Available online at www.sciencedirect.com

ScienceDirect

IFAC Papers Online CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 58-6 (2024) 42–47

Learning Cost Functions for Reinforced Learned Controllers in a Quadrupedal Robot ⋆

Gabriel Torre * **Omayra Yago Nieto** **
Alexandre Anahory Simoes *** **Juan I. Giribet** ****
Leonardo J. Colombo †

* *G. Torre is with at LINAR Laboratory, Universidad de San Andrés and Universidad de Buenos Aires, Buenos Aires, Argentina. email: torreg@udesa.edu.ar*

** *O.Yago Nieto is with Universidad Politécnica de Madrid (UPM), José Gutiérrez Abascal, 2, 28006 Madrid, Spain. email: omayra.yago.nieto@alumnos.upm.es*

*** *A.Anahory Simoes is with the School of Science and Technology, IE University, Spain. email: alexandre.anahory@ie.edu*

**** *J. Giribet is with CONICET and Universidad de San Andrés, Buenos Aires, Argentina. email: jgiribet@conicet.gov.ar*

† *L.Colombo is Centre for Automation and Robotics (CSIC-UPM), Centre for Automation and Robotics (CSIC-UPM), Ctra. M300 Campo Real, Km 0,200, Arganda del Rey - 28500 Madrid, Spain. email: leonardo.colombo@csic.es*

Abstract: In this work, we will consider a reinforced learning controller developed for a quadrupedal robot and we learn for which cost function such a controller is an optimal control. In particular, we will transform the learning problem into a quadratic programming problem and solve it to obtain the learned cost function. Our approach is based on second-order Lagrangian mechanics since we will use that an optimal control problem is equivalent to a second-order variational problem. We also obtain error bounds for the approximation of the cost function.

Keywords: Optimal Control, Learning-based Control, Quadratic Programming, Quadrupedal Robots, Inverse Reinforcement Learning, Higher-order Lagrangian Mechanics.

1. INTRODUCTION AND PROBLEM FORMULATION

The use of Reinforcement Learning (RL) to learn control policies has been an active research effort during the last years in the robotics and control community. Despite its prevalence, RL generally does not yield the optimal trajectory, even in scenarios involving Linear Quadratic Regulator (LQR) functionals.

This research delves into an optimization challenge linked to control policies learned through RL. Specifically, we assume the existence of a control policy acquired via RL for a specific control task and aim to discern the cost function that this learned controller optimizes. In other words, we seek to identify the cost function for which this data-driven controller is optimal. Through the approximation of the cost function using a polynomial basis, we present a methodology to ascertain the cost minimized by the RL controller by formulating and solving a quadratic programming problem.

The study of locomotion for quadrupedal robots has a long and rich history to outstand their dynamics, their stability, and build agile machines without a formal analysis, thanks to the multi-support nature of such systems. Some famous quadrupedal robots which have appeared in the literature include, but are not limited to, the Boston Dynamics' robots BigDog [Raibert et al. (2008)] and SpotMini [BostonDynamics (2016)], Minitaur [De and Koditschek (2018)], ANYmal [Hutter et al. (2016)] and Cheetah robot [Boussema et al. (2019)]. State-of-the-art approaches for the control and planning of quadrupedal robots mainly utilize model reduction to partly mitigate the computational complexity of the full-order techniques arising from the nonlinearity and hybrid nature of models. There has been also a recent effort to translate techniques on virtual constraints and hybrid zero dynamics for bipedal robots [Westervelt et al. (2018)] to quadrupedal robots to generate periodic gaits in their motion [Ma et al. (2019), Hamed et al. (2020), Fawcett et al. (2021)]. While they

⋆ O. Yago Nieto and G. Torre have contributed equally and both must be considered first authors of the work. J. Giribet was partially supported by NVIDIA Applied Research Program Award 2021, PICT-2019-2371 and PICT-2019-0373 projects from Agencia Nacional de Investigaciones Científicas y Tecnológicas, and UBACyT-0421BA project from the Universidad de Buenos Aires (UBA), Argentina. The authors acknowledge financial support from Grant PID2022-137909NB-C21 funded by MCIN/AEI/ 10.13039/501100011033 and the LINC Global project from CSIC "Wildlife Monitoring Bots" INCGL20022.

10.1016/j.ifacol.2024.08.254

do have many implementation advantages, one needs to design controllers that overcome the uncertainty induced by the difference between modeling and reality. In that sense, the use of RL controllers has shown to be an efficient way to learn controllers to mitigate such uncertainties and perform complex tasks by quadrupedal robots in their motion as climbing stairs or being coordinated with others to transport a payload for instance [Peng et al. (2020), Lee et al. (2020), Castañeda et al. (2020), Ji et al. (2021), Westenbroek et al. (2022)].

Currently, researchers at the LINAR Laboratory in Buenos Aires, Argentina, are actively engaged in the development of quadrupedal robots. One notable creation among these robots is the *Dyna-0* (see Fig. 1 [Torre et al. (2022)]). Employing advanced Reinforcement Learning (RL) algorithms, this robot demonstrates remarkable capabilities in navigating irregular terrains, showcasing exceptional mobility in unstructured environments. The RL-based control algorithm empowers the quadruped robot to precisely follow a predefined trajectory. In a specific scenario detailed in [Torre et al. (2022)], the authors successfully achieved a stable trot—characterized by the alternating movement of diagonally opposite leg pairs—that proved resistant to terrain disturbances and moderately fast. Impressively, this accomplishment relied solely on information derived from an Inertial Measurement Unit (IMU) sensor, underscoring the robot's robustness in dynamic environments.

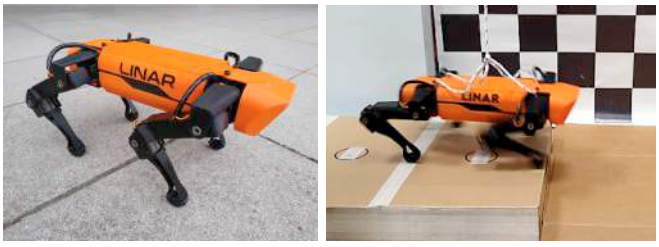

Fig. 1. Left: Dyna-0 in rest position. Right: Experimental setup, variable height step that allows the difficulty of tests to be varied in a simple way.

In particular, Dyna-0 can perceive the information of the environment around it for the creation and location of maps using laser sensors and cameras. Therefore, this quadrupedal robot can select suitable support points while walking and plan its navigation route autonomously. The controller developed in [Torre et al. (2022)], and used in this paper, is for trot, since this produces the most robust results at medium speeds, in contrast to walks that depend on static balance and are limited to very slow speeds.

In this work, we will consider the RL controller developed in [Torre et al. (2022)] for Dyna-0 and we will learn for which cost function such a controller is an optimal control. In particular, we will transform the learning problem into a quadratic programming problem and solve it to obtain the learned cost function. Our approach is based on higher-order Lagrangian mechanics since we will use that an optimal control problem is equivalent to a second-order variational problem [Bloch (2003)].

Utilizing this strategy enables the employment of positions, velocities, jerk, and fourth-order derivatives as data points. These parameters can be measured with the robot in contrast to momenta if we use the Pontryagin maximum principle for necessary conditions of the optimal control problem.

The methodology to transform the data-driven problem into a quadratic programming problem was given in [Ahmadi et al. (2018)] and further extended in [Rodríguez et al. (2020)] to the setting of multi-agent systems.

The remainder of the paper is structured as follows. In Section 2 we review how necessary conditions for optimality can be obtained by solving a second-order variational problem. In Section 3 we solve the problem of learning cost functions as a quadratic programming problem and in Section 4 we study error bounds for the approximation of the cost functions. Finally in Section 5 we present a case-study with the proposed method with the Dyna-0 robot.

2. OPTIMAL CONTROL PROBLEM AS A SECOND-ORDER VARIATIONAL PROBLEM

2.1 Lagrangian mechanics

Let Q be a differentiable manifold with dimension n. Throughout the text, q^i will denote a particular choice of local coordinates on this manifold and TQ denotes its tangent bundle, with T_qQ denoting the tangent space at a specific point $q \in Q$ generated by the coordinate vectors $\frac{\partial}{\partial q^i}$. Usually v_q denotes a vector at T_qQ and, in addition, the coordinate chart q^i induces a natural coordinate chart on TQ denoted by $(q^i, \dot{q}^i)$. T_qQ has a vector space structure, so we may consider its dual space, T_q^*Q and define the cotangent bundle as $T^*Q := \bigcup_{q\in Q} T_q^*Q$, with local coordinates (q^i, p_i). The second-order tangent bundle $T^{(2)}Q$ of a smooth manifold Q consists of the equivalent classes of curves on Q that agree up to their acceleration. Locally, it is described by positions, velocities, and accelerations, that is, $(q^i, \dot{q}^i, \ddot{q}^i) \in T^{(2)}Q$.

Given a Lagrangian function $L : TQ \to \mathbb{R}$, the (fully-actuated) controlled Euler-Lagrange equations describing the dynamics of standard mechanical control systems are given by

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}^i}\right) - \frac{\partial L}{\partial q^i} = u_i, \tag{1}$$

where u_i denotes the control inputs, and where we are using Einstein summation notation, where the sum over repeated indexes is understood.

A Lagrangian L is said to be *regular* if $\det \mathcal{M} \neq 0$, where $\mathcal{M} = (\mathcal{M}_{ij}) := \left(\frac{\partial^2 L}{\partial \dot{q}^i \partial \dot{q}^j}\right)$ for all i, j with $1 \leq i, j \leq n$. If L is regular, the Euler-Lagrange equations induce a vector field $X_L : TQ \to T(TQ)$ describing the dynamics of the Lagrangian system, given by

$$X_L(q^i, \dot{q}^i) = \left(q^i, \dot{q}^i; \dot{q}^i, \mathcal{M}_{ij}^{-1}\left(\frac{\partial L}{\partial q^i} - \frac{\partial^2 L}{\partial \dot{q}^i \partial q^j}\dot{q}^j\right)\right).$$

2.2 Simple hybrid systems

Roughly speaking, the term *hybrid system* refers to a dynamical system which exhibits both continuous and dis-

crete time behaviours. In the literature, one finds slightly different definitions of hybrid system depending on the specific class of applications of interest. For simplicity, and following Johnson (1994) and Westervelt et al. (2018), we will restrict ourselves to the so-called simple hybrid mechanical systems in Lagrangian form.

Simple hybrid systems (see Johnson (1994) and Westervelt et al. (2018)) are characterized by the 4-tuple $\mathcal{H} = (D, X, S, \Delta)$, where D is a smooth manifold called the *domain*, X is a smooth *vector field* on D, S is an embedded submanifold of D with co-dimension 1 called the *switching surface*, and $\Delta : S \to D$ is a smooth embedding called the *impact map*. S and Δ are also referred to as the *guard* and the *reset map*, respectively.

The dynamics associated with a simple hybrid system is described by an autonomous system with impulse effects as in Westervelt et al. (2018).

We denote by $\Sigma_{\mathcal{H}_L}$ the *simple hybrid dynamical control system* generated by a Lagrangian L, that is,

$$\Sigma_{\mathcal{H}_L} : \begin{cases} \dot{\gamma}(t) = X_L(\gamma(t), u(t)), & \text{if } \gamma^-(t) \notin S, \\ \gamma^+(t) = \Delta(\gamma^-(t)), & \text{if } \gamma^-(t) \in S, \end{cases} \tag{2}$$

where $\gamma : I \subset \mathbb{R} \to TQ$, and γ^-, γ^+ denote the states immediately before and after the times when integral curves of the Lagrangian vector field X_L associated with (1) intersect S (i.e., pre and post impact of the solution $\gamma(t)$ with S), namely $\gamma^-(t) := \lim_{\tau \to t^-} \gamma(\tau)$, $\gamma^+(t) := \lim_{\tau \to t^+} \gamma(\tau)$ are the left and right limits of the state trajectory $\gamma(t)$. Note that here we are using as domain D the tangent bundle to Q.

2.3 Optimal control problem

We consider the *optimal control problem*,

$$\min_{u(\cdot)} \int_0^T C(q, \dot{q}, u) dt,$$

subject to

- the simple hybrid dynamical control system (2) ,
- $q(0) = q_0, q(T) = q_T$,

Necessary conditions for optimality in the optimal control problem are determined by Pontryagin's Maximum Principle (PMP) for a Hamiltonian function in T^*Q, parametrized by a control $u \in \mathbb{R}^n$. Alternatively, necessary conditions for optimality can be obtained as solution curves for a second-order variational problem as we explain in the following (see Bloch (2003) Chapter 7 for details).

Note that by replacing the controls in the cost function C by equation (1), the cost functional may then be recast as the second-order Lagrangian function $\mathcal{L} : T^{(2)}Q \to \mathbb{R}$,

$$\mathcal{J} := \int_0^T C\Big(q, \dot{q}, \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}}\right) - \frac{\partial L}{\partial q}\Big)\, dt = \int_0^T \mathcal{L}(q, \dot{q}, \ddot{q})\, dt.$$

Using calculus of variation, the necessary equations for a trajectory to be optimal are given by the second-order Euler-Lagrange equations for the second-order Lagrangian function (the cost function) $\mathcal{L} : T^{(2)}Q \to \mathbb{R}$,

$$\frac{d^2}{dt^2}\left(\frac{\partial \mathcal{L}}{\partial \ddot{q}^i}\right) - \frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{q}^i}\right) + \frac{\partial \mathcal{L}}{\partial q^i} = 0\,.$$

Note that these equations are a set of n fourth-order ordinary differential equations.

3. LEARNING COST FUNCTIONS VIA CONVEX OPTIMIZATION

Let

$$\{t_k, q(t_k), \dot{q}(t_k), \ddot{q}(t_k),\ \dddot{q}(t_k), q^{(iv)}(t_k), \Delta(q(t_l), \dot{q}(t_l))\}_{k=1}^N$$

be the set of sampling states, where $t_l \in \{t_k\}_{k=1}^N$ denotes the impact times within the set of sample times. The strategy to solve the problem is to approximate the Lagrangian function $\mathcal{L}$ by imposing that the second-order Euler-Lagrange equations must be satisfied at the sample data. These values allow us to find the optimal approximation of the Lagrangian function within the chosen polynomial basis by solving a quadratic programming problem. From now on, when we say find or learn the cost function we are refering to the second-order Lagrangian function $\mathcal{L}$.

Consider an d-dimensional space $\mathbb{M}^d(Q, \mathbb{R}) \subset C^2(Q, \mathbb{R})$ with a finite basis of functions $\{\phi_p\}_{p=1}^d$. As previously, we wish to find a cost function $\mathcal{L}(q, \dot{q}, \ddot{q})$ such that $(q, \dot{q}, \ddot{q})$ minimizes $\int_0^T \mathcal{L}(q, \dot{q}, \ddot{q}) dt$. To solve our problem, we will assume that the cost function can be written as a linear combination of the basis of polynomials as

$$\widehat{\mathcal{L}}(q, \dot{q}, \ddot{q}) = \sum_{i=1}^d \alpha_i \phi_i(q, \dot{q}, \ddot{q}) \tag{3}$$

for $\widehat{\mathcal{L}} : T^{(2)}Q \to \mathbb{R}$, and it has to satisfy the following second-order Euler-Lagrange equations at the sample points

$$\frac{d^2}{dt^2}\left(\frac{\partial \widehat{\mathcal{L}}}{\partial \ddot{q}}\right) - \frac{d}{dt}\left(\frac{\partial \widehat{\mathcal{L}}}{\partial \dot{q}}\right) + \frac{\partial \widehat{\mathcal{L}}}{\partial q} = 0 \tag{4}$$

where

$$\begin{aligned} \frac{d^2}{dt^2}\left(\frac{\partial \widehat{\mathcal{L}}}{\partial \ddot{q}}\right)\Big|_{t_k} &= \frac{d^2}{dt^2}\left(\sum_{i=1}^d \alpha_i \frac{\partial \phi_i}{\partial \ddot{q}}(q, \dot{q}, \ddot{q})\right)\Big|_{t_k} \\ &= \sum_{i=1}^d \alpha_i \frac{d^2}{dt^2}\left(\frac{\partial \phi_i(q, \dot{q}, \ddot{q})}{\partial \ddot{q}}\right)\Big|_{t_k}, \\ \frac{d}{dt}\left(\frac{\partial \widehat{\mathcal{L}}}{\partial \dot{q}}\right)\Big|_{t_k} &= \frac{d}{dt}\left(\sum_{i=1}^d \alpha_i \frac{\partial \phi_i}{\partial \dot{q}}(q, \dot{q}, \ddot{q})\right)\Big|_{t_k} \\ &= \sum_{i=1}^d \alpha_i \frac{d}{dt}\left(\frac{\partial \phi_i}{\partial \dot{q}}(q, \dot{q}, \ddot{q})\right)\Big|_{t_k}, \\ \frac{\partial \widehat{\mathcal{L}}}{\partial q}\Big|_{t_k} &= \sum_{i=1}^d \alpha_i \frac{\partial \phi_i}{\partial q}(q, \dot{q}, \ddot{q})\Big|_{t_k}. \end{aligned}$$

For each $j = 1, \ldots, n$, we define the following cost function

$$J_j(\alpha,t) := \sum_{i=1}^{d}\alpha_i \sum_{k=1}^{n}\Bigg(\frac{\partial^3\phi_i(q,\dot q,\ddot q)}{\partial\ddot q_j\partial^2\ddot q_k}q_k^{(iv)} + \frac{\partial^3\phi_i(q,\dot q,\ddot q)}{\partial\ddot q_j\partial\ddot q_k\partial\dot q_\ell}\ddot q_\ell\dddot q_k + \frac{\partial^3\phi_i(q,\dot q,\ddot q)}{\partial\ddot q_j\partial\ddot q_k\partial q_\ell}\dot q_\ell\dddot q_k + \frac{\partial^2\phi_i(q,\dot q,\ddot q)}{\partial\ddot q_j\partial\ddot q_k}q_k^{(iv)} + \frac{\partial^3\phi_i(q,\dot q,\ddot q)}{\partial\ddot q_j\partial\ddot q_\ell\partial\dot q_k}\dddot q_\ell\ddot q_k + \frac{\partial^3\phi_i(q,\dot q,\ddot q)}{\partial\ddot q_j\partial^2\dot q_k}\dddot q_k + \frac{\partial^3\phi_i(q,\dot q,\ddot q)}{\partial\ddot q_j\partial\dot q_\ell\partial q_k}\dot q_k\ddot q_\ell + \frac{\partial^2\phi_i(q,\dot q,\ddot q)}{\partial\ddot q_j\partial\dot q_k}\dddot q_k + \frac{\partial^3\phi_i(q,\dot q,\ddot q)}{\partial\ddot q_j\partial\ddot q_\ell\partial q_k}\dddot q_\ell\dot q_k + \frac{\partial^3\phi_i(q,\dot q,\ddot q)}{\partial\ddot q_j\partial\dot q_\ell\partial q_k}\ddot q_\ell\dot q_k + \frac{\partial^3\phi_i(q,\dot q,\ddot q)}{\partial\ddot q_j\partial^2 q_k}\ddot q_k + \frac{\partial^2\phi_i(q,\dot q,\ddot q)}{\partial\ddot q_j\partial q_k}\ddot q_k\Bigg) - \sum_{i=1}^{d}\alpha_i\sum_{k=1}^{n}\Bigg(\frac{\partial^2\phi_i(q,\dot q,\ddot q)}{\partial\dot q_j\partial\ddot q_k}\dddot q_k + \frac{\partial^2\phi_i(q,\dot q,\ddot q)}{\partial\dot q_j\partial q_k}\dot q_k\Bigg) + \sum_{i=1}^{d}\alpha_i\frac{\partial\phi_i(q,\dot q,\ddot q)}{\partial q_j}.$$

To ease the notation, consider the time-dependent $n\times n$ matrices A, B and C such that

$$J_j(\alpha,t) = \sum_{i=1}^{d}\alpha_i\left(A_{ij}(t) - B_{ij}(t) + C_{ij}(t)\right),$$

corresponding to the three terms multiplying α_i in the coordinate expression of $J_j(\alpha,t)$.

Finally, we will consider the final aggregate objective function given by

$$J(\alpha) = \frac{1}{N}\sum_{k=1}^{N} J^T(\alpha,t_k)J(\alpha,t_k), \tag{5}$$

where $J(\alpha,t_k) = (J_1(\alpha,t_k),\dots,J_n(\alpha,t_k))$, thus,

$$J(\alpha,t_k) = (A(t_k) - B(t_k) + C(t_k))\alpha.$$

Note that J is a convex function since it is quadratic in α.

The objective function (5) is a quadratic programming of the form

$$J(\alpha) = \alpha^T Q\alpha$$

where Q is the symmetric matrix

$$Q = \frac{1}{N}\sum_{k=1}^{N}(A(t_k)-B(t_k)+C(t_k))^T(A(t_k)-B(t_k)+C(t_k)),$$

since

$$J^T(\alpha,t_k)J(\alpha,t_k) = \alpha^T(A(t_k)-B(t_k)+C(t_k))^T (A(t_k)-B(t_k)+C(t_k))\alpha.$$

Remark 1. Note that the matrix Q is symmetric. If it is also positive definite, then minimize the functional J over α is equivalent to solve the Quadratic Programming problem (5). ◇

As the domain is convex, it is also compact, so we can apply local-global minimum fundamental theorem, which affirms that, the local minimum of our convex function $J(\alpha)$ is also a global minimum if and only if the domain is convex. Also, by checking the Karush-Kuhn-Tucker conditions we also obtain the local minimum which also is global.

For positive definite Q, the ellipsoid method can be used to solve the quadratic program. This can be compared to the nonlinear optimization formulation solved by the Nelder-Mead method, which can take an enormous amount of iterations with negligible improvement in function value.

4. ERROR BOUNDS AND DEGREE SELECTION

Consider the Euler-Lagrange operator $EL^{(2)} : C^\infty(T^{(2)}Q) \to T^*Q$ given by

$$EL^{(2)}[\mathcal{L}] = \frac{d^2}{dt^2}\left(\frac{\partial\mathcal{L}}{\partial\ddot q}\right) - \frac{d}{dt}\left(\frac{\partial\mathcal{L}}{\partial\dot q}\right) + \frac{\partial\mathcal{L}}{\partial q}.$$

For our problem, denote the unknown real cost functional by $\mathcal{L}$. The actual cost functional might be written as the sum of the learned cost functional plus an error term $\Delta\mathcal{L}\in C^\infty(T^2Q)$, i.e.,

$$\mathcal{L} = \widehat{\mathcal{L}} + \Delta\mathcal{L}.$$

Applying the second-order Euler-Lagrange operator to both sides of the equation, we obtain

$$EL^{(2)}[\mathcal{L}] = EL^{(2)}[\widehat{\mathcal{L}}] + EL^{(2)}[\Delta\mathcal{L}].$$

Now, over the trajectories of the system, $EL^{(2)}[\mathcal{L}]$ vanishes and if we denote by $\epsilon_{\mathcal{L}}$ the term $-EL^{(2)}[\Delta\mathcal{L}]$, we get the equation

$$EL^{(2)}[\widehat{\mathcal{L}}] = \epsilon_{\mathcal{L}}$$

on points belonging to an optimal trajectory.

The error between the actual cost functional and the estimated cost function from the data can be seen as an external force acting on the approximate system. We will use this term to make a bound on the error term.

Proposition 2. Let $\gamma^* := \min_\alpha J(\alpha)$. Then

$$\|\epsilon_{\mathcal{L}}(q_i,\dot q_i,\ddot q_i,\dddot q_i,q_i^{(iv)})\| \le \sqrt{N\gamma^*}$$

over the trajectory $\{q_i,\dot q_i,\ddot q_i,\dddot q_i,q_i^{(iv)}\}$, $i=1,\dots,N$.

Proof. First notice that by definition

$$J(\alpha) = \frac{1}{N}\sum_{i=1}^{N}\|EL^{(2)}[\widehat{\mathcal{L}}](q_i,\dot q_i,\ddot q_i,\dddot q_i,q_i^{(iv)})\|^2.$$

Therefore, for each index $i=1,\dots,N$, we have that

$$\frac{1}{N}\|\epsilon_{\mathcal{L}}(q_i,\dot q_i,\ddot q_i,\dddot q_i,q_i^{(iv)})\|^2 \leqslant \gamma^*$$

from where the proposition follows.

Remark 3. The last proposition guarantees that for low values of γ^* the optimal solutions associated with the learned cost function $\widehat{\mathcal{L}}$ remain close to the trajectories associated with the cost function $\mathcal{L}$.

5. CASE STUDIED - QUADRUPEDAL SIMULATION

We have applied our results to the learning of the cost function for a reinforcement learned controller trained on a physical simulation of the previously mentioned *Dyna-0.*

The simulation was created in the *Bullet3 Physics Engine* [Coumans and Bai (2016–2021)], and in order to reduce the state space dimensions, the robot is constrained to the XZ plane turning the control problem to 2D in the physical domain, said simulation can be seen in Figure 5. It should be noted that the primary focus of this simulation is to accurately replicate the physical dynamics of each limb, specifically simulating the corresponding tensor of inertia. The "pill" shaped representations of the limbs serve solely as a visual aid and do not influence the core simulation mechanics. Contact interactions are exclusively confined to the robot's feet which are accurately simulated.

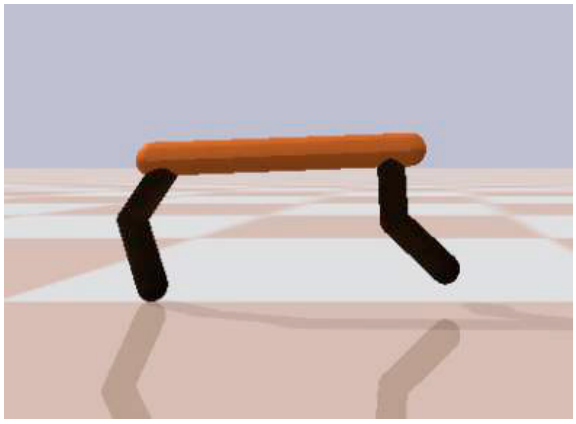

Fig. 2. Simplified simulation constrained to the X-Z plane and combining left and right limbs.

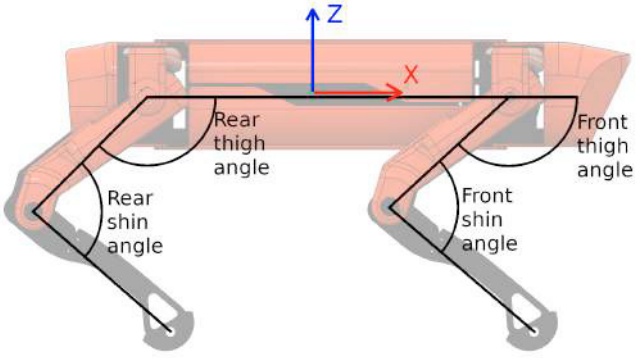

Fig. 3. State dimension conformed by each mechanical degree of freedom in the legs and its derivative (noted speed) and, Z, Z speed, X speed and pitch(XZ)

The state is given by the angle and angular speed of every joint, the speed of the robot base and the pitch and height of the base, the full list of state dimension can be seen in Table 1. The actions of the policy are the control torque on each of the four actuators (rear and front, shin and thigh) The control policy is a parameterized with a 2 layer, 64 neuron ReLU neural network, trained with the *Soft Actor Critic algorithm* [Haarnoja et al. (2018)], with the objective of maximizing forward velocity whilst not using too much power, the complete cost function can be seen in Equation (6). It should be noted that the modeled actuators have limited torque thus the policy learns how to deal with this limitation and still optimize for the cost function.

$$\begin{aligned}
\text{alive bonus} &= \begin{cases} +1 & \text{If the robot's pitch is within} \\ & \text{limits and thigh and body are} \\ & \text{not in contact with the ground} \\ -1 & \text{Otherwise} \end{cases} \\
\text{progress} &= \text{X speed} \\
\text{electricity} &= 2.0 \times \text{mean}(|u| \times \text{joint_speeds}) \\
\text{stall torque} &= 0.1 \times \text{mean}(u^2) \\
\text{joints at limit} &= 0.1 \times \text{joints_at_limits} \\
\text{cost function} &= \text{electricity} + \text{stall torque} + \text{joints at limit} \\
&\quad - \text{alive bonus} - \text{progress}
\end{aligned} \tag{6}$$

The best polynomial from a base, comprising of polynomials up to degree 7 and order 4, was fitted for each state dimension by solving via quadratic optimization Equation 5 with 20000 time steps of forward gait (see Table 1), these states are related to dimension seen in Figure 5

State Dim.	State Name	Polynom.	γ^*
S_1	Front thigh angle	T6xu	5.5997e-06
S_2	Front thigh ang. speed	T4xu	2.0068e-03
S_3	Front shin angle	T6xu	6.0996e-05
S_4	Front shin ang. speed	T4xu	8.4353e-03
S_5	Rear thigh angle	Gxu	1.1263e-03
S_6	Rear thigh ang. speed	T4xu	5.9601e-03
S_7	Rear shin angle	T6xu	1.2961e-04
S_8	Rear shin ang. speed	T4xu	9.7281e-03
S_9	Z	T5xu	2.7530e-09
S_{10}	Speed X	T5xu	4.1645e-06
S_{11}	Speed Z	G7	1.3488e-07
S_{12}	Pitch angle	G7	3.4838e-08

Table 1. States with best polynomial and associated cost, see table 3 for polynomial detail and table 2 for associated alpha coefficients

Most polynomials related to actuators in Tables 2 and 3 have bigger alphas in the quadratic terms, which is consistent with the penalization to the energy in the cost function used to train the RL controller. The forward speed, S_{10}, has bigger alphas in the derivative terms α_4 y α_5 that can be explained by optimizing for a smooth forward speed which comes in hand with smooth gaits.

6. CONCLUSIONS

In conclusion, this study introduces a new approach to identifying optimal cost functions for reinforced learning controllers in quadrupedal robots. Through the transformation of the learning problem into a quadratic programming challenge and the application of second-order Lagrangian mechanics, our research successfully derived cost functions that elucidate the learned gait of the Dyna-0 robot within a simulated 2D environment.

As part of our future work, we aspire to broaden the scope of our methodology by extending it to underactuated systems. Furthermore, we aim to implement this approach in refining learned gaits for the Dyna-0 robot within a simulated 3D environment. This expansion will not only contribute to a more comprehensive understanding of the proposed methodology but also enable its application to more complex robotic systems. This perspective offers a promising application in designing and optimizing control strategies for complex robotic systems, enhancing the relevance and applicability of our contributions in the field of robotic engineering.

REFERENCES

Ahmadi, M., Topcu, U., and Rowley, C. (2018). Control-oriented learning of lagrangian and hamiltonian systems. In *2018 Annual American Control Conference (ACC)*, 520–525. IEEE.

Bloch, A. (2003). *Nonholonomic Mechanics and Control*. Interdisciplinary Applied Mathematics Series, 24. Springer-Verlag.

BostonDynamics (2016). Legged robot spotmini `https://youtu.be/tf7IEVTDjng`.

State Dim.	Polynomial	α_1	α_2	α_3	α_4	α_5	α_6	α_7	α_8	α_9	α_{10}
S_1	T6xu	1.37e-04	1.44e-04	7.51e-06	9.45e-01	3.74e-02	1.22e-03	2.17e-02			
S_2	T4xu	-5.77e-05	-1.84e-02	1.91e-03	1.21e-01	8.96e-01					
S_3	T6xu	-2.47e-04	-8.59e-04	-1.87e-03	7.45e-01	6.34e-02	-7.00e-03	2.06e-01			
S_4	T4xu	6.70e-05	-2.27e-02	5.23e-04	1.26e-01	8.96e-01					
S_5	Gxu	2.29e-01	1.04e-01	-1.19e-01	0.0	-1.12	-3.07e-02	0.0	5.23e-01	0.0	1.42
S_6	T4xu	-1.13e-04	-2.40e-02	3.31e-04	1.20e-01	9.04e-01					
S_7	T6xu	-8.94e-04	-2.68e-03	-8.27e-03	1.60e-01	2.01e-02	-4.91e-02	8.83e-01			
S_8	T4xu	4.62e-05	-1.39e-02	-3.56e-04	1.09e-01	9.05e-01					
S_9	T5xu	3.36e-05	-5.08e-08	-3.12e-05	2.59	3.46e-02	0.0	5.36e-03			
S_{10}	T5xu	4.77e-05	1.11e-05	-3.11e-04	9.81e-01	2.12e-02	0.0	5.95e-03			
S_{11}	G7	2.71e-04	-9.31e-04	-2.51e-02	1.07e-03	3.24e-01	5.81e-01	2.85e-01	2.48e-05		
S_{12}	G7	-2.38e-05	3.13e-03	4.76e-03	-2.54e-01	3.73e-02	3.57	5.11	-1.66e-03		

Table 2. Alphas corresponding to Polynomials in Table 3 for states in Table 1

G7	$\alpha_1 q + \alpha_2 q^2 + \alpha_3 q^3 + \alpha_4 q^4 + \alpha_5 q^5 + \alpha_6 q^6 + \alpha_7 q^7 + \alpha_8 \ddot{q}^2$
Gxu	$\alpha_1 q + \alpha_2 q^2 + \alpha_3 q^3 + \alpha_4 \dot{q} + \alpha_5 \dot{q}^2 + \alpha_6 \ddot{q}^3 + \alpha_7 q\dot{q} + \alpha_8 q\dot{q}^2 + \alpha_9 q^2\dot{q} + \alpha_{10}\ddot{q}^2$
T4xu	$8\alpha_1 q^4 - 8\alpha_2 q^2 + 8\alpha_3 \dot{q}^4 - 8\alpha_4 \dot{q}^2 + 2 + \alpha_5 \ddot{q}^2$
T5xu	$16\alpha_1 q^5 - 20\alpha_2 q^3 + 5\alpha_3 q + 16\alpha_4 \dot{q}^5 - 20\alpha_5 \dot{q}^3 + 5\alpha_6 \dot{q} + \alpha_7 \ddot{q}^2$
T6xu	$32\alpha_1 q^6 - 48\alpha_2 q^4 + 18\alpha_3 q^2 + 32\alpha_4 \dot{q}^6 - 48\alpha_5 \dot{q}^4 + 18\alpha_6 \dot{q}^2 - 2 + \alpha_7 \ddot{q}^2$

Table 3. Selected polinomials.

Boussema, M., M. J. Powell, and, G.B., Ijspeert, A.J., Wensing, P.M., and Kim., S. (2019). Online gait transitions and disturbance recovery for legged robots via the feasible impulse set. *EEE Robotics and Automation Letters, 4(2):1611–1618.*

Castañeda, F., Wulfman, M., Agrawal, A., Westenbroek, T., Sastry, S., Tomlin, C., and Sreenath, K. (2020). Improving input-output linearizing controllers for bipedal robots via reinforcement learning. In *Learning for Dynamics and Control*, 990–999. PMLR.

Coumans, E. and Bai, Y. (2016–2021). Pybullet, a python module for physics simulation for games, robotics and machine learning. `http://pybullet.org`.

De, A. and Koditschek, D.E. (2018). Vertical hopper compositions for pre-flexive and feedback-stabilized quadrupedal bounding, pacing, pronk-ing, and trotting. *The International Journal of Robotics Research, 37(7):743–778.*

Fawcett, R.T., Pandala, A., Ames, A.D., and Hamed, K.A. (2021). Robust stabilization of periodic gaits for quadrupedal locomotion via qp-based virtual constraint controllers. *IEEE Control Systems Letters*, 6, 1736–1741.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. (2018). Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor.

Hamed, K.A., Kamidi, V.R., Pandala, A., Ma, W.L., and Ames, A.D. (2020). Distributed feedback controllers for stable cooperative locomotion of quadrupedal robots: A virtual constraint approach. In *2020 American Control Conference (ACC)*, 5314–5321. IEEE.

Hutter, M., Gehring, C., Jud, D., Lauber, A., Bellicoso, C.D., Tsounis, V., Hwangbo, J., Bodie, K., Fankhauser, P., Bloesch, M., Diethelm, R., Bachmann, S., Melzer, A., and Hoepflinger, M. (2016). Anymal - a highly mobile and dynamic quadrupedal robot. *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 38-44.*

Ji, Y., Zhang, B., and Sreenath, K. (2021). Reinforcement learning for collaborative quadrupedal manipulation of a payload over challenging terrain. In *2021 IEEE 17th International Conference on Automation Science and Engineering (CASE)*, 899–904. doi: 10.1109/CASE49439.2021.9551481.

Johnson, S.D. (1994). Simple hybrid systems. *Int. J. Bifurcation Chaos*, 04(06), 1655–1665. doi: 10.1142/S021812749400126X.

Lee, J., Hwangbo, J., Wellhausen, L., Koltun, V., and Hutter, M. (2020). Learning quadrupedal locomotion over challenging terrain. *Science Robotics, vol. 5, no. 47.*

Ma, W.L., Hamed, K.A., and Ames, A.D. (2019). First steps towards full model based motion planning and control of quadrupeds: A hybrid zero dynamics approach. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 5498–5503. IEEE.

Peng, X.B., Coumans, E., Zhang, T., Lee, T.W.E., Tan, J., and Levine, S. (2020). Learning agile robotic locomotion skills by imitating animals. *Robotics: Science and Systems, 07.*

Raibert, M., Blankespoor, K., Nelson, G., and Playter., R. (2008). Bigdog, the rough-terrain quadruped robot. *IFAC Proceedings Volumes, 41(2):10822–10825. 17th IFAC World Congress.*

Rodríguez, V.A., Fernández, M.G., Moreno, P., and Colombo, L. (2020). A data-driven method based on quadratic programming for distance-based formation control of euler–lagrange systems. *IEEE Control Systems Letters*, 5(1), 313–318.

Torre, G., Grillo, A., and Bunge, R. (2022). Dyna-0: Cuadrúpedo para aprendizaje automático. In *Jornadas Argentinas de Robótica, Bariloche, Río Negro, Argentina.*

Westenbroek, T., Castañeda, F., Agrawal, A., Sastry, S., and Sreenath, K. (2022). Lyapunov design for robust and efficient robotic reinforcement learning. In *6th Annual Conference on Robot Learning.*

Westervelt, E.R., Grizzle, J.W., Chevallereau, C., Choi, J.H., and Morris, B. (2018). *Feedback Control of Dynamic Bipedal Robot Locomotion.* CRC Press.