2020 IEEE International Conference on Robotics and Automation (ICRA)
31 May - 31 August, 2020. Paris, France

Extracting Legged Locomotion Heuristics with Regularized Predictive Control

Gerardo Bledt[1] and Sangbae Kim[1]

***Abstract*— Optimization based predictive control is a powerful tool that has improved the ability of legged robots to execute dynamic maneuvers and traverse increasingly difficult terrains. However, it is often challenging and unintuitive to design meaningful cost functions and build high-fidelity models while adhering to timing restrictions. A novel framework to extract and design principled regularization heuristics for legged locomotion optimization control is presented. By allowing a simulation to fully explore the cost space offline, certain states and actions can be constrained or isolated. Data is fit with simple models relating the desired commands, optimal control actions, and robot states to identify new heuristic candidates. Basic parameter learning and adaptation laws are then applied to the models online. This method extracts simple, but powerful heuristics that can approximate complex dynamics and account for errors stemming from model simplifications and parameter uncertainty without the loss of physical intuition while generalizing the parameter tuning process. Results on the Mini Cheetah robot verify the increased capabilities due to the newly extracted heuristics without any modification to the controller structure or gains.**

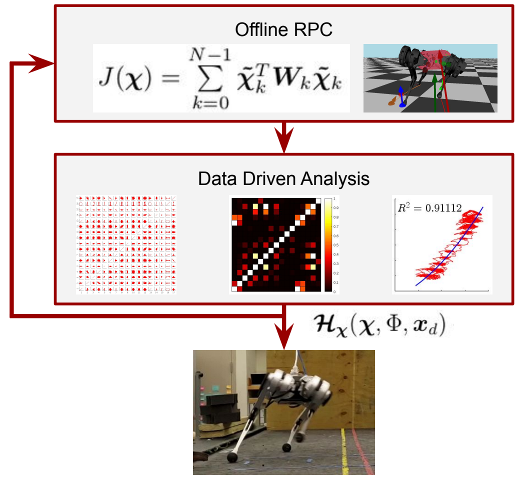


Fig. 1: **Data Driven Heuristic Extraction.** Methods presented in this paper improve the robot's capabilities by extracting heuristic models from simulated data.

I. INTRODUCTION

Legged robots have the potential to be highly dynamic machines capable of outperforming humans and animals in locomotion tasks over irregular and dangerous environments. Unfortunately, they have not yet delivered on that promise as they still lack the agility and robustness needed to traverse arbitrary terrains with the same grace as animals. However, recent advances in the field are showing increasingly impressive behaviors and robustness to both external disturbances and unexpected obstacles.

Heuristic-based legged locomotion control has shown success in traversing rough terrains, particularly with degraded perception. Since experts already have knowledge about locomotion, it is often a better choice to simply encode intelligent heuristics rather than leave it to an optimization solver. Focchi at al. demonstrate a heuristic planning framework on the HyQ robot to overcome uncertain conditions [1].

Whole-body optimization-based controls using a full dynamics model have been successful in both humanoids [2], [3] and quadrupeds [4], [5]. Model Predictive Control (MPC) approaches have been gaining popularity for their ability use simple control models for prediction, while still achieving good performance [6], [7], [8]. Results on the MIT Cheetah 3 robot platform using a linear convex MPC framework demonstrated highly dynamic, robust locomotion with various gaits [9], [10].

Regularized Predictive Control (RPC) allows known dynamics and locomotion principles to be exploited through heuristic regularization. Previous work on a model of the MIT Cheetah 2 showed the benefits of both heuristics and optimization by injecting heuristic regularization into predictive control. This increased the robot's ability to reject disturbances and decreased computation time as well as frequency of undesirable local minima [11]. The robot was able to stabilize multiple gaits at a variety of different velocities and environmental uncertainty using a single cost function and set of gains across all situations. Further work proved this type of controller to be tractable on the MIT Cheetah 3 quadruped hardware platform [12].

Parameter adaptation and automatic tuning is notably difficult in legged systems because of the rapid, harsh discrete mode changes as feet enter and exit stance and swing phases. Early work on automatic tuning for robotic systems concluded that because of the discontinuous gradients it was best to develop a controller able to move stably before employing the parameter tuning [13]. Other work successfully used evolutionary optimization that did not attempt to approximate gradients to learn to walk [14], while another showed an automatic gain tuning method vastly improved the robot's performance without tedious hand tuning [15].

Recently, neural networks have shown promising results for learning control policies through simulation. Hwangbo et al. show a simulation-based learned policy ported over for real hardware locomotion [16]. Work from DeepMind has shown the ability to learn to move in complex environments on various different dynamic systems [17]. However, as is often a drawback of neural networks, it is not always clear how to modify specific policy parameters in order to tune the behaviors without the need to retrain the entire network.

[1]Department of Mechanical Engineering, MIT, Cambridge, MA 02139, USA: gbledt@mit.edu, sangbae@mit.edu

A. Contributions

The main contribution of the paper is a framework that provides methods for the extraction of simple legged locomotion heuristics from robot experience data. The result is autonomous discovery of simple, but powerful regularization heuristics that improve the performance of the RPC without any explicit changes to control gains or controller structure. Each simple heuristic is generalizable, easily tunable, and carries a clear physical meaning. With newly extracted heuristics from using this framework, the robot is able to spin in place rapidly at $4\frac{rad}{s}$ and turn at $2\frac{rad}{s}$ while moving at a high speed of $1.5\frac{m}{s}$. Previously, these maneuvers were could not be executed stably with the exact same controller. The heuristics found through the framework are generalizable and adequate for locomotion under unexpected disturbances.

II. Regularized Predictive Control

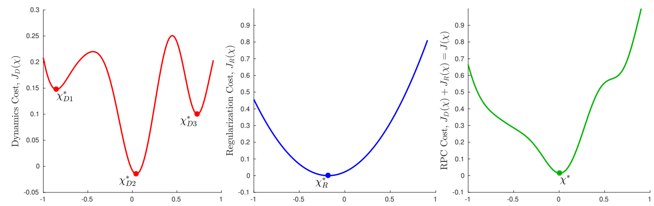
As opposed to traditional MPC techniques, RPC directly exploits simple heuristics to simplify complex cost spaces and find feasible solutions quickly. Rather than treating the problem of robot locomotion as a black box optimization, simple physics-based heuristics are encoded into the cost function and constraints to bias the optimization towards a sensible solution while remaining free to explore the surrounding cost space for the possibility of a better result.

Heuristics for the decision variables, $\boldsymbol{\chi}$, are embedded directly into the cost function through error terms

$$\tilde{\boldsymbol{\chi}} = \boldsymbol{\mathcal{H}}_{\boldsymbol{\chi}}(\boldsymbol{\chi}, \boldsymbol{\Phi}, \boldsymbol{x}_d) - \boldsymbol{\chi} \quad (1)$$

where the heuristics $\boldsymbol{\mathcal{H}}_{\boldsymbol{\chi}}$ are functions of the robot states, $\boldsymbol{x} = \begin{bmatrix}\boldsymbol{p}^T & \boldsymbol{\Theta}^T & \dot{\boldsymbol{p}}^T & \dot{\boldsymbol{\Theta}}^T\end{bmatrix}^T$, desired states, $\boldsymbol{x}_d$, and the scheduled gait phase, $\boldsymbol{\Phi}$, as defined in [18]. The robot model is chosen to be a simple lumped mass model with a massless leg assumption making the states the position of the CoM, $\boldsymbol{p}$, the roll-pitch-yaw Euler Angle CoM orientation, $\boldsymbol{\Theta}$, the velocity in the world frame, $\dot{\boldsymbol{p}}$, and the Euler angle rates, $\dot{\boldsymbol{\Theta}}$. Inputs are chosen to be the foot location vectors for each foot i from the CoM, $\boldsymbol{r}_i$, and the corresponding ground reaction force, $\boldsymbol{f}_i$, which are ordered in the input vector $\boldsymbol{u} = \begin{bmatrix}\boldsymbol{r}_1^T & \boldsymbol{f}_1^T & \dots & \boldsymbol{r}_4^T & \boldsymbol{f}_4^T\end{bmatrix}^T$. The combination of states and inputs for all predicted timesteps are chosen to be the decision variables, $\boldsymbol{\chi}_k = \begin{bmatrix}\boldsymbol{x}_k^T & \boldsymbol{u}_k^T\end{bmatrix}^T$. The power of the RPC framework is that it allows extremely rich information about the robot to be embedded in the optimization with no modification to the controller structure.

The inputs are solved for through a nonlinear optimization using a direct transcription method [19]. The objective is to minimize the quadratic cost function on the deviation from the state and input heuristics. We pose the optimization-based control essentially as a nonlinear MPC that relies heavily on regularization heuristics rather than an accurate model or realistic dynamics. In fact, the model has massless legs and almost completely linearized dynamics as described in [11]. Generally it can be written



(a) $J_D(\boldsymbol{\chi})$ (b) $J_R(\boldsymbol{\chi})$ (c) $J(\boldsymbol{\chi})$

Fig. 2: **Regularizing Nonlinear Cost Functions.** A generic nonlinear cost function (2a) with multiple local minima is additively regularized with a simple quadratic cost function (2b). The resulting cost function (2c) captures a smoothened nonlinear function with only one minimum.

$$\begin{aligned}
\min_{\boldsymbol{\chi}} \quad & J(\boldsymbol{\chi}) = \sum_{k=0}^{N-1} \tilde{\boldsymbol{\chi}}_k^T \boldsymbol{W}_k \tilde{\boldsymbol{\chi}}_k \\
\text{s.t.} \quad & \boldsymbol{x}_{k+1} = \boldsymbol{A}_k \boldsymbol{x}_k + \boldsymbol{B}_k h(\boldsymbol{\chi}_k, \boldsymbol{\Phi}_k) + \boldsymbol{d}_k \\
& \boldsymbol{\zeta}_k(\boldsymbol{\chi}_k, \boldsymbol{\Phi}_k) \leq 0 \\
& \boldsymbol{\zeta}'_k(\boldsymbol{\chi}_k, \boldsymbol{\chi}_{k+1}, \boldsymbol{\Phi}_k, \boldsymbol{\Phi}_{k+1}) \leq 0
\end{aligned}$$

where $\boldsymbol{\chi}_k = \begin{bmatrix}\boldsymbol{x}_k^T & \boldsymbol{u}_k^T\end{bmatrix}^T$, $\forall k \in \{0, ..., N-1\}$ are decision variables and the number of timesteps, N, is user selected. Physical feasibility constraints are $\boldsymbol{\zeta}$ and $\boldsymbol{\zeta}'$ as described in [12], and the gait phase is $\boldsymbol{\Phi}$ as introduced in [18]. The simplified discrete dynamics matrices are written

$$\boldsymbol{A}_k = \begin{bmatrix}\mathbf{I}_6 & dt_k \mathbf{I}_6 \\ \mathbf{0}_6 & \mathbf{I}_6\end{bmatrix} \boldsymbol{B}_k = \begin{bmatrix}\frac{dt_k^2}{2}\boldsymbol{I}^{-1} \\ dt_k \boldsymbol{I}^{-1}\end{bmatrix} \boldsymbol{d}_k = \begin{bmatrix}\frac{dt_k^2}{2}\boldsymbol{a}_g \\ dt_k \boldsymbol{a}_g\end{bmatrix}$$

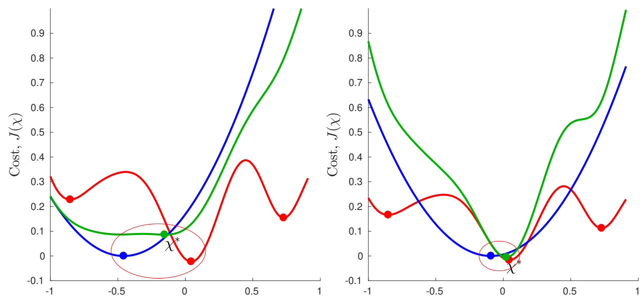
where $\boldsymbol{I}$ is the inertia tensor and $\boldsymbol{a}_g$ is the gravity vector. The nonlinear inputs make up the combined forces and torques on the CoM as

$$\begin{aligned}
h(\boldsymbol{\chi}_k, \boldsymbol{\Phi}_k) &= \begin{bmatrix}\boldsymbol{f} \\ {}^{\mathcal{B}}\boldsymbol{\tau}\end{bmatrix} \\
&= \sum_{i=1}^{4} \begin{bmatrix}\mathbf{I}_3 \\ \boldsymbol{R}(\boldsymbol{\Theta}_k)^T \left[\boldsymbol{r}_{i,k}\right]_\times\end{bmatrix} \boldsymbol{s}_{\boldsymbol{\Phi},i,k} \boldsymbol{f}_{i,k}
\end{aligned}$$

with $\boldsymbol{R}(\boldsymbol{\Theta}_k)$ being the rotation matrix of the CoM, and $\boldsymbol{s}_{\boldsymbol{\Phi},i,k} \in \{0 : \text{swing}, 1 : \text{contact}\}$ being the boolean contact state dependent on the scheduled gait phase of each foot.

To demonstrate the effects of regularization, a simple example is depicted in Figure 2. The underlying cost function (2a) is chosen to be nonlinear with multiple local minima. A quadratic regularization function (2b) is designed to be additive and smooth out the dynamics cost where the result is a well-conditioned combined cost function (2c) with no extraneous local minima. Through regularization, the shape of the cost function is changed without canceling out the original nonlinear function.

Although these heuristics benefit the RPC, designing them in practice is challenging and can be unintuitive. Simply adding a regularization function does not improve the optimization problem. In fact, it can often yield detrimental



(a) Bad Regularization (b) Good Regularization

Fig. 3: **Designing Regularization Functions.** Regularization functions should smooth the dynamics function while not having a significant effect on the optimal solution.

results. Figure 3 shows how a poorly designed regularization function (3a) can pull the combined minimum away from the underlying optimum, whereas a well designed regularization function (3b) simplifies the surrounding cost space, but does not affect the optimal solution. The key is to design good heuristics that meaningfully shape the cost space.

Qualitatively, a heuristic is good if it causes the system to behave as desired when executed. Quantitatively, a set of heuristics is considered good if the optimal solution of the underlying function, the heuristic regularization portion, and combined cost function are all approximately equal to each other for all states within the set of desired reasonable locomotion states written as

$$\boldsymbol{\chi}_R^* \approx \boldsymbol{\chi}_D^* \approx \boldsymbol{\chi}^*, \forall \{\boldsymbol{x} | \boldsymbol{x} \in \boldsymbol{X}\} \tag{2}$$

This viable operating space, $\boldsymbol{X}$, is intentionally defined vaguely allowing for different desired performance metrics. For example, the set of heuristics for quasi-static locomotion may not be adequate for more dynamic movement. What may be considered failure for one robot, may be perfectly within reasonable locomotion for another robot. While the individual parts of the cost function may look very different from each other, the optimal solutions for each part should be near each other in most cases.

III. Extracting Heuristics

To design these heuristics, we propose using the framework described in Figure 4. The framework pairs expert design with data-driven analysis. Humans provide natural intuition that algorithms may not find as easily. Meanwhile, computers can analyze large amounts of data and recognize patterns automatically that may not be obvious to an expert. The framework seeks to exploit the advantages of each method to build the richness of the regularization heuristics that will be embedded in the robot for better online control. Expert design encompasses observing nature and analyzing simple physics. For example, the notion of vertical impulse scaling [20], or the widely used capture point [21]. These required experts in the field to make clever assumptions and simplifications for force generation and footstep placement in specific cases.

We propose a more autonomous method to quickly discover these relationships. This paper will focus on using the data gathered from simulations running the RPC offline in different scenarios. Since the solution times do not need to adhere to any real-time constraints offline, the cost space can be fully explored for nonintuitive local minima and tested for generality. Running the controller in a simulated environment also allows restriction of certain states to investigate the performance without some of the nonlinear coupling effects that would be present in the real system.

The states, inputs, commands, gait schedule, and combinations of each from the simulations make up the set

$$\boldsymbol{V} = \begin{bmatrix} \boldsymbol{x}^T & \boldsymbol{u}^T & \boldsymbol{x}_d^T & \boldsymbol{\Phi}^T & (\dot{\boldsymbol{p}} \times \dot{\boldsymbol{\Theta}})^T & t & \ldots \end{bmatrix}^T \tag{3}$$

to be tested as potential functions of each other. Various simple model fits are applied to the data sets and highly correlated models are identified as candidates for a new heuristic policy. These candidates are further analyzed for repeatability and compared with the results of the predictive optimization to determine if the proposed model is a valid extracted heuristic. The framework computes model fits on the robot data during simulation where forward, lateral, and turning rate commands are provided.

The fits tested for statistical correlation are polynomial and sum of sines fits where the model is described as

$$\text{Polynomial:} \quad \mathcal{H}_{v_i}(v_j) = \sum_{n=0}^{D \in [0,9]} a_n^{v_i} v_j^n \tag{4}$$

$$\text{Sum of Sines:} \quad \mathcal{H}_{v_i}(v_j) = \sum_{n=1}^{D \in [1,8]} b_n^{v_i} \sin\left(c_n^{v_i} v_j + d_n^{v_i}\right) \tag{5}$$

with D being the degree of polynomial and the number of summed sine terms respectively for the heuristic model $\mathcal{H}_{v_i}(v_j)$ where the dependent variable, v_i, is written as a function of the independent variable, v_j and both are taken from $\boldsymbol{V}$. These models were deliberately chosen over more complicated highly nonlinear functions or neural networks because they are easily tunable and can provide intuition for the underlying physical relationships. It is straightforward to determine the effects of varying the parameters of a polynomial or sine wave model, while approximating highly nonlinear behaviors.

Figure 5 shows an example result for the R^2 values for a first order polynomial model fit on data gathered from varying the commanded forward speed of the robot. Similar plots were made for the various model fits in (4) and (5). In the past, some of these relationships were discovered through experimental observation and parameters were hand tuned, but with this framework, the model types and associated parameters automatically discovered. This simplifies the process, while retaining intuition over the robot control.

IV. Evaluating Heuristics

Extracted heuristic models are dependent on the exploration done in simulation, as well as the choice of dependent and independent variables to test for correlation. This means

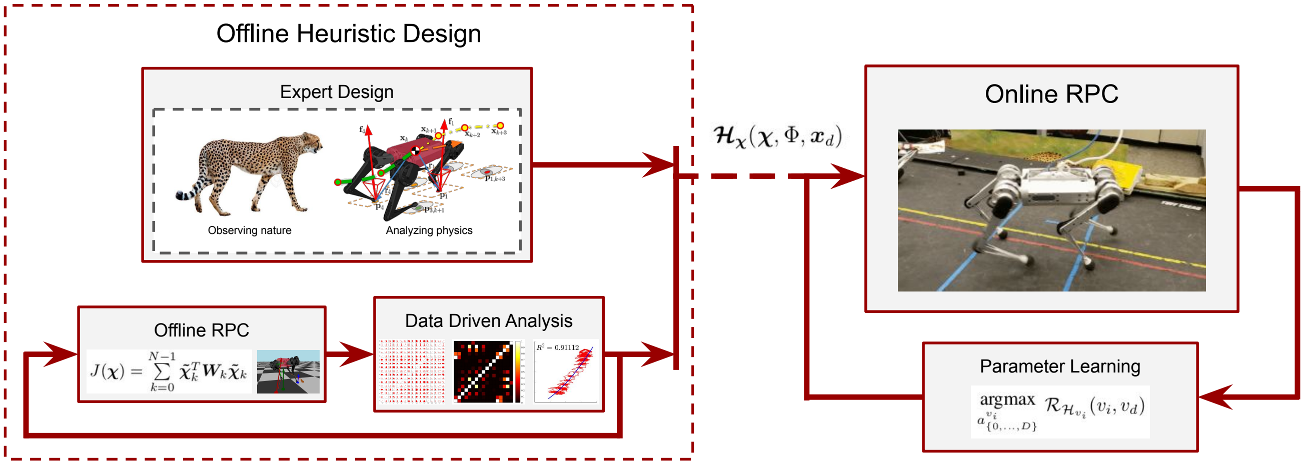
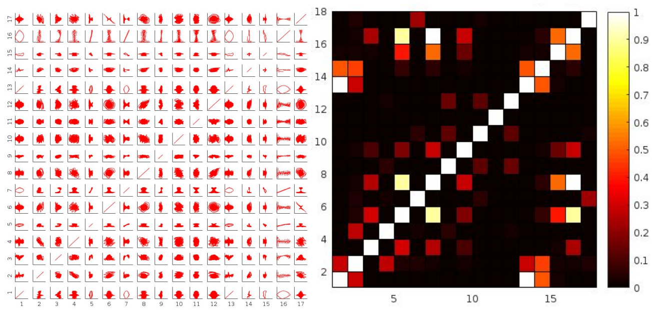
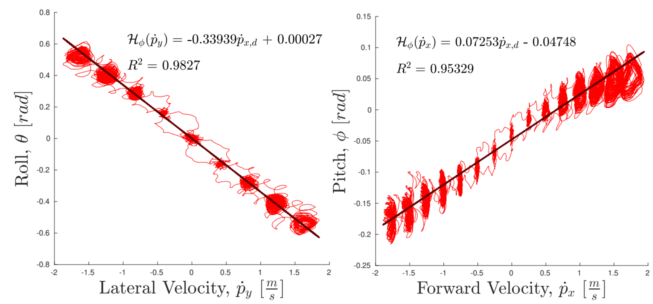


Fig. 4: **Heuristic Extraction Framework.** Expert design and data analysis from running the RPC offline both yield heuristics that can be used to inform the controller online. Basic Reinforcement Learning adapts parameters to account for model inaccuracies and timing discretization delays that prove to be difficult to model.



(a) Variable Relationships (b) Heuristic Candidates R^2

Fig. 5: **Identifying Heuristic Model Candidates.** Model fits are computed for each of the variable relationships in 6a to identify new heuristic candidates for a subset of $\boldsymbol{V}$. The R^2 values for a 1^{st} order polynomial fit are seen in 6b.



(a) $\mathcal{H}_\theta(\dot{p}_y)$ (b) $\mathcal{H}_\phi(\dot{p}_x)$

Fig. 6: **Linear Velocity Induced Orientation.** The framework identified the tendency towards linear orientation offset in pitch and roll with faster forward and lateral trotting.

that while the framework can identify potential candidate models between two states, they do not necessarily represent meaningful heuristics. It is still necessary to have a knowledgeable expert design exploration methods and differentiate between models with overfitting or false dependencies.

Two of the stronger correlations found were linear dependencies of roll, θ, on the robot's lateral velocity, $\dot{p}_y$, and pitch, ϕ, based on the forward velocity, $\dot{p}_x$. The fits had an R^2 value of 0.9827 and 0.9533 respectively. These were found to be

$$\mathcal{H}_\theta(\dot{p}_y) = a_1^\theta \dot{p}_y + a_0^\theta = -0.339\dot{p}_y + 0.00027 \tag{6}$$
$$\mathcal{H}_\phi(\dot{p}_x) = a_1^\phi \dot{p}_x + a_0^\phi = 0.073\dot{p}_x - 0.0475 \tag{7}$$

and shown in Figure 6 for both cases. The dominant linear nature of the relationships is clear, although this does not fully fit the data.

The same simulations were run again and analyzed, but this time new heuristics were discovered. Namely, a sinusoidal relationship for the periodic pitch behavior as a function of the gait phase with $R^2 = 0.835$ described by the new heuristic

$$\begin{aligned}\mathcal{H}_\phi(\boldsymbol{\Phi}) &= b_1^\phi \sin(c_1^\phi \boldsymbol{\Phi} + d_1^\phi) \\ &= 0.023 \sin(12.553\boldsymbol{\Phi} + 2.999).\end{aligned} \tag{8}$$

The results of which are depicted in Figure 7. Interestingly, this matches the intuition for a periodic pitch limit cycle during a trot gait where $c_1 = 2\pi f$ and the two diagonal pairs of legs perform complementary actions over a single gait cycle, $f = 2$, giving $c_1^\phi \approx 12.566$.

By combining the two extracted heuristics for pitch in equations (7) and (8), we get

$$\begin{aligned}\mathcal{H}_\phi(\dot{p}_x, \boldsymbol{\Phi}) &= \mathcal{H}_\phi(\dot{p}_x) + \mathcal{H}_\phi(\boldsymbol{\Phi}) \\ &= a_1\dot{p}_x + a_0 + b_1 \sin(c_1\boldsymbol{\Phi} + d_1).\end{aligned} \tag{9}$$

Figure 8 shows that although the commanded robot speed is $1.5\frac{m}{s}$ with no pitch, the robot's dynamics vary in practice. However using heuristic (9) we can approximate the complex dynamics that take place naturally using only the simple intuitive models that are easy to modify.

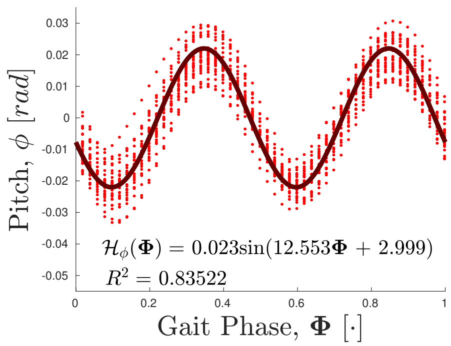


Fig. 7: **Periodic Pitch Behavior.** The robot exhibits natural pitch dynamics that are be extracted from simulation data.

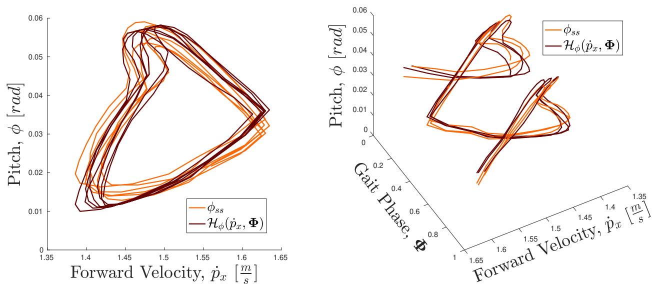


Fig. 8: **Intuitive Approximation.** The dark red shows the adjusted reference pitch during a steady state trot from the simple heuristics and matches closely to the actual pitch of the robot in orange from a commanded $1.5\frac{m}{s}$ forward speed.

Since the data was gathered in simulation without real-time computation restrictions, the parameters used for the heuristics may often differ on the actual system or even on the same simulated model with real-time restrictions in place. Therefore, a very simple parameter learning law was developed to adjust the values during operation. A reward function is defined as

$$\mathcal{R}_{\mathcal{H}_{v_i}} = -\left\|\sum_{\Phi}(v_d - v_i)\right\| \tag{10}$$

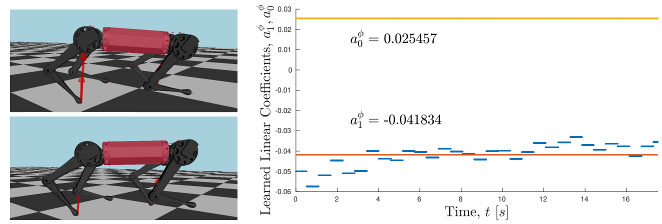
where the robot is rewarded for tracking the desired behavior summed over a gait period, $\sum\limits_{\Phi}$. The parameter update law for the linear model case is written to be

$$\begin{cases} a_{0,k}^{v_i} = a_{0,k-1}^{v_i} - \nabla\mathcal{R}_{\mathcal{H}_{v_i}} & , |v_j| < v_{j,min} \\ a_{1,k}^{v_i} = a_{1,k-1}^{v_i} - \frac{\nabla\mathcal{R}_{\mathcal{H}_{v_i}}}{v_j} & , |v_j| \geq v_{j,min}. \end{cases} \tag{11}$$

This adaptation law was able to successfully learn parameters that reduced errors in the robot at runtime as seen in Figure 9.

V. Results

The exact reason for the velocity induced orientation errors is not immediately clear, but can likely be a combination of various factors including unmodeled dynamics, incorrect friction parameters, timestep discretization, optimization solve time, or something we have not thought of. With the heuristic implemented on the robot, pitch and roll errors



(a) Pitch Compensation (b) Learned Parameter Adaptation

Fig. 9: **Learned Pitch Compensation Parameters.** Both parameters a_0^ϕ and a_1^ϕ are learned online to satisfy the reward function and averaged to be constants in the robot's memory.

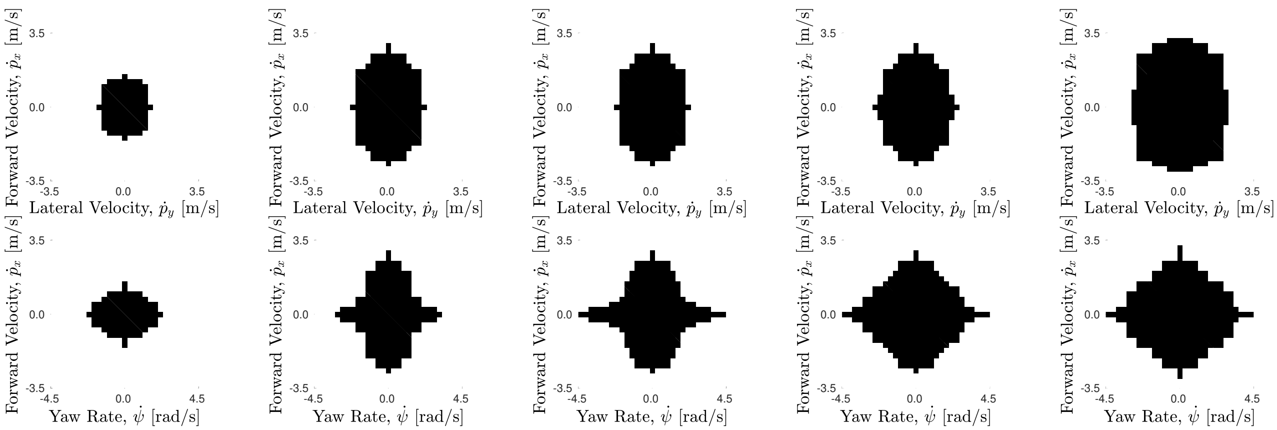
disappeared with a pitch error of $0.16 \pm 0.52°$ for forward velocities of $\dot{p}_x \in \left[-2, 2\right] \frac{m}{s}$. Regardless of the underlying cause of the errors, we are able to find a simple, easily tunable approximate model to compensate for them.

The same method was applied to iteratively to build the set of heuristics, $\boldsymbol{\mathcal{H}}$. In addition to the previously discussed pitch and roll compensation and the periodic orientation behavior, heuristics were extracted for foot placement during straight walking, in-place turning, and high speed turning, as well as a curious error where the robot began to dip in height as it moved faster. The extracted models are listed in Table I. Note that the simplest model was chosen to prevent overfitting despite many datasets also being slightly better approximated by higher degree polynomials and larger sum of sines.

Newly extracted heuristics were added to the RPC in simulation to verify that each one would generalize to situations other than the exploration in which they were extracted. As new heuristics are injected into the RPC optimization, the capabilities of the robot increase. In Figure 10 the robot was commanded forward and lateral velocities from $(\dot{p}_x, \dot{p}_y) \in \left[-3.5, 3.5\right] \frac{m}{s}$ and turning rates of $\dot{\psi} \in \left[-4.0, 4.0\right] \frac{rad}{s}$. Maximum forward and lateral velocities, as well as turning rate were all doubled when all models were added.

RPC was implemented on the MIT Cheetah 3 and Mini Cheetah robots using the IPOPT solver [22] in C++ with the optimization running at a situationally-variable frequency of 100-200Hz as described in [12]. New behaviors such as the rapid in-place turning and high speed turning which were not previously possible when using a naïve optimization are now able to be executed as seen in Figure 11. When turning in place, the feet are placed along the direction of the turn as the robot is given a command of $\dot{\psi}_d = 6.5\frac{m}{s}$. for high speed turning with commands of $\{\dot{p}_{x,d}, \dot{\psi}_d\} = \{1.5\frac{m}{s}, 2\frac{rad}{s}\}$, the foot placement is dominated by the need to apply centrifugal forces by placing them roughly out along the turning radius.

The controller generalizes to handle unexpected situations as the heuristics bias the optimization towards a solution without constraining it, meaning that the optimization can still search for better footstep locations and forces. The controller was tested for robustness by giving the robot an impulsive push as seen in Figure 12a, as well as commanded to trot forward at $\dot{p}_d = 1.5\frac{m}{s}$ over random debris scattered over the ground as seen in Figure 12b. In both cases, the robot is able to reject disturbances. More experiment videos

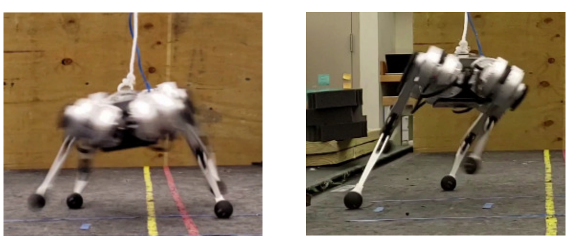


(a) No Heuristics (b) $\boldsymbol{\mathcal{H}_r}(\dot{\boldsymbol{p}})$ (c) $\boldsymbol{\mathcal{H}_r}(\dot{\psi})$ (d) $\boldsymbol{\mathcal{H}_r}(\dot{\boldsymbol{p}} \times \dot{\boldsymbol{\Theta}})$ (e) $\boldsymbol{\mathcal{H}_\Theta}(\dot{\boldsymbol{p}}, \boldsymbol{\Phi})$

Fig. 10: **Viable Operating Regions.** Each plot from 10a to 10e adds one of the heuristics discussed in the paper. As new heuristics are discovered and subsequently injected into the optimization, the set of viable operating regions where the robot is in stable locomotion increases. The dark areas signify where a fall did not occur, but does not qualify the movements.

TABLE I: Extracted Heuristics

Heuristic	Extracted Heuristic Model
Forward Stepping	$\boldsymbol{\mathcal{H}_r}(\dot{\boldsymbol{p}}) = \boldsymbol{a}_1^r \dot{\boldsymbol{p}} + \boldsymbol{a}_0^r$
In-Place Turning	$\boldsymbol{\mathcal{H}_r}(\dot{\psi}) = \boldsymbol{a}_1^r \dot{\psi} + \boldsymbol{a}_0^r$
High Speed Turning	$\boldsymbol{\mathcal{H}_r}(\dot{\boldsymbol{p}} \times \dot{\boldsymbol{\Theta}}) = \boldsymbol{a}_1^r(\dot{\boldsymbol{p}} \times \dot{\boldsymbol{\Theta}}) + \boldsymbol{a}_0^r$
Orientation Compensation	$\boldsymbol{\mathcal{H}_\Theta}(\dot{\boldsymbol{p}}) = \boldsymbol{a}_1^\Theta \dot{\boldsymbol{p}} + \boldsymbol{a}_0^\Theta$
Periodic Orientation	$\boldsymbol{\mathcal{H}_\Theta}(\boldsymbol{\Phi}) = \boldsymbol{b}_1^\Theta \sin(\boldsymbol{c}_1^\Theta \boldsymbol{\Phi} + \boldsymbol{d}_1^\Theta)$
Height Compensation	$\mathcal{H}_z(\dot{p}_x) = a_2^z \dot{p}_x^2 + a_1^z \dot{p}_x + a_0^z.$



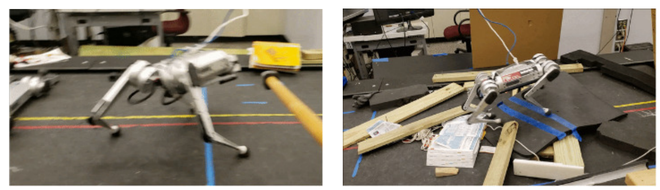
(a) In-Place Turning (b) High Speed Turning

Fig. 11: **Improved Performance.** Extracted heuristics allow the robot to spin in place at $2\pi \frac{rad}{s}$ (11a) and make high speed turns of $2\frac{rad}{s}$ while moving forward at $1.5\frac{m}{s}$ (11b).

are shown in the attached video [23]. Results support the goal of a framework to systematically improves robot performance by extracting heuristics that approximate unmodeled dynamics and compensate errors with simple models.

VI. Conclusion

The results have shown controller performance improved without any explicit changes to the control gains or the controller structure. Theoretically, if a set of heuristics, $\boldsymbol{\mathcal{H}}$, is found where $\boldsymbol{\chi}_R^* = \boldsymbol{\chi}_D^* = \boldsymbol{\chi}^*, \forall\{\boldsymbol{x}|\boldsymbol{x} \in \boldsymbol{X}\}$ rather than the approximate equation (2), then these heuristics are the solution to the optimization, $\boldsymbol{\mathcal{H}} = \boldsymbol{\mathcal{H}}^*$, for all of the viable operating conditions, $\boldsymbol{X}$. This then implies that there is no more need for heavy optimization as the optimal dynamics-based solution would always be equal to



(a) Push Rejection (b) Rough Terrain

Fig. 12: **Robust Locomotion.** Heuristics generalize to be robust to external disturbances and unstructured terrain.

the heuristic solution. Finding $\boldsymbol{\mathcal{H}}^*$ is likely unfeasible, but the proposed method can be used to build the set $\boldsymbol{\mathcal{H}}$, and better approximate $\boldsymbol{\mathcal{H}}^*$ for a large number of cases. This will take the burden off of the optimization as the initial guess and regularized solution will be closer to the actual dynamics-based optimal solution.

Recent work has shown great results in the area of robotic learning and is being considered as a possible enhancement to the framework. However, in this work we argue that it is important to have a good grasp of the physical intuition behind the controller. By choosing the set of heuristics, $\boldsymbol{\mathcal{H}}$, to be built up from simple functions as described in the paper you retain control over important state relationships and have an awareness for the controller's possible strengths and weaknesses, while allowing a learning or parameter adaptation law pick the values of the gains through the robot's experience. A policy network may be able to adaptively change which heuristics are active in certain situations and would be worth exploring. The generality of the controller makes it possible to explore a variety of techniques without any major changes to the core robot control architecture.

Acknowledgments

This work was supported by the Toyota Research Institute, Centers for ME Research and Education at MIT and SUSTech, Naver Labs, and the Air Force Office of Scientific Research. Video link: https://youtu.be/7_c82NYmk4Q.

REFERENCES

[1] M. Focchi, R. Orsolino, M. Camurri, V. Barasuol, C. Mastalli, D. G. Caldwell, and C. Semini, "Heuristic planning for rough terrain locomotion in presence of external disturbances and variable perception quality," *CoRR*, vol. abs/1805.10238, 2018.

[2] S. Kuindersma, R. Deits, M. Fallon, A. Valenzuela, H. Dai, F. Permenter, T. Koolen, P. Marion, and R. Tedrake, "Optimization-based locomotion planning, estimation, and control design for the atlas humanoid robot," *Autonomous Robots*, vol. 40, pp. 429–455, Mar 2016.

[3] F. Farshidian, M. Neunert, A. W. Winkler, G. Rey, and J. Buchli, "An efficient optimal planning and control framework for quadrupedal locomotion," in *ICRA*, 2017.

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

[5] C. D. Bellicoso, F. Jenelten, C. Gehring, and M. Hutter, "Dynamic locomotion through online nonlinear motion optimization for quadrupedal robots," no. 99, 2018-01-17.

[6] M. J. Powell, E. A. Cousineau, and A. D. Ames, "Model predictive control of underactuated bipedal robotic walking," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pp. 5121–5126, May 2015.

[7] Y. Tassa, T. Erez, and E. Todorov, "Synthesis and stabilization of complex behaviors through online trajectory optimization," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 4906–4913, Oct 2012.

[8] M. Naveau, M. Kudruss, O. Stasse, C. Kirches, K. Mombaur, and P. Soures, "A reactive walking pattern generator based on nonlinear model predictive control," *IEEE Robotics and Automation Letters*, vol. 2, pp. 10–17, Jan 2017.

[9] G. Bledt, M. J. Powell, B. Katz, J. D. Carlo, P. M. Wensing, and S. Kim, "MIT cheetah 3: Design and control of a robust, dynamic quadruped robot," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, (Madrid, Spain), Oct. 2018.

[10] J. D. Carlo, P. M. Wensing, B. Katz, G. Bledt, and S. Kim, "Dynamic locomotion in the MIT cheetah 3 through convex model-predictive control," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, (Madrid, Spain), Oct. 2018.

[11] G. Bledt, P. M. Wensing, and S. Kim, "Policy-regularized model predictive control to stabilize diverse quadrupedal gaits for the MIT cheetah," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, (Vancouver, Canada), Sept. 2017.

[12] G. Bledt and S. Kim, "Implementing regularized predictive control for simultaneous real-time footstep and ground reaction force optimization," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, (Macau, China), Nov. 2019.

[13] R. Ringrose, "Automatically tuning control systems for simulated legged robots," in *AAAI 1994*, 1994.

[14] S. Chernova and M. Veloso, "An evolutionary approach to gait learning for four-legged robots," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, vol. 3, pp. 2562–2567 vol.3, Sep. 2004.

[15] J. D. Weingarten, G. A. D. Lopes, M. Buehler, R. E. Groff, and D. E. Koditschek, "Automated gait adaptation for legged robots," in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004*, vol. 3, pp. 2153–2158 Vol.3, April 2004.

[16] 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.

[17] N. Heess, D. TB, S. Sriram, J. Lemmon, J. Merel, G. Wayne, Y. Tassa, T. Erez, Z. Wang, S. M. A. Eslami, M. A. Riedmiller, and D. Silver, "Emergence of locomotion behaviours in rich environments," *CoRR*, vol. abs/1707.02286, 2017.

[18] G. Bledt, P. M. Wensing, S. Ingersoll, and S. Kim, "Contact model fusion for event-based locomotion in unstructured terrains," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, (Brisbane, Australia), Accepted 2018.

[19] M. Diehl, H. Bock, H. Diedam, and P.-B. Wieber, "Fast direct multiple shooting algorithms for optimal robot control," in *Fast Motions in Biomechanics and Robotics* (M. Diehl and K. Mombaur, eds.), vol. 340 of *Lecture Notes in Control and Information Sciences*, pp. 65–93, Springer Berlin Heidelberg, 2006.

[20] H. W. Park, S. Park, and S. Kim, "Variable-speed quadrupedal bounding using impulse planning: Untethered high-speed 3d running of MIT cheetah 2," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5163–5170, May 2015.

[21] J. Pratt, J. Carff, S. Drakunov, and A. Goswami, "Capture point: A step toward humanoid push recovery," in *IEEE-RAS Int. Conf. on Humanoid Robots*, (Genova, Italy), pp. 200–207, Dec. 2006.

[22] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Math. Program.*, vol. 106, pp. 25–57, Mar. 2006.

[23] "Extracting legged locomotion heuristics with regularized predictive control," https://youtu.be/7_c82NYmk4Q, https://www.youtube.com/user/MITbiomimetics.