# Receding-Horizon Perceptive Trajectory Optimization for Dynamic Legged Locomotion with Learned Initialization

Oliwier Melon, Romeo Orsolino, David Surovik, Mathieu Geisert,
Ioannis Havoutis and Maurice Fallon

***Abstract*— To dynamically traverse challenging terrain, legged robots need to continually perceive and reason about upcoming features, adjust the locations and timings of future footfalls and leverage momentum strategically. We present a pipeline that enables flexibly-parametrized trajectories for perceptive and dynamic quadruped locomotion to be optimized in an online, receding-horizon manner. The initial guess passed to the optimizer affects the computation needed to achieve convergence and the quality of the solution. We consider two methods for generating good guesses. The first is a *heuristic* initializer which provides a simple guess and requires significant optimization but is nonetheless suitable for adaptation to upcoming terrain. We demonstrate experiments using the ANYmal C quadruped, with fully onboard sensing and computation, to cross obstacles at moderate speeds using this technique. Our second approach uses *latent-mode trajectory regression* (LMTR) to imitate expert data—while avoiding invalid interpolations between distinct behaviors—such that minimal optimization is needed. This enables high-speed motions that make more expansive use of the robot's capabilities. We demonstrate it on flat ground with the real robot and provide numerical trials that progress toward deployment on terrain. These results illustrate a paradigm for advancing beyond short-horizon dynamic reactions, toward the type of intuitive and adaptive locomotion planning exhibited by animals and humans.**

## I. Introduction

State-of-the-art legged-locomotion controllers can maintain dynamic motion while handling perturbations from external forces on moderately uneven terrain. Alternatively, locomotion planners can reason over longer horizons to strategically plan footholds and paths across more prominent obstacles, such as those seen in industrial settings. However, the benefits of these two paradigms—speed and foresight, respectively—are not trivial to combine. As a result legged robots have limited performance in scenarios that demand careful foothold selection and momentum shaping.

Meanwhile, legged animals can dynamically traverse obstacles with relative ease. They are able to make rapid inferences about their circumstances and subconsciously relate them to relevant prior experiences to leverage the properties of their bodies in a coordinated manner. Biological locomotion serves as inspiration to motion-generation research in robotics.

This work was supported by the UKRI/EPSRC RAIN Hub [EP/R026084/1] and the EU H2020 Projects MEMMO and THING, the EPSRC grant 'Robust Legged Locomotion' [EP/S002383/1] and a Royal Society University Research Fellowship (Fallon). This work was conducted as part of ANYmal Research, a community to advance legged robotics. The authors are with Oxford Robotics Institute, University of Oxford, UK. Email: {omelon, rorsolino, dsurovik, mathieu, ioannis, mfallon}@robots.ox.ac.uk.

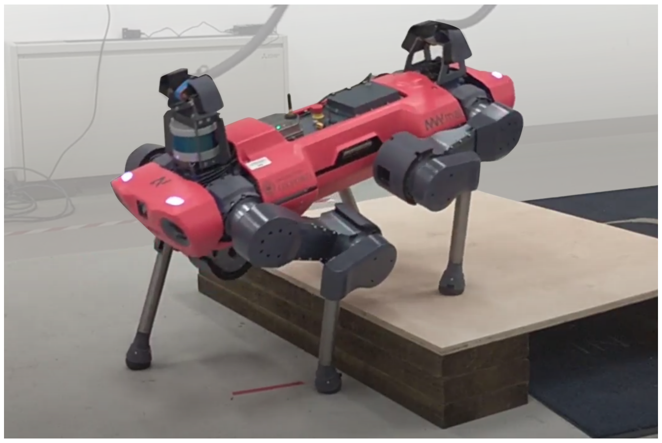


Fig. 1. The ANYmal C quadruped using online, receding-horizon trajectory optimization and real-time perception to cross a $0.20\,\mathrm{m}$ step.

### A. Planning Dynamic Motion Online

Dynamically complex robot motion plans are commonly generated through trajectory optimization—a process which minimizes an objective function while satisfying equality and inequality constraints. The Trajectory Optimization for Walking Robots (TOWR) framework [1] introduced a solution parametrization appropriate for handling the discontinuities inherent to legged locomotion, which ordinarily creates problems for gradient descent, without imposing overly rigid assumptions. Adapting this approach for online use carries several additional challenges, such as the need to minimize the required computation time and the need to employ the optimizer in a *receding-horizon* manner without compromising the solution's expressiveness.

One increasingly popular strategy for online planning is to produce high-quality initial guesses by leveraging offline experience, so that only minor computation is needed to adapt to a scenario seen at runtime. Many earlier examples of this approach focused on manipulators [2], leading to later studies on more dynamic systems [3], [4], [5]. Recent works have considered more complex variants that may be needed when the relevant solution space is multimodal [6], [7]. For the most part, these works impose strong assumptions about the beginning and end of the motion plan being produced. For generalized, sustained behavior by dynamic quadrupeds, a formulation is needed that can handle their multimodality—and to do so without being required to routinely revisit a small and restrictive class of checkpoint states.

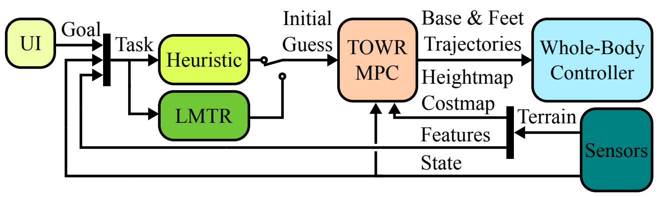


Fig. 2. Overview of the proposed onboard system. The user, or a high-level planning algorithm, specifies the goal as a SE(2) base pose within a couple meters of the robot. The sensors provide the estimated state, along with environment features. Together, these form a task that is input to one of the proposed initializers, Heuristic or LMTR, which produces an asynchronously-parametrized trajectory segment for the trajectory optimizer (TOWR MPC) to refine so that it satisfies dynamic and kinematic constraints. It uses the heightmap, a dense 2.5D elevation map produced using depth cameras or LIDAR, and the costmap to plan favorable footholds. The resulting base and feet trajectories are tracked by the whole-body controller.

### B. Contributions and Outline

Motivated by current literature discussed in Sec. II, we aim to enable *online replanning* of dynamic motions that can be *sustained indefinitely* while anticipating upcoming terrain features by employing *diverse modes of behavior*. This paper presents a substantial extension and integration of our previous works [5], [7]; its novel contributions are:

- Implementation of an *asynchronous*, *segmentation*-based trajectory optimization formulation for legged robots based on the single rigid body dynamic (SRBD) model to enable *receding-horizon* planning (Sec. III-A).
- Incorporation of two alternate methods for initializing the nonlinear programming problem (NLP): a heuristic that makes efficient re-use of the previous plan (Sec. III-B), and the data-driven approach of latent-mode trajectory regression (LMTR) [7] that can generalize the gait in anticipation of upcoming terrain (Sec. III-C).
- Evaluation of the proposed system in simulation and on hardware using a real ANYmal C quadruped (Sec. IV).

Figure 2 provides an overview of our replanning framework, which is detailed throughout Sec. III. Additional features include future state prediction based tracking error, state update within the solver at each solve iteration, a contact regain behavior and use of real-time perception of terrain to select favorable footholds (Sec. III-D–III-F). Following discussion of the findings in Sec. IV, we summarize our conclusions and identify future directions in Sec. V.

## II. Related Work

### A. Motion Planning via Model Predictive Control

Many of the most successful approaches to dynamic legged locomotion use trajectory optimization. They may include the kinematics and dynamics of the limbs [8], [9], [10] or use simplified models such as centroidal dynamics [11] or single rigid body dynamics (SRBD) [1]; they can also adopt different strategies for contact scheduling such as completely predefining the timings, only predefining the sequence [12] or fully discovering both [13], [14]. However, few of these methods are suitable for online motion generation.

To be applicable online, approaches usually rely on simplified dynamics criteria like the Zero-Moment-Point [15], modeling as a single rigid body with fixed orientation [16], [17]. Some works decouple the perceptive foothold selection to reduce computation time [18], [19], [20], [21]. While simultaneous optimization of footholds has been achieved [22], [23], [24], doing it on terrain remains a challenge.

More sophisticated nonlinear MPC approaches use the whole-body model of the legged robot but have optimization horizons that are too short to be able to adequately traverse large obstacles [25]. Other implementations based on the full robot model employ differential dynamic programming [10] with an efficient feasibility-driven formulation that promises to enable high-frequency receding-horizon control. Another family of approaches is mixed-integer programming (MIP) [26], [27] which, however, does not scale well with a large number of decision variables. Other authors successfully used the SRBD which represents a compromise between model accuracy and complexity that allows online replanning [28]. While the SRBD assumes the inertia to be invariant, the changing property can be accounted and compensated for without increasing model complexity [20].

### B. Data-Driven Locomotion

Model-based approaches to motion generation may be limited by optimization time or the discovery effort needed to construct complex or subtle behavior online. This has lead to data-driven approaches that *amortize* this cost and effort by conducting it offline. Bledt *et al.* observed correlations between tasks and optimized behaviors to identify heuristic costs for regularizing online optimization [17], [29]. Alternately, machine learning approaches reduce the role of humans in curating the amortized knowledge.

A hierarchy of two reinforcement learning (RL) policies was used in [30]—one for planning footholds and phase durations, and one for base and end-effector motion. To gradually build up the complexity of the learned behaviors, Xie *et al.* [31] used curriculum learning for bipedal stepping-stone motions, evaluated in simulation. However, these methods become sample-inefficient when dealing with a large state-space and typically require considerable reward shaping.

For producing longer-horizon actions, the work of Mansard *et al.* [3] “warm-started” nonlinear MPC using a model learned from a sampling-based approach. Similar use of a function approximator has enabled a real quadruped to quickly plan several-steps-long dynamic trajectories which it successfully executed to ascend terrain [5]. Guided Trajectory Learning uses consensus optimization to learn a function approximator that only reconstructs feasible motions [32]. In [4], a “memory of motion” of dynamic, collision-free locomotion has been generated using the *HPP Loco3D* planner [33] for the *Crocoddyl* solver [10].

To adequately capture highly varied behavior, multiple regressors can be used, as in computer animation of dogs and humans [34], [35]. For legged robots, [36] used imitation learning of optimal control demonstrations, improving sample efficiency over RL. To remove the need for multiple networks and discrete categories of actions, latent variables can be used to express continuous modal variation [6], [7].

## III. Framework

In this section we present our approach to online replanning and execution of dynamic locomotion. The developed framework is shown in Fig. 2. We consider two alternate initialization methods. The Heuristic is the traditional MPC-style initialization scheme which reuses a part of the previous solution. The remaining section of the new initial guess, for which no prior data is available, is set up using linear interpolation. By contrast, LMTR leverages offline experience to adapt to the current combination of state, goal, and terrain, *i.e.*, the task. The Heuristic imposes more conservative behavior but aptly demonstrates the receding-horizon formulation. Meanwhile, more aggressive and adaptive motions can be expressed via LMTR, although its deployability depends upon the quality of its training data.

### A. Optimization Formulation

Our formulation is based on Hermite-Simpson direct collocation which represents the state using cubic polynomials of Hermite form, *i.e.*, the splines are defined by the values $\mathcal{N}$ and derivatives $\dot{\mathcal{N}}$ of the nodes. We use phase-based end-effector parametrization [1], which enables the base and end-effector nodes to be asynchronous, allowing the phase durations to be optimized to achieve an aperiodic contact sequence. To enable MPC and facilitate its initialization, we keep the dimensionality of the problem constant by proposing a *segmentation*-based formulation with *asynchronous* base and feet time horizons as shown in Fig. 3. It is also vital for the training of the LMTR (Sec. III-C); the regressor has a fixed-size output of a given set of optimization variables $\boldsymbol{\tau}$ representing a trajectory with a predefined time horizon.

The size of the proposed problem is:

$$\begin{aligned}
\#\boldsymbol{\tau} = {} & (3+3+3+3) \times n^{\text{base}}_{\text{nodes}} && \text{base-motion} \\
& + n_{\text{feet}} \times (n_{\text{st}} - 1 + \ n_{\text{sw}}) && \text{durations} \\
& + n_{\text{feet}} \times [3 \times n_{\text{st}} + (3+2) \times n_{\text{sw}}] && \text{foot-motion} \\
& + n_{\text{feet}} \times (3+3) \times n^{\text{force/foot/st}}_{\text{nodes}} \times n_{\text{st}} && \text{forces}
\end{aligned} \quad (1)$$

where the individual components are:

*1) Base-motion:* Base linear position $\mathbf{r}(t) \in \mathbb{R}^3$, linear velocity $\dot{\mathbf{r}}(t) \in \mathbb{R}^3$, angular position $\boldsymbol{\theta}(t) \in \mathbb{R}^3$ and angular velocity $\boldsymbol{\omega}(t) \in \mathbb{R}^3$ values are multiplied by the number of base nodes $n^{\text{base}}_{\text{nodes}}$ which is calculated using the durations of the horizon $\Delta T_{\text{H}}$ and the time step of the base polynomials $dt_{\text{base-poly}}$ as $n_{\text{base-nodes}} = 1 + \text{round}(\Delta T_{\text{H}}/dt_{\text{base-poly}})$.

*2) Durations:* For a quadruped, the number of legs $n_{\text{feet}}$ is 4. The chosen number of stance $n_{\text{st}}$ and swing $n_{\text{sw}}$ phases for each foot $i$ is 2 and 1, respectively.

*3) Foot-motion:* The stance phase is defined by the linear position $\mathbf{p}_i(t) \in \mathbb{R}^3$ values, while the swing phase has both linear position $\mathbf{p}_i(t) \in \mathbb{R}^3$ and linear velocity $\dot{\mathbf{p}}_i(t) \in \mathbb{R}^2$ values. The velocity in the z-dimension is an implicit constraint set to 0.

*4) Forces:* The force values and derivatives $\mathbf{f}_i(t), \dot{\mathbf{f}}_i(t) \in \mathbb{R}^3$ are multiplied by the number of force nodes per foot per stance $n^{\text{force/foot/st}}_{\text{nodes}}$, which is 3 as forces are constrained to $\mathbf{0}$ at the start and end of each *swing* phase. For every mid-motion stance phase, this number further decreases to 2 as the forces at the start and end are $\mathbf{0}$.

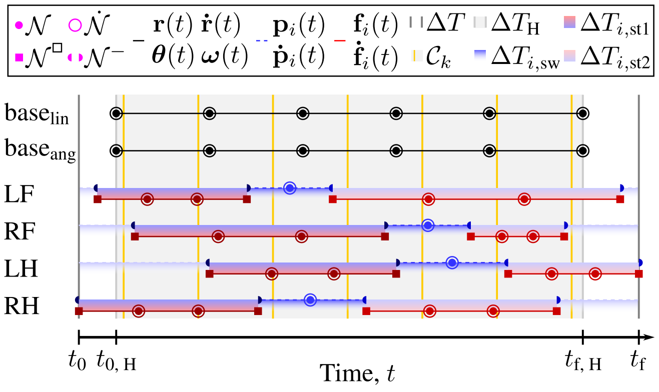


Fig. 3. Our direct collocation, *segmentation*-based problem formulation uses phase-based end-effector (LF, RF, LH, RH) parametrization [1]. To facilitate initialization of the solver and allow the contact phase durations to be optimized, we keep the dimensionality constant by defining base and feet segments *asynchronously*. The components of the figure are thoroughly described in Sec. III-A.

The problem is defined over a duration $\Delta T$ between time $t_0$ of the earliest beginning of the *first* set of stances $\Delta T_{i,\text{st1}}$ and the time $t_{\text{f}}$ of latest ending of the *second* set of stances $\Delta T_{i,\text{st2}}$; in between, the swing phases $\Delta T_{i,\text{sw}}$ occur. If a leg is in swing at $t_{0,H}$, that swing is not re-optimized. Instead, the next stance-swing-stance sequence is considered in the optimization.

Figure 3 illustrates the problem formulation for the base horizon $\Delta T_{\text{H}} = 5\,dt_{\text{base-poly}}$ which is defined with respect to the first and last base nodes occuring at $t_{0,\text{ H}}$ and $t_{\text{f, H}}$, respectively. The figure shows all the constructing nodes, implicit constraints $\mathcal{N}^{\square}$, polynomials, and durations. In addition, a constraint $\mathcal{C}_k$, discretized at a fixed time step $k$, is shown; such a type of constraint can be used to enforce the dynamics equality or the range-of-motion inequality constraints. Notably, these can be enforced not only at the nodes but anywhere along the polynomials. The optimization values that solely represent values for a period of time $t$ are denoted by $\mathcal{N}^{-}$.

In this work, we use the SRBD model and employ geometric shapes known as *superquadrics*, shown in Fig. 8, as range-of-motion constraints described by:

$$|p_{i,x}(t)/A|^a + |p_{i,y}(t)/B|^b + |p_{i,z}(t)/C|^c = 1 \quad (2)$$

whose curvature is defined by the exponents $a, b, c$ and dimensions are specified by the scalings $A, B, C$. This shape conveys a more natural range of foothold choices and avoids unwanted behaviors that a box shape can promote, *i.e.*, yawing to maximize the size of a stride.

### B. Heuristic Initialization

The Heuristic initializes the base trajectory variables using the corresponding values of the solution obtained at the previous replanning cycle. The non-overlapping part of the new base trajectory is linearly interpolated up to the user-defined goal. The footholds corresponding to the first stance phase coincide with the location given by the previous plan whilst the footholds of the second stance phase are located at a nominal distance from target goal.

Unlike LMTR, the Heuristic initializes the NLP with periodic gaits which result in the two stance phases having the same duration ($\Delta T_{i,\text{st1}} = \Delta T_{i,\text{st2}} = \Delta T_{i,\text{st}}$) and in all the feet having the same stance and swing durations $\Delta T_{\text{st}}$ and $\Delta T_{\text{sw}}$ (these values can, however, still be varied during the optimization). The durations can be set using the duty factor $d$ and the cycle factor $c$ such that $\Delta T_{\text{st}} + \Delta T_{\text{sw}} = (1.0 - c)\Delta T_H$ and $d = \Delta T_{\text{st}}/T_{\text{c}}$ (where $T_{\text{c}}$ is the gait cycle time $T_{\text{c}} = \Delta T_{\text{st}} + \Delta T_{\text{sw}}$). The relative time offset $t_{0,i}$ between the start of the base trajectory and that of the $i^{\text{th}}$ foot can be set as the fraction $\alpha_i \in (0, 1)$ such that: $t_{0,i} = \alpha_i \Delta T_H$.

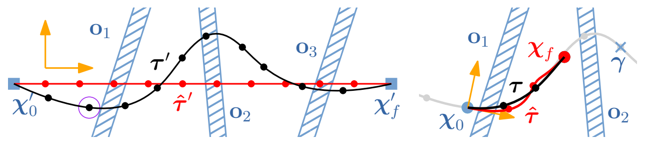
### C. Learned Initialization via LMTR

Solving a complex, long-horizon NLP *from scratch* can require many seconds of computation per second of generated motion. Using shorter horizons reduces computation but makes performance increasingly limited by how intelligently the final boundary conditions can be defined. We approach these issues via imitation learning of a set long-horizon expert trajectories that have been processed into segments, as originally detailed in [7]. This produces a regression model (LMTR) that expresses a diverse continuum of behaviors suited for handling a range of obstacle configurations.

*1) Change of Scope:* Key to the imitation learning setup is the use of segmentation (Sec. III-A) to convert expert data from *stationary-horizon* (SH) to *receding-horizon* (RH) scope, as sketched in Fig. 4, with SH variables using $\cdot'$ notation. SH involves static boundary states $\boldsymbol{\chi}'$ and long trajectories $\boldsymbol{\tau}'$, while in RH $\boldsymbol{\chi}$ are dynamic and $\boldsymbol{\tau}$ are shorter segments. Initialized values are denoted $\hat{\cdot}$, and $\mathbf{o}_i$ are discrete terrain elements. In RH scope, the task $\mathbf{x} = (\boldsymbol{\chi}_0, \boldsymbol{\gamma}, \mathbf{o}_i, \mathbf{o}_{i+1})$ includes $\mathbf{o}_i$ within a local window and an SE(2) goal $\boldsymbol{\gamma}$ taken from a base pose on the SH trajectory. With $\boldsymbol{\gamma}$ occurring further along than $\boldsymbol{\chi}_f$, the size of $\boldsymbol{\tau}$ can be controlled separately from the amount of foresight encompassed by $\mathbf{x}$.

*2) Expert Dataset:* Training data are generated in bulk as in Fig. 5. Within SH scope, tasks $\mathbf{x}'$ are sampled from a distribution of static robot states and terrains and $\hat{\boldsymbol{\tau}}'$ are initialized naively with linear interpolation and standard gait timings. Allowed a high number $N_{max}$ of iterations per sample, TOWR with analytical costs [5] returns expert trajectories $\boldsymbol{\tau}'$. The segmentation process of Sec. III-A is then used to generate the RH dataset, *i.e.*, pairs of RH tasks and segments $\{\mathbf{x}, \boldsymbol{\tau}\}$, desired for imitation learning.

*3) Generalization via LMTR:* Due to the strong nonconvexity of the NLP, similar $\mathbf{x}$ may be paired with substantially different $\boldsymbol{\tau}$, reflecting a rich inventory of behaviors. This prevents directly fitting a "unimodal" regressor $f : \mathbf{x} \longrightarrow \boldsymbol{\tau}$, which must smoothly vary based on $\mathbf{x}$ alone. Instead, a "mode" $\mathbf{z}$ that represents task-independent variation can be learned in an unsupervised fashion by training a Conditional Variational Autoencoder, where $\boldsymbol{\tau}$ is the reconstructed variable, $\mathbf{x}$ is the condition, and $\mathbf{z}$ is the latent variable. The decoder module of this model then serves as the regressor



(a) Stationary Horizon (SH) (b) Receding Horizon (RH)

Fig. 4. Illustration of two scopes with which to view trajectories that cross discrete obstacles (rectangles). Blue indicates components of the "task", *i.e.*, inputs. Red and black denote outputs of the initializer and optimizer, respectively. (a) SH scope, with static robot states at the boundaries of a long trajectory, and naive initialization. (b) RH scope, with a trajectory segment bounded by dynamic states, and learning-based initialization (LMTR).

$f : (\mathbf{z}; \mathbf{x}) \longrightarrow \boldsymbol{\tau}$ deployed online. Selection of $\mathbf{z}$ is done using nearest-neighbor lookup on expert pairs $\{\mathbf{x}, \mathbf{z}\}$ with a weighted distance metric.

### D. State Prediction due to Optimization Delay

Most MPC algorithms target relatively high frequencies of 40 Hz to 200 Hz [17], [29]. Because our framework is concerned with solving longer-horizon dynamic motion, it operates at a much lower, fixed frequency of a few Hz. As a result, it experiences delays measured in hundreds of milliseconds between measurements of the state and execution of resulting solutions. To manage this issue, the starting point of the computed trajectory corresponds to the time at which it will be executed and occurs after the computation begins. We implement a simple *predictor* to handle this by using the desired position computed by the previous plan and the current tracking error:

$$\hat{\boldsymbol{\chi}}(t_{0,k+1}) = \boldsymbol{\chi}^*_k(t_{0,k+1}) + \boldsymbol{\alpha}(t) \odot \overline{\boldsymbol{\chi}}(t) \tag{3}$$

$$\overline{\boldsymbol{\chi}}(t) = \boldsymbol{\chi}(t) - \boldsymbol{\chi}^*_k(t) \tag{4}$$

Where $t$ is the current time, $t_{0,k+1}$ is the time at start of the new plan $k+1$, $\hat{\boldsymbol{\chi}}(t_{0,k+1})$ is the predicted state used as initial state for the next plan, $\boldsymbol{\chi}^*_k(t)$ is the desired state from the current plan, $\overline{\boldsymbol{\chi}}(t)$ is the current tracking error and $\boldsymbol{\alpha}$ is a scaling vector.

For the scaling, values of 0 means that the prediction relies only on the desired state while values of 1 fully take into account the tracking error and correspond to the actual state of the robot when prediction is with a short horizon. For each foot, the scaling vector is set to $\mathbf{1}$ if it is in stance and this phase corresponds to the same one at the beginning of the next plan, or $\mathbf{0}$ otherwise. For the base, the scaling factors are tuned between those two values to allow feedback

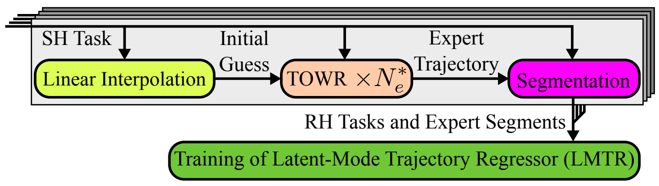


Fig. 5. Diagram of bulk data generation for imitation learning. Stationary-horizon tasks are sampled and extensively optimized into high-quality trajectories. The segmentation process converts data into receding-horizon segments and tasks, used for training the multi-modal trajectory regressor.

on the actual state while still being close enough to the previous plan for stable numerical behavior. This kind of simple prediction is only valid if the horizon is short with respect to the dynamics of the tracking error. To have a more accurate initial state for the optimized trajectory, the constraint on the initial state is updated with a new prediction between each iteration of the optimizer.

### *E. Terrain Perception*

The onboard RealSense depth cameras and the GridMap library [37] are used to perceive the terrain. GridMap constructs a 2.5D heightmap which constrains the footholds and a costmap to aid obstacle avoidance. A raw version of the heightmap is used to match the $z$ coordinate of the planned footholds to the actual terrain and a smooth version of the heightmap is used to set a cost on each candidate foothold. By making the cost proportional to the slope of the terrain, the solver is incentivized to choose footholds away from edges and inclined surfaces. In addition, the raw heightmap data is processed to detect edges which become part of the task input provided to the LMTR. Both the heightmap and the edges are updated at a frequency of 6 Hz.

### *F. Whole-Body Controller and Robustness to Terrain*

To execute the generated plans, the whole-body controller [38] is used to track the trajectory of the base and the end-effectors at 400 Hz. The controller includes different heuristics to improve its behavior in case of slip or other unexpected events.

To be more robust to errors in the terrain height measured by the depth cameras and to be able to make contact at the right instant, the swing trajectory has been modified. On the planner side, the formulation of splines in TOWR has been adapted to allow non-zero vertical velocities for the foot motions at the transition from swing to stance. Additionally, the controller is able to modify the setpoints of the feet computed by the planner by continuing the swing trajectory downward until contact is detected. Those two features allow our controller to quickly react to delayed contacts while having smooth foot trajectories despite the low frequency of the proposed planner.

## IV. Results and Discussion

The receding-horizon planner was tested with both trajectory initialization options, Heuristic and LMTR. The robot was tasked with computing dynamic motions toward a goal. The quality of the initial guesses and the refined trajectories were quantified by analyzing the cost and constraint violation computed at each iteration of the nonlinear solver. Execution of the replanning pipeline on the ANYmal C quadruped demonstrated its viability on hardware for most scenarios.

In the most challenging case—aggressive behaviors on terrain planned by LMTR—viability on hardware or in high-fidelity physics simulators has not been fully achieved. Instead, we provide numerical trials in which the replanner receives the task that would be seen if the previous plan were followed nominally for the duration of one replanning cycle, and refinement is conducted using the same computation time limit as hardware trials.

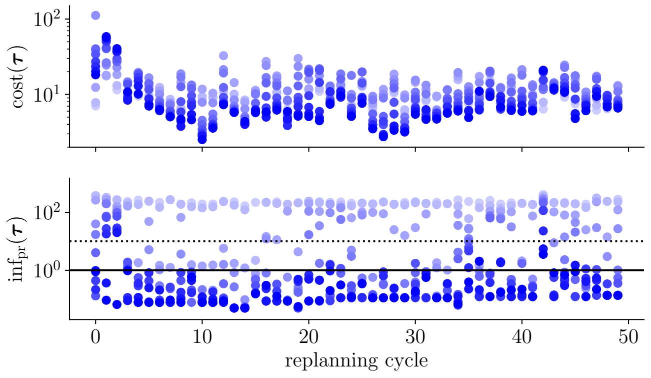


Fig. 6. The values of $\mathrm{cost}(\boldsymbol{\tau})$ and $\mathrm{inf}_{\mathrm{pr}}(\boldsymbol{\tau})$ during refinement of initial guesses provided by LMTR while traversing the terrain shown in Fig. 8 for 50 replanning cycles. The gradient (light to dark) represents subsequent iterations limited to a maximum of 10 per cycle. Horizontal lines show preferred (solid) and acceptable (dotted) thresholds of $\mathrm{inf}_{\mathrm{pr}}(\boldsymbol{\tau})$.

### *A. LMTR Setup and Numerical Performance*

An LMTR regressor was trained for both environment types, using $\mathbf{z} \in \mathbb{R}^{12}$ and with $\boldsymbol{\tau} \in \mathbb{R}^{312}$ containing a 1.2 s base horizon and stance-swing-stance phases. For the *flat-ground* scenario, the offline expert's initial conditions used random offsets of the lateral position and the yaw relative to a desired line of travel. Without any obstacles, $\mathbf{x} \in \mathbb{R}^{27}$.

The *terrain* scenario included randomized stairstep features $\mathbf{o}_i$ spaced apart by $0.25\,\mathrm{m}$ to $1\,\mathrm{m}$, with height changes of $-0.16\,\mathrm{m}$ to $0.16\,\mathrm{m}$ and edge orientations of $-15°$ to $15°$. During deployment, these terrain features are extracted from the elevation map, and the closest two are included in $\mathbf{x} \in \mathbb{R}^{35}$. Numerical results shown in Fig. 8 demonstrate motions computed at 2.5 Hz that reach mathematical feasibility under the SRBD formulation of Sec. III-A—attaining reliable deployment in physics simulators and on hardware is a prime focus of our ongoing work.

When facing a new scenario within the bounds of the experience set, the network consistently reconstructed a valid initial guess which was refined to a satisfactory level of optimality within the iteration budget available for online use, as shown in Fig. 6. Occasionally, some cycles presented conditions that were challenging to satisfy, potentially due to gaps in the regressor's coverage. If refinement failed, the remaining portion of the previous valid trajectory was used for an additional replanning cycle, which was possible due to a greater than 2-to-1 ratio of these two durations.

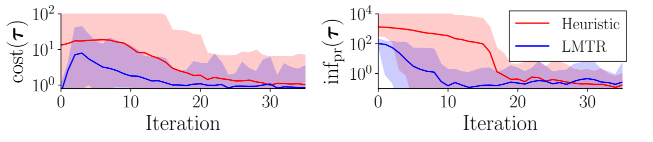


Fig. 7. Mean values of the cost and primal infeasibility at each refinement iteration of 1.2 s long initial guesses provided by the Heuristic (linear interpolation) and LMTR (near-optimal trajectories) on flat terrain at 2 Hz. Here, both cases use the same mid-motion final state $\chi_f$ intelligently provided by LMTR as shown in Fig. 4—a key strength of the regressor.

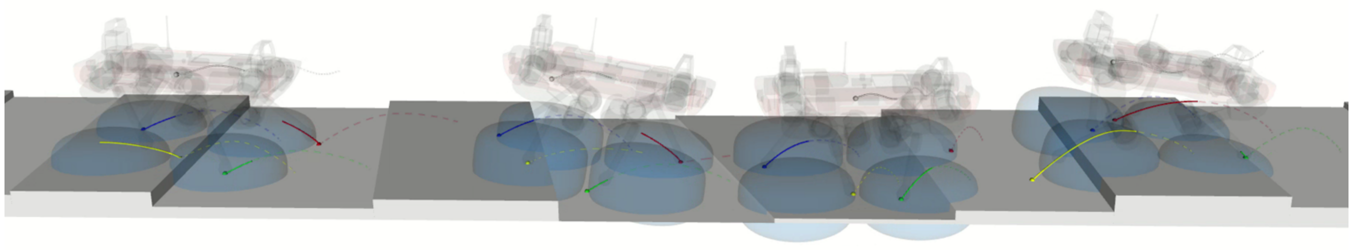


Fig. 8. Snapshots of a typical numerical trial of the LMTR which leverages similar prior experience to quickly plan very dynamic trajectories in a receding-horizon fashion while being informed by real-time perception. Steps were of arbitrary depth, height and orientation. The generated plans were refined online by the solver. The dashed lines indicate planned base and feet trajectories while the solid lines emphasize the portion of the plan executed during each MPC control cycle. The blue superquadrics approximate the operational space of the end-effectors by serving as range-of-motion constraints.

Figure 7 shows the evolution of the cost and the constraint violation along the iteration of the solver for the Heuristic and the LMTR initialization. While the Heuristic could be used with any predefined gait, to have a fair comparison with respect to the difficulty of the gait, both were evaluated on the contact timing and footholds generated by the regressor. The results show that the LMTR initialization was able to converge around 2 times faster than the Heuristic.

## B. Experimental Evaluation

*a) Heuristic initialization:* The initializations provided by the Heuristic, based on user-commanded target robot base pose, enabled the continuous execution of a walk and a trotting gait during which the robot traversed obstacles of heights up to $0.2\,\mathrm{m}$ as shown in Fig. 1. The base tracking error is shown in Fig. 9 for a hardware experiment where the robot walked over obstacles up to $0.13\,\mathrm{m}$ high for over $5.5\,\mathrm{min}$.

The replanning frequency for the walk gait was $1\,\mathrm{Hz}$ with an optimization horizon of $3\,\mathrm{s}$. The trot could run at $2\,\mathrm{Hz}$ over a replanning window of $1\,\mathrm{s}$. For both walk and trot, the Heuristic was operated at speeds of about $0.1\,\mathrm{m\,s^{-1}}$ on flat ground and while negotiating the obstacles shown in the accompanying video. The average step length on flat ground was $0.25\,\mathrm{m}$ for the trot and $0.3\,\mathrm{m}$ for the walk.

*b) LMTR initialization:* In the experiments with the *learned* initialization, the LMTR was used to enable a $4\,\mathrm{m}$, $12\,\mathrm{s}$ trot-like motion on flat ground which resulted in the base velocity in the $x$-direction $\dot{r}_x(t)$ reaching $0.7\,\mathrm{m\,s^{-1}}$ and swing lengths of $0.2\,\mathrm{m}$ to $0.6\,\mathrm{m}$ with durations $\Delta\mathbf{T}_{\text{sw}}$ of $0.3\,\mathrm{s}$ to $0.5\,\mathrm{s}$ . Thanks to the LMTR initialization, such a plan can be adapted online by the optimizer at frequencies from $2\,\mathrm{Hz}$ to $3.3\,\mathrm{Hz}$ with planning horizons of $1.2\,\mathrm{s}$ to take into account the model of the terrain received from the sensors and walk over obstacles as shown in Fig. 10.

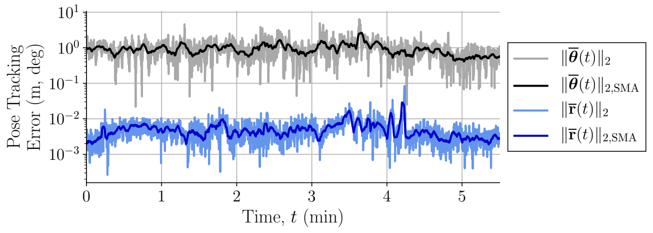


Fig. 9. Despite a low replanning frequency ($1.4\,\mathrm{Hz}$) and a large time step $dt_{\text{base-poly}}$ ($0.1\,\mathrm{s}$), the direct collocation formulation produces trajectories which satisfy the dynamics to third-order enabling accurate and precise tracking by the whole-body controller. The figure shows base pose tracking errors, and their simple moving averages (SMA), or a $5.5\,\mathrm{min}$ execution of the MPC initialized by the Heuristic, on terrain, by the real robot. The root-mean-squared error (RMSE) values of the linear position and orientation were $6.55 \times 10^{-3}\,\mathrm{m}$ and $1.07°$, respectively.

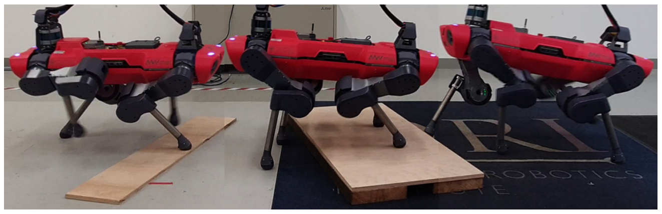


Fig. 10. The initial guesses provided by the LMTR trained on flat-ground can be adapted to mild height variation of $0.02\,\mathrm{m}$ to $0.08\,\mathrm{m}$ by the receding-horizon optimization, showing the ability to cope with minor imprecisions.

## V. Conclusion

This work presents a pipeline for iterative perception-aware planning of dynamic legged locomotion in a receding-horizon fashion via online trajectory optimization with a novel asynchronous, segmentation-based formulation using the single rigid body dynamics (SRBD) model. We present two initialization schemes. The *Heuristic* combines previous solutions with a linearly interpolated guess to accelerate the computation. The *latent-mode trajectory regressor (LMTR)*, which imitates expert data, is used to extend the planning horizon, proposing a behavior dependent on the task at hand, which may include prominent terrain features. We deploy this pipeline on a quadrupedal robot, ANYbotics ANYmal C, using fully onboard sensing and computation. Hardware trials with the Heuristic demonstrate the ability to traverse terrain obstacles using walking and trotting gaits. The LMTR enabled more dynamic, trot-like locomotion but with aperiodic contact sequences that hint at its capacity to express a diverse inventory of viable behaviors. Numerical results show how its expressiveness is suitable for adapting to challenging terrain while moving at fast pace. Future work will seek to realize these precise and highly dynamic motions by explicitly compensating for the approximations in the planning model, in lower-level control, and by updating the LMTR based on deployed experiences.

## References

[1] 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.

[2] N. Jetchev and M. Toussaint, "Fast motion planning from experience: Trajectory prediction for speeding up movement generation," *Autonomous Robots*, vol. 34, no. 1, pp. 111–127, Jan. 2013.

[3] N. Mansard, A. DelPrete, M. Geisert, S. Tonneau, and O. Stasse, "Using a Memory of Motion to Efficiently Warm-Start a Nonlinear Predictive Controller," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, May 2018, pp. 2986–2993.

[4] T. S. Lembono, C. Mastalli, P. Fernbach, N. Mansard, and S. Calinon, "Learning How to Walk: Warm-Starting Optimal Control Solver with Memory of Motion," *arXiv:2001.11751 [cs, math]*, Feb. 2020.

[5] O. Melon, M. Geisert, D. Surovik, I. Havoutis, and M. Fallon, "Reliable Trajectories for Dynamic Quadrupeds using Analytical Costs and Learned Initializations," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, May 2020, pp. 1410–1416.

[6] C. Lynch, M. Khansari, T. Xiao, V. Kumar, J. Tompson, S. Levine, and P. Sermanet, "Learning Latent Plans from Play," in *Conference on Robot Learning*, May 2020, pp. 1113–1132.

[7] D. Surovik, O. Melon, M. Geisert, M. Fallon, and I. Havoutis, "Learning an Expert Skill-Space for Replanning Dynamic Quadruped Locomotion over Obstacles," in *2020 Conference on Robot Learning (CoRL)*, Nov. 2020.

[8] H. Dai, A. Valenzuela, and R. Tedrake, "Whole-body motion planning with centroidal dynamics and full kinematics," in *2014 IEEE-RAS International Conference on Humanoid Robots*, 2014, pp. 295–302.

[9] J. Carpentier and N. Mansard, "Multi-contact locomotion of legged robots," *IEEE Transactions on Robotics*, vol. 34, no. 6, pp. 1441–1460, Dec. 2018.

[10] 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 *2020 IEEE International Conference on Robotics and Automation (ICRA)*, May 2020, pp. 2536–2542.

[11] D. Orin, A. Goswami, and S. Lee, "Centroidal dynamics of a humanoid robot," *Autonomous Robots*, vol. 35, pp. 161–176, 2013.

[12] B. Ponton, A. Herzog, A. Del Prete, S. Schaal, and L. Righetti, "On time optimization of centroidal momentum dynamics," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, May 2018, pp. 5776–5782.

[13] M. Posa, C. Cantu, and R. Tedrake, "A direct method for trajectory optimization of rigid bodies through contact," *The International Journal of Robotics Research*, vol. 33, no. 1, pp. 69–81, 2014.

[14] I. Mordatch, E. Todorov, and Z. Popović, "Discovery of complex behaviors through contact-invariant optimization," *ACM Trans. Graph.*, vol. 31, no. 4, July 2012.

[15] M. Vukobratovic and B. Borovac, "Zero-Moment Point — Thirty five years of its life," *International Journal of Humanoid Robotics*, vol. 01, no. 01, pp. 157–173, 2004.

[16] J. Di Carlo, P. M. Wensing, B. Katz, G. Bledt, and S. Kim, "Dynamic locomotion in the MIT cheetah 3 through convex model-predictive control," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 1–9.

[17] G. Bledt and S. Kim, "Implementing Regularized Predictive Control for Simultaneous Real-Time Footstep and Ground Reaction Force Optimization," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Nov. 2019, pp. 6316–6323.

[18] P. Fankhauser, M. Bjelonic, C. D. Bellicoso, T. Miki, and M. Hutter, "Robust Rough-Terrain Locomotion with a Quadrupedal Robot," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, May 2018, pp. 5761–5768.

[19] F. Jenelten, T. Miki, A. E. Vijayan, M. Bjelonic, and M. Hutter, "Perceptive Locomotion in Rough Terrain – Online Foothold Optimization," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5370–5376, Oct. 2020.

[20] O. Villarreal, V. Barasuol, P. M. Wensing, D. G. Caldwell, and C. Semini, "MPC-Based Controller with Terrain Insight for Dynamic Legged Locomotion," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, May 2020, pp. 2436–2442.

[21] D. Kim, D. Carballo, J. D. Carlo, B. Katz, G. Bledt, B. Lim, and S. Kim, "Vision Aided Dynamic Exploration of Unstructured Terrain with a Small-Scale Quadruped Robot," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, May 2020, pp. 2464–2470.

[22] A. Herdt, N. Perrin, and P. Wieber, "Walking without thinking about it," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct. 2010, pp. 190–195.

[23] M. Naveau, M. Kudruss, O. Stasse, C. Kirches, K. Mombaur, and P. Souères, "A Reactive Walking Pattern Generator Based on Nonlinear Model Predictive Control," *IEEE Robotics and Automation Letters*, vol. 2, no. 1, pp. 10–17, Jan. 2017.

[24] A. W. Winkler, F. Farshidian, M. Neunert, D. Pardo, and J. Buchli, "Online walking motion and foothold optimization for quadruped locomotion," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, May 2017, pp. 5308–5313.

[25] F. Farshidian, E. Jelavic, A. Satapathy, M. Giftthaler, and J. Buchli, "Real-time motion planning of legged robots: A model predictive control approach," in *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*, 2017, pp. 577–584.

[26] T. Apgar, P. Clary, K. Green, A. Fern, and J. Hurst, "Fast Online Trajectory Optimization for the Bipedal Robot Cassie," in *Robotics: Science and Systems XIV*, vol. 14, June 2018.

[27] B. Aceituno-Cabezas, C. Mastalli, H. Dai, M. Focchi, A. Radulescu, D. G. Caldwell, J. Cappelletto, J. C. Grieco, G. Fernández-López, and C. Semini, "Simultaneous contact, gait, and motion planning for robust multilegged locomotion via mixed-integer convex optimization," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 2531–2538, 2018.

[28] O. Cebe, C. Tiseo, G. Xin, H.-c. Lin, J. Smith, and M. Mistry, "Online Dynamic Trajectory Optimization and Control for a Quadruped Robot," *arXiv:2008.12687 [cs]*, Sept. 2020.

[29] G. Bledt and S. Kim, "Extracting Legged Locomotion Heuristics with Regularized Predictive Control," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, May 2020, pp. 406–412.

[30] V. Tsounis, M. Alge, J. Lee, F. Farshidian, and M. Hutter, "DeepGait: Planning and Control of Quadrupedal Gaits Using Deep Reinforcement Learning," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3699–3706, Apr. 2020.

[31] Z. Xie, H. Y. Ling, N. H. Kim, and M. van de Panne, "ALLSTEPS: Curriculum-Driven Learning of Stepping Stone Skills," *arXiv:2005.04323 [cs]*, Aug. 2020.

[32] A. Duburcq, Y. Chevaleyre, N. Bredeche, and G. Boéris, "Online Trajectory Planning Through Combined Trajectory Optimization and Function Approximation: Application to the Exoskeleton Atalante," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, May 2020, pp. 3756–3762.

[33] S. Tonneau, A. Del Prete, J. Pettré, C. Park, D. Manocha, and N. Mansard, "An Efficient Acyclic Contact Planner for Multiped Robots," *IEEE Transactions on Robotics*, vol. 34, no. 3, pp. 586–601, June 2018.

[34] H. Zhang, S. Starke, T. Komura, and J. Saito, "Mode-adaptive neural networks for quadruped motion control," *ACM Transactions on Graphics*, vol. 37, no. 4, pp. 1–11, Aug. 2018.

[35] S. Starke, Y. Zhao, T. Komura, and K. Zaman, "Local motion phases for learning multi-contact character movements," *ACM Transactions on Graphics*, vol. 39, no. 4, pp. 54:54:1–54:54:13, July 2020.

[36] J. Carius, F. Farshidian, and M. Hutter, "MPC-Net: A First Principles Guided Policy Search," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2897–2904, Apr. 2020.

[37] P. Fankhauser and M. Hutter, "A Universal Grid Map Library: Implementation and Use Case for Rough Terrain Navigation," in *Robot Operating System (ROS): The Complete Reference (Volume 1)*, ser. Studies in Computational Intelligence, A. Koubaa, Ed. Cham: Springer International Publishing, 2016, pp. 99–120.

[38] C. Dario Bellicoso, C. Gehring, J. Hwangbo, P. Fankhauser, and M. Hutter, "Perception-less terrain adaptation through whole body control and hierarchical optimization," in *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*. Cancun, Mexico: IEEE, Nov. 2016, pp. 558–564.