fig. 8: Discovered gait patterns for the 3D quadruped with four phases.

To model all possible gait pattern combinations, we would need 8 phases. This makes the problem challenging for the MINLP in the current form, since there are $16^8 = 4\,294\,967\,296$ possible combinations. One way to handle this is by introducing convex relaxations as in [1], [2], which would allow us to use mixed integer convex optimization. This would remove the need to solve MINLP with multiple randomized initial seeds; albeit increasing the problem size due to the introduction of auxilary integer variables [1], [2] needed to select convex pieces of nonlinear functions, but saving up time in large dimensions. Alternatively, we select the same number of phases as in the 2D case, which corresponds to $16^4 = 65\,536$ possible combinations. This restricts the number of possible gait patterns but still allows us to examine whether the 2D conclusions also occur for the 3D case.

The result is illustrated in Fig. 7, and some example motions are shown in Fig. 9. We observe a structured gait pattern map here too. For low speeds, the model favors a series of gait patterns that avoid flying phases (gaits 1 to 5 as shown in Fig. 8). In higher speed regimes, the model transits to gait patterns with noticeable flying phases such as

Fig. 9: Snapshots of different computed gaits: three-beat walking, walking trot, four-beat walking and running trot. Red arrows represent contact force vectors.

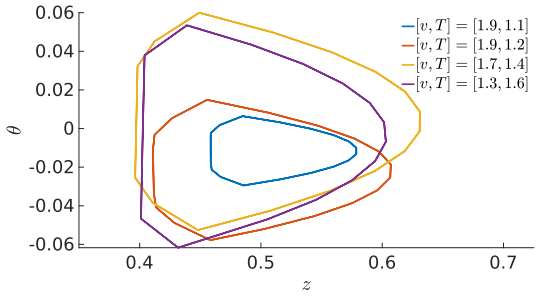


Fig. 10: Sampled state space trajectory for Gait 6 (Running-Trot).

running trot and pace (gaits 6 and 7). Note that the number of gait patterns we obtained is even larger now, demonstrating further the need to automate the gait pattern selection aspect.

Next, we focus on the properties of state trajectories within individual gait pattern regions. We noticed that the shape remains the same and only the scale of the solutions change, as in the 2D case. For example, state space trajectories of the running-trot gait are shown in Fig. 10, for the same state variables shown in Fig. 6. To test the effectiveness of the warm-staring strategy in 3D, we randomly pick 10 task descriptors and compare them with the same two TO methods: one with pre-specified gait pattern only and one with both pattern and nominal state initialization.

We report the results of the comparison in Table II. The optimization problems now have 1948 decision variables and 5008 constraints, and become significantly more challenging to solve. Still, pre-specifying gait pattern and initializing with nominal state trajectories lead to a large computational improvements, without sacrificing the number of successful runs and the consistency.

TABLE II: Differences between two TO methods for computing 3D quadruped locomotion plans with optimal gait pattern initialization, with and without nominal state.

Method	Computation time	Successful runs	Consistency
MINLP	5+ hours	10	N/A
TO: Gait pattern only	19.136±10.23	7	5
TO: Gait pattern & state	**0.3159±0.05**	10	10

Further, to validate our method, we adapt the limits of contact force and foot velocity to be close to the capabilities of the ANYmal robot, and we successfully demonstrate a real-world trial (Fig. 1) of a motion computed by our MINLP approach. We use the controller developed in [28] to track the motion.

V. Conclusion & Future Work

We presented a framework that automates gait pattern selection for legged robots. It enables us to systematically select gait patterns according to task and environment specifications, without requiring heuristics or input from a human operator. Another important aspect is that it can be directly interfaced with existing TO methods that require gait initialization, providing them with an informed initial seed regarding both the gait pattern and the state trajectory.

Future extensions of our work will improve two aspects. First, we will focus on addressing the scalability on high-dimensional models (i.e. multi-rigid body system), large number of phases, and complex environments. These problems stem from either the large number of variables that increase the complexity or from the non-convexity of the

dynamics model. The number of variables can be reduced by improving the accuracy of our integration scheme, e.g. by using a collocation method as in [36]. The non-convexity of the model can be addressed by using convex relaxations of the model's dynamics, for example as in [1].

Second, a more systematic study is required regarding the structural pattern of the optimal gait pattern map, as well as the trajectory properties within the regions of the map. In this work, we exploited the structure of the map and similarity of state trajectories to warm-start TO methods. We believe these findings can be generalized for different tasks (i.e. acceleration and deceleration), performance objectives, and environments. The computational improvements that we enjoy using these properties can allow us to achieve online motion re-planning in real-world scenarios; the human operator will specify speed, while the gait pattern and state initialization will be automatically selected to bootstrap the computation.

ACKNOWLEDGEMENTS

This research is supported by EU H2020 project Memory of Motion (MEMMO, project ID: 780684) and EPSRC UK RAI Hub for Offshore Robotics for Certification of Assets (ORCA, grant reference EP/R026173/1).

REFERENCES

[1] B. Ponton *et al.*, "A convex model of humanoid momentum dynamics for multi-contact motion generation," in *Proc. IEEE Int. Conf. Humanoid Robots*, 2016, pp. 842–849.

[2] B. Aceituno-Cabezas *et al.*, "Simultaneous contact, gait, and motion planning for robust multilegged locomotion via mixed-integer convex optimization," *IEEE Robot. Autom. Lett.*, vol. 3, no. 3, pp. 2531–2538, 2018.

[3] M. Kalakrishnan *et al.*, "Fast, robust quadruped locomotion over challenging terrain," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2010, pp. 2665–2670.

[4] A. W. Winkler *et al.*, "Planning and execution of dynamic whole-body locomotion for a hydraulic quadruped on challenging terrain," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2015, pp. 5148–5154.

[5] C. Mastalli *et al.*, "Trajectory and foothold optimization using low-dimensional models for rough terrain locomotion," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2017, pp. 1096–1103.

[6] J. Carpentier and N. Mansard, "Multicontact locomotion of legged robots," *IEEE Trans. Robot.*, vol. 34, no. 6, 2018.

[7] S. Tonneau *et al.*, "An efficient acyclic contact planner for multiped robots," *IEEE Trans. Robot.*, vol. 34, no. 3, pp. 586–601, 2018.

[8] C. D. Bellicoso *et al.*, "Dynamic locomotion through online nonlinear motion optimization for quadrupedal robots," *IEEE Robot. Autom. Lett.*, vol. 3, no. 3, pp. 2261–2268, 2018.

[9] H.-W. Park *et al.*, "Online planning for autonomous running jumps over obstacles in high-speed quadrupeds," in *Proceedings of Robotics: Science and Systems*, Jul. 2015.

[10] C. Mastalli *et al.*, "Crocoddyl: An Efficient and Versatile Framework for Multi-Contact Optimal Control," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2020.

[11] D. A. Winter, *Biomechanics and motor control of human movement*. John Wiley & Sons, 2009.

[12] M. Srinivasan and A. Ruina, "Computer optimization of a minimal biped model discovers walking and running," *Nature*, vol. 439, no. 7072, pp. 72–75, 2006.

[13] C. D. Remy, "Optimal exploitation of natural dynamics in legged locomotion," Ph.D. dissertation, ETH Zurich, 2011.

[14] W. Xi *et al.*, "Selecting gaits for economical locomotion of legged robots," *Int. J. Robotics Res.*, vol. 35, no. 9, pp. 1140–1154, 2016.

[15] K. Naderi *et al.*, "Discovering and synthesizing humanoid climbing movements," *ACM Trans. Graph.*, vol. 36, no. 4, 2017.

[16] M. Hutter *et al.*, "Anymal-a highly mobile and dynamic quadrupedal robot," in *Proc. IEEE Int. Conf. Intell. Robots Syst.* IEEE, 2016, pp. 38–44.

[17] M. Posa *et al.*, "A direct method for trajectory optimization of rigid bodies through contact," *Int. J. Robotics Res.*, vol. 33, no. 1, pp. 69–81, 2014.

[18] C. Mastalli *et al.*, "Hierarchical planning of dynamic movements without scheduled contact sequences," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2016.

[19] I. Mordatch *et al.*, "Discovery of complex behaviors through contact-invariant optimization," *ACM Trans. Graph.*, vol. 31, no. 4, 2012.

[20] M. Neunert *et al.*, "Trajectory optimization through contacts and automatic gait discovery for quadrupeds," *IEEE Robot. Autom. Lett.*, vol. 2, no. 3, pp. 1502–1509, 2017.

[21] A. W. Winkler *et al.*, "Gait and trajectory optimization for legged systems through phase-based end-effector parameterization," *IEEE Robot. Autom. Lett.*, vol. 3, no. 3, pp. 1560–1567, 2018.

[22] Y. Tassa *et al.*, "Synthesis and stabilization of complex behaviors through online trajectory optimization," in *Proc. IEEE Int. Conf. Intell. Robots Syst.*, 2012.

[23] J. Carius *et al.*, "Trajectory optimization with implicit hard contacts," *IEEE Robot. Autom. Lett.*, vol. 3, no. 4, pp. 3316–3323, 2018.

[24] M. Toussaint *et al.*, "Differentiable physics and stable modes for tool-use and manipulation planning," in *Robotics: Science and Systems*, Jun. 2018, pp. 1–9.

[25] ——, "Differentiable physics and stable modes for tool-use and manipulation planning – extended abstract," 2019, sister Conference Best Paper Track – Extended abstract of the R:SS'18 paper.

[26] J. Clausen, "Branch and bound algorithms-principles and examples," *Department of Computer Science, University of Copenhagen*, 1999.

[27] R. Deits and R. Tedrake, "Footstep planning on uneven terrain with mixed-integer convex optimization," in *Proc. IEEE Int. Conf. Humanoid Robots*, 2015, pp. 279–286.

[28] G. Xin *et al.*, "An optimization-based locomotion controller for quadruped robots leveraging cartesian impedance control," *Frontiers in Robotics and AI*, vol. 7, p. 48, 2020.

[29] C. Mastalli *et al.*, "Motion Planning for Quadrupedal Locomotion: Coupled Planning, Terrain Mapping and Whole-Body Control," *IEEE Trans. Robot.*, 2020.

[30] M. Neunert *et al.*, "Whole-Body Nonlinear Model Predictive Control Through Contacts for Quadrupeds," *IEEE Robot. Autom. Lett.*, vol. 3, pp. 1458–1465, 2018.

[31] L. A. Wolsey and G. L. Nemhauser, *Integer and combinatorial optimization*. John Wiley & Sons, 2014.

[32] M. Soler *et al.*, "Multiphase mixed-integer optimal control framework for aircraft conflict avoidance," in *IEEE Conference on Decision and Control (CDC)*, 2012, pp. 1740–1745. [Online]. Available: https://doi.org/10.1109/CDC.2012.6426709

[33] C. M. Bishop, *Pattern recognition and machine learning*. springer, 2006.

[34] R. H. Byrd *et al.*, "KNITRO: An integrated package for nonlinear optimization," in *Large-Scale Nonlinear Optimization*, G. Di Pillo and M. Roma, Eds. Springer US, 2006, pp. 35–59.

[35] J. A. E. Andersson *et al.*, "CasADi – a software framework for nonlinear optimization and optimal control," *Math. Program. Comput.*, vol. 11, no. 1, Mar. 2018.

[36] F. A. Koolen, "Balance control and locomotion planning for humanoid robots using nonlinear centroidal models," Ph.D. dissertation, MIT, 2019.