# Energy Efficient Pronking of a Series Elastic Actuated Quadrupedal Robot Using Trajectory Optimization and Functional Iterative Learning Control

Josh Yeung Ho Pho

TU Delft

Research Centre for Robotics
Università di Pisa

In Supremæ Dignitatis
1343

# Energy Efficient Pronking of a Series Elastic Actuated Quadrupedal Robot Using Trajectory Optimization and Functional Iterative Learning Control

by

Josh Yeung Ho Pho

to obtain the degree of Master of Science in Robotics
at the Delft University of Technology,
to be defended publicly on Thursday 14 November 2024 at 10:30 AM.

| | | |
|---|---|---|
| Student Number: | 4541197 | |
| Project Duration: | August, 2023 – November, 2024 | |
| Faculty: | Mechanical Engineering (ME) | |
| Department: | Cognitive Robotics (CoR) | |
| Daily Supervisors: | Asst. Prof. C. Della Santina | TU Delft |
| | Asst. Prof. F. Angelini | UniPi |
| | Assoc. Prof. M. Garabini | UniPi |
| Thesis Committee: | Asst. Prof. C. Della Santina | TU Delft |
| | Assoc. Prof. J. Kober | TU Delft |
| | Asst. Prof. F. Angelini | UniPi |
| | Assoc. Prof. M. Garabini | UniPi |

Cover image photographed by F. Iotti.

TU Delft

Research Center E. Piaggio
University of Pisa

Università di Pisa

# Preface

This master's thesis concludes a chapter of eight years of my life as a student, during which I had the opportunity to spend the last two years abroad. In this period I have learnt what it means to keep pushing my boundaries outside of my comfort zone. Living in three different countries across two continents has been a challenge and a rollercoaster, to say the least, but I feel fortunate to have met incredible people from all over the world. Having the chance to live in these diverse countries has allowed me to understand their cultures on a deeper level and has shaped me into the person I am today.

I want to thank Dr. Cosimo Della Santina for supervising my thesis and giving me the opportunity to collaborate with the Research Center E. Piaggio in Pisa, Italy. A huge thank you to Dr. Franco Angelini and Dr. Manolo Garabini for guiding me through the thesis, always providing valuable feedback and helping me solve problems that I encountered along the way. I also want to thank all the PhD students from the Natural Intelligence project, who were always open to answering my robotics-related questions. Furthermore, I specifically want to thank PhD students Pietro Gori and Francesco Iotti, who helped me a lot with the experiments in the final weeks. At last, I want to thank Maarten Hugenholtz for proofreading my thesis and giving me valuable feedback.

I could not have completed this thesis without the continuous support of my family and friends. I am deeply thankful to my parents and sister for their constant encouragement, particularly during the difficult times. I am grateful to all my friends who have made great efforts to visit me, wherever I was in the world. And last but not least, I want to thank all my Italian amici, whose warmth and hospitality made my time in Italy unforgettable—from climbing mountains in the Dolomites to visiting the Trevi Fountain in Rome. Grazie ragazzi, per tutto e a dopo!

*Josh Yeung Ho Pho*
*Delft, November 2024*

# Energy Efficient Pronking of a Series Elastic Actuated Quadrupedal Robot Using Trajectory Optimization and Functional Iterative Learning Control

Josh Yeung Ho Pho

***Abstract*—Monitoring expeditions in endangered habitats are currently performed by human experts. However, this approach has several disadvantages, including the limited amount of experts, cost-intensive expeditions, and the dangers that are posed by exploring dangerous terrains. Therefore, one can look into using quadrupedal legged robots that would collaborate with the human operators, which would be able to assist them by performing extra measurements in these dangerous habitats. An open problem in quadrupedal legged locomotion is robust periodic forward jumping, a.k.a. pronking, specifically for quadrupedal robots that have flexible joints placed in series. In this paper, we therefore propose a novel framework that generates an energy efficient pronking motion for a quadrupedal series elastic actuated legged robot. This periodic pronking motion is generated using a reduced order model and the serial elasticity of the joints is taken into account using an template-anchor approach. To minimize the trajectory error we use Functional Iterative Learning Control (fILC) as feedforward control in parallel with a proportional-derivative feedback controller. The advantage of using fILC is that the elasticity of the quadrupedal robot is preserved and that the controller is able to learn the pronking motion in a small number of iterations. This framework is validated on an eight degrees of freedom series elastic actuated robot. In- and outdoor experiments show that this framework is able to work in unknown terrains.**

## I. Introduction

DUE to the heavy usage of fossil fuels and agriculture since the late 18th century, the amount of carbon dioxide and other greenhouse gases have increased monotonically [1]. The consequence of this is a change in climate on Earth, which leads to increase of endanger of ecosystems and species, water stress, coastal flooding, mortality rates due to heat stress and droughts [2]. The European Union therefore proposes the European Green Deal to mitigate this issue, which contains a proposed strategy to tackle the aforementioned climate change problem[1]. One of the elements of the Green Deal involves preservation and restoration of ecosystems protected by the Natura 2000 Network (N2000N). Here, the goal is also to increase the biodiversity of existing species[2]. The N2000N terrestrial habitats are currently monitored by human operators, as they have the expertise to perform an examination of the habitat and they are able to explore unknown environments. A downside of using human operators is that there is only a limited amount of experts in the field, and with the increasing amount of endangered habitats there is a need for more human operators. Such a monitoring expedition is also cost-intensive and might be dangerous for older experts with lower mobility in areas such as Alpine screes [3]. One can look into using robots to aid human operators in performing experiments. The robot can for example perform extra measurements in addition to the human operators on an expedition, or in case of extreme weather/dangerous areas take over the task of the human operator and autonomously perform measurements. Angelini *et al.* [3] propose to use soft legged quadrupedal robots to aid human operators in habitat monitoring. Although aerial vehicles are more agile and are not limited to the surface of earth, its main drawback is the limited amount of battery charge. The term 'soft' in legged quadrupeds indicates the addition of elasticity in the joints. To be able to anticipate unpredictable behaviour in robots, researchers have been inspired by the body structures of animals [4][5]. Adding elastic elements that are biologically inspired can provide benefits such as (1) regulation of muscle power, by storing energy and releasing it in a quick manner for purposes such as jumping tasks. (2) Recycling energy, by storing the elastic strain energy in tendons and returning it as elastic recoil [6]. (3) Reducing the impact forces on the foot at touchdown [7]. Examples of these soft legged quadrupeds are Spacebok [8], ANYmal [9], Delft E-Go [10] and MULINEX [11].

The planned movement and execution of a quadrupedal robot is separated into two phases, a planner and controller phase, as illustrated in Fig. 1. In the planning phase the motion of the robot is generated and optimized using Trajectory Optimization (TO). To achieve an energy efficient motion using TO we can minimize for a certain cost function. This cost function can for example be the amount of energy spent to produce the trajectory and is often expressed in metrics such as Cost of Transport (CoT) [12][13] or torque squared [14]. Using the CoT as cost function for example means that the trajectory of the joints can be optimized for using as less energy as possible. Another aspect to be considered in TO is the model and motion of the robot. For quadrupedal robots, the motion of an energy efficient periodic forward jump, called 'pronking' is still a challenge [15]. To make this problem more tractable, it is often chosen to opt for a reduced order model. However, many articles consider the quadrupedal robot as a robot with rigid links, therefore not taking the

[1]https://ec.europa.eu/environment/nature/natura2000/index_en.htm

[2]https://eur-lex.europa.eu/legal-content/EN/TXT/?qid=1576150542719&uri=COM%3A2019%3A640%3AFIN

elasticity in the joints into account [16][17]. Examples of reduced order models that do take them into account are the Spring Loaded Inverted Pendulum (SLIP) [18], 3-D SLIP [19] and Trunk-Spring Loaded Inverted Pendulum (T-SLIP) [12]. Ding *et al.* [15] propose a compliant single-mass model which takes the elasticity of the joints into account using a template-anchor approach [20]. In practice this means that they use the T-SLIP model to generate a pronking motion that also takes the Parallel Elastic Actuators (PEA) into account. With this method, they capture the periodic jumping motion with a simple T-SLIP model whilst still taking the parallel elasticity in the joints into account. To the best of the author's knowledge, no similar paper has been found that combines the T-SLIP model and modelling of Series Elastic Actuators (SEA) in a quadrupedal robot.

In the controller phase a simple Proportional-Derivative (PD) feedback controller is often used to track the reference trajectory [21][22]. However, Della Santina *et al.* [23] mention that closed-loop control is equivalent to increasing the stiffness of the system, defeating the purpose of adding elasticity in the joints. Instead, it is suggested to use open loop feedforward control in combination with low PD feedback control. Iterative Learning Control (ILC) is an intelligent feedforward control method that is able to learn to reduce the signal reference error in a short amount of iterations and send feedforward signals to the robot, given that the task is periodic [24]. Gori *et al.* [17] use ILC to close the sim-to-real gap for a quadruped jumping task. Pierallini *et al.* and Chhatoi *et al.* show successful trajectory tracking using ILC in compliant underactuated arms [25][26][27]. A special type called functional Iterative Learning Control (fILC) works for underactuated (non-square) systems. With fILC, intermittent control is possible and the generated feedforward signal is continuous of nature [28]. Drost *et al.* show trajectory tracking in a one-link flexible arm using fILC [29]. Ding *et al.* extend this work by applying fILC on a PEA quadruped robot for a pronking task [15]. To the best of the author's knowledge, there is no paper found that has provided experimental validation of fILC on a SEA quadrupedal robot.

In summary, the contribution of this thesis work is threefold:

1) An energy efficient periodic pronking motion is achieved in a TO framework, which also takes the elasticity of the joints that are configured in series into account. The trajectory of the robot in Cartesian space is modeled using the T-SLIP model [12]. Using a template-anchor approach the serial elasticity of the joints are embedded into the T-SLIP model.
2) To reduce the sim-to-real gap of the performance of the quadrupedal robot we use fILC. fILC is an intelligent control method that works for underactuated systems and is able to learn to reduce the trajectory error in a short number of iterations. We verify and validate in simulation and real experiments that fILC works on a SEA quadrupedal robot.
3) These two methods, combined in one framework, are validated on an 8 Degrees of Freedom (DoF) SEA driven prototype robot through both indoor and outdoor experiments. The TO is calculated offline, while fILC can perform all its calculations online.

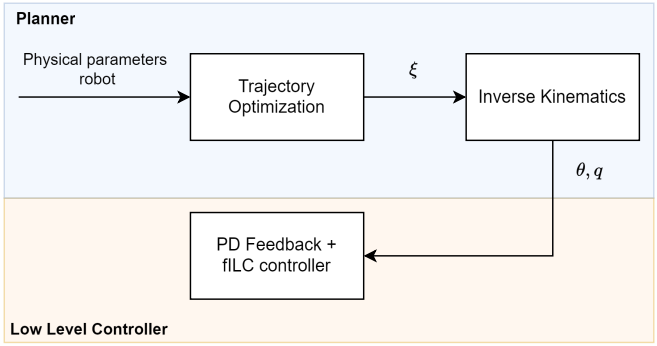



Fig. 1. Pipeline of the whole framework.

The structure of this paper is split up into two main blocks, as illustrated in Fig. 1. The first block, as elaborated on in Section II, is the trajectory optimization of the motion of the robot. Here, the T-SLIP model in combination with SEA model are used to generate the trajectory. With the physical parameters of the robot, the trajectory can be optimized to minimize the energy used to perform the pronking motion. The output of this block are the Cartesian coordinates, and with inverse kinematics the motor and link trajectories in Joint coordinate frame can be obtained. To be able to follow the reference trajectories, PD feedback in parallel with fILC feedforward control are used, which is explained in more detail in Section III. The simulation and experimental results of the method are explained in Section IV and V, respectively. At last, a conclusion and recommendations for future work are given in Section VI.

## II. Trajectory Generation and Optimization

This section will focus on the formulation of the trajectory optimization framework. To achieve an energy efficient pronking gait the problem can be formulated as an Optimal Control Problem (OCP). The OCP can be converted into a Non-Linear Problem (NLP) [30]. We follow a bottom-up approach in which each component of the NLP is explained in more detail in the following subsections.

### A. Objective Function (14)

Cost of Transport (CoT) is often used as a metric to compare the efficiency between robots, independent of their weight and size. CoT is a dimensionless number and can be defined as the work required to move an unit body an unit distance [12]. Specifically, Mechanical Cost of Transport ($CoT_{mech}$) is defined as the mechanical work $E$ produced by the actuators divided by the mass of the robot $m$ multiplied by a travel distance $x_N$ [13]. This is given by

$$CoT_{mech} = \frac{E}{mgx_N} = \frac{\sum_{n=1}^{N}\sum_{i=2}^{5}\tau_i^n\dot{\theta}_i^n}{mgx_N}. \quad (1)$$

Here $\tau$ is the torque delivered by the actuator and $\dot{\theta}$ angular velocity of the actuator.

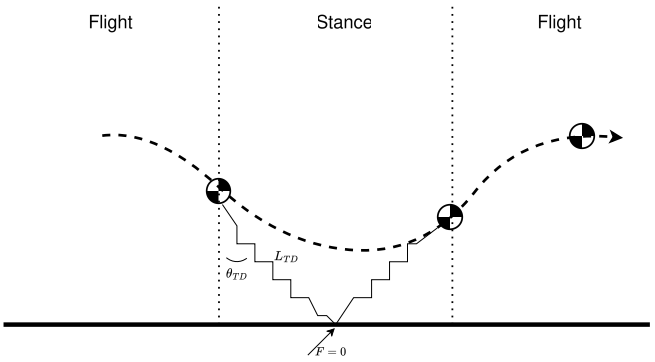



Fig. 2. Illustration of the T-SLIP model, with the TD event in between the flight-stance phase and TO event in the stance-flight phase.

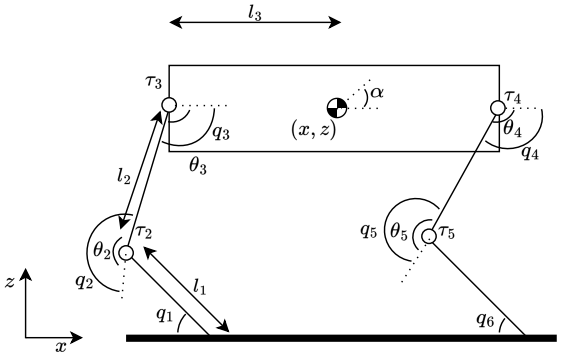



Fig. 3. View of the quadruped robot with its joint angles and torque inputs from the motors.

### *B. Robot Model* (15)-(17)

The trajectory of the robot is modelled using the Trunk-Spring Loaded Inverted Pendulum (T-SLIP) [31], which is an extension of the SLIP model [18] by adding an extra DoF in the Center of Mass (CoM). The T-SLIP model is a cyclic gait that consists of three phases, as illustrated by Fig. 2. In these three phases it is assumed that the robot is a Single Rigid Body (SRB), i.e. all mass is concentrated in the center of the robot. The SRB consists of three Cartesian coordinates, as shown by Fig. 3, which are the horizontal $x$ and vertical $z$ position, respectively and the trunk angle $\alpha$. These three Cartesian coordinates are combined in $\xi = [x, z, \alpha]^T$. The quadrupedal robot has four motor side angles $\theta_2, \ldots, \theta_5$, with four actuated joints $\tau_2, \ldots, \tau_5$ and six link side angles $q_1, \ldots, q_6$, all in Joint space.

The first phase, called the flight phase, starts at the apex after which the CoM follows a ballistic trajectory according to

$$M(\xi)\ddot{\xi} + G(\xi) = 0, \tag{2}$$

where $M$ is the inertia matrix $\in \mathbb{R}^{3\times3}$ and $G \in \mathbb{R}^{3\times1}$ the gravitational term. After the first flight phase the T-SLIP model comes in contact with the ground, which is called the stance phase. The transition from the first flight phase to the stance phase is called the Touch-Down (TD) event. The TD event, as given by Fig. 2, happens at a TD angle $\theta_{TD}$ and uncompressed spring length $l_{TD}$, after which the virtual spring gets loaded in compression. The virtual spring force is elongated and a transition of stance to flight phase occurs, which is called the Take-Off (TO) event. After the TO event the robot goes into a second flight phase until it reaches the apex again. The T-SLIP model is given in Cartesian coordinates and is connected with the links of the robot by means of inverse kinematics. The SEA of the robot are modeled according to the reduced Spong model [32]. In Spong's model, the elasticity in the joints can be modeled by separating the motor $\theta \in \mathbb{R}^{n_j}$ and link $q \in \mathbb{R}^{n_j}$ side dynamics, with $n_j$ being the number of joints. In between each motor and link there is a spring with stiffness $K$. Spong's model considers the following assumptions [20][33]:

(A1) Deflection of the joints are small, therefore the elasticity of the joints can assumed to be linear.
(A2) Center of the mass of each motor is on the rotation axis.
(A3) Each motor is located on the previous link.
(A4) The kinetic energy of the motor is due to its own rotation, i.e. independent of the rotation of the link.

The equations of motion can be derived using the previous assumptions and the Euler-Lagrangian formulation, as explained in Appendix A. This results in the link and motor side equations described by

$$\begin{aligned} M(\xi)\ddot{\xi} + G(\xi) + J_h^T(\xi)K(q-\theta) &= 0, \\ B\ddot{\theta} + K(\theta - q) &= \tau_m, \end{aligned} \tag{3}$$

where $B \in \mathbb{R}^{n_j \times n_j}$ is the motor inertia matrix, $K \in \mathbb{R}^{n_j \times n_j}$ the stiffness matrix and $\tau_m \in \mathbb{R}^{n_j}$ the torque generated by the motor. The flight and stance phases are combined and solved for the acceleration terms as given by

$$\begin{aligned} &\ddot{\xi} = -M(\xi)^{-1}G(\xi), \\ &\ddot{\xi} = M(\xi)^{-1}(-G(\xi) + J_h^T(\xi)K(\theta - q)) \quad \text{(Cartesian space)}, \\ &\ddot{\theta} = B^{-1}(\tau_m - K(\theta - q)) \qquad \text{(Joint space)}. \end{aligned} \tag{4}$$

### *C. System Dynamics* (18)

The state of the system is chosen to be

$$\zeta = [\xi, \dot{\xi}, \theta, \dot{\theta}]^T = [x, z, \alpha, \dot{x}, \dot{z}, \dot{\alpha}, \theta_2, \ldots, \theta_5, \dot{\theta}_2, \ldots, \dot{\theta}_5]^T \in \mathbb{R}^{14} \tag{5}$$

and the control input to the robot

$$u = [\tau_2, \ldots, \tau_5]^T \in \mathbb{R}^4. \tag{6}$$

The system dynamics can be formulated as a first order differential equation $\dot{\zeta} = f(x, u)$ and is solved using Direct Multiple Shooting (DMS) method. The state in the next timestep $\zeta_{n+1}$ can be calculated using Runge-Kutta 4 method.

### *D. Initial Condition* (19)

According to the T-SLIP model the robot starts in flight phase at the apex. Hence, the initial horizontal $x$ and vertical velocity $\dot{z}$ at the first timestance are equal to 0, as given by

$$[x, \dot{z}]^T = [0, 0]^T. \tag{7}$$

TABLE I
STATE AND CONTROL INPUT LIMITS OF THE ROBOT.

| State $\zeta$ Limits | Value | Input $u$ Limits | Value |
|---|---|---|---|
| $x \quad [m]$ | 0, 2 | $\tau_2 \quad [Nm]$ | -15, 15 |
| $z \quad [m]$ | 0, 2 | $\tau_3 \quad [Nm]$ | -15, 15 |
| $\alpha \quad [deg]$ | -20, 20 | $\tau_4 \quad [Nm]$ | -15, 15 |
| $\dot{x} \quad [m/s]$ | 0, 5 | $\tau_5 \quad [Nm]$ | -15, 15 |
| $\dot{z} \quad [m/s]$ | -2, 2 | | |
| $\dot{\alpha} \quad [m/s]$ | -2, 2 | | |
| $dt \quad [s]$ | 0.001, 0.1 | | |
| $\theta_{TD} [deg]$ | 0, 40 | | |
| $\theta_2 \quad [deg]$ | -180, 180 | | |
| $\theta_3 \quad [deg]$ | -180, 180 | | |
| $\theta_4 \quad [deg]$ | -180, 180 | | |
| $\theta_5 \quad [deg]$ | -180, 180 | | |
| $\dot{\theta}_2 \quad [deg/s]$ | -180, 180 | | |
| $\dot{\theta}_3 \quad [deg/s]$ | -180, 180 | | |
| $\dot{\theta}_4 \quad [deg/s]$ | -180, 180 | | |
| $\dot{\theta}_5 \quad [deg/s]$ | -180, 180 | | |

### E. Periodicity Constraint (20)

The T-SLIP model assumes that the trajectory is a cyclic motion, therefore the state in the last node should be equal to the zeroth state in the next iteration, therefore

$$\zeta_0 = \zeta_N. \tag{8}$$

### F. Event Based Switch Constraints (21)-(22)

The event based switch constraints consists of two events; the TD and TO events as mentioned before. The TD event occurs in the first flight phase, so $\dot{z} < 0$. Furthermore, it is assumed that this event occurs at the moment the virtual spring touches the ground, i.e.

$$z = l_{TD} cos(\theta_{TD}). \tag{9}$$

The TO event occurs when the robot is moving in upwards direction ($\dot{z} > 0$) and Ground Reaction Force (GRF) $F$ of the virtual spring equals zero, such that

$$F = K(l - l_{virt}) = 0. \tag{10}$$

The length of the spring $l$ can be calculated as

$$l^2 = \sqrt{x^2 + z^2}, \tag{11}$$

so substitution of (11) in (10) results in the TO event as given by

$$\sqrt{x^2 + z^2} - l_{virt} = 0. \tag{12}$$

### G. State & Input Limits (23)-(24)

The state and control input limits of the robot are given in Tab. I. The state limits are estimated from [15] and the limits of the control input are obtained from the EM-Act actuator datasheet [11].

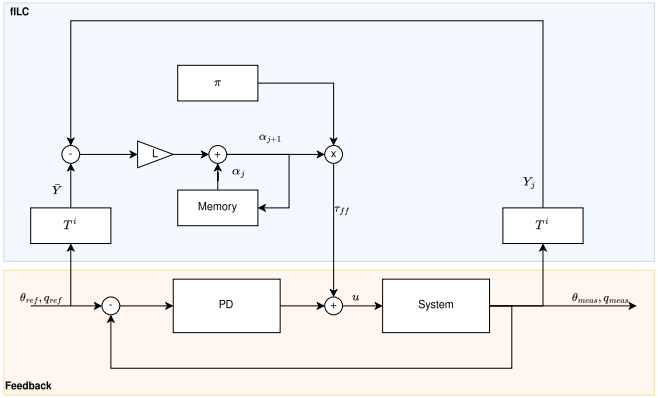



Fig. 4. Overview of the low level control framework.

### H. Friction Constraint (25)

To minimise the slippage of the feet when it is in contact with the ground a Coulomb friction constraint is added, as given by

$$F_x = \mu F_z, \tag{13}$$

where $F_x$ and $F_z$ are the horizontal and vertical contact forces in x- and z- direction, respectively, and $\mu$ the friction coefficient.

### I. Non-Linear Problem

All equations and constraints ((14)-(25)) that are explained in the previous subsections can be combined into the following TO framework.

$$\underset{\substack{x_0,\dots,x_N,\\ u_0,\dots,u_N,\\ dt,\theta_{TD},l_{TD}}}{\text{minimize}} \quad \text{CoT}_{\text{mech}} = \frac{\sum_{n=1}^{N}\sum_{i=2}^{5}\tau_i^n\dot{\theta}_i^n}{mgx_N} \tag{14}$$

$$\text{subject to} \quad \ddot{\xi} = -M^{-1}G \quad \forall n \in [0, n_{TD}) \cup [n_{TO}, N] \tag{15}$$

$$\ddot{\xi} = M^{-1}(-G + J_h^T K(\theta - q)) \quad \forall n \in [n_{TD}, n_{TO}) \tag{16}$$

$$\ddot{\theta} = B^{-1}(\tau - K(\theta - q)) \quad \forall n \in [n_{TD}, n_{TO}) \tag{17}$$

$$\zeta_{n+1} = \zeta_n + \int_{\zeta_n}^{\zeta_{n+1}} f(\zeta_n, u_n)dt \quad \forall n \in [0, N] \tag{18}$$

$$[\zeta_0 \quad \dot{\zeta_0}]^T = 0 \tag{19}$$

$$\zeta_0 = \zeta_N \tag{20}$$

$$\zeta_{n_{TD}} \in \chi_{TD} = \{x | z - l_{virt}\cos(\theta_{virt}) = 0 \wedge \dot{z} < 0\} \tag{21}$$

$$\zeta_{n_{TO}} \in \chi_{TO} = \{x | \sqrt{x^2 + z^2} - l_{virt} = 0 \wedge \dot{z} > 0\} \tag{22}$$

$$\zeta_{min} \leq \zeta_n \leq \zeta_{max} \tag{23}$$

$$u_{min} \leq u_n \leq u_{max} \tag{24}$$

$$F_x \leq \mu F_z \quad \forall n \in [0, N] \tag{25}$$

## III. Low Level Control Framework

The low level control framework can be broken down into two main components as illustrated in Fig. 4. The idea of this control framework is to combine a low feedback gain with a feedforward term, as using high gain feedback alone increases the mechanical stiffness of the robot, which defeats the purpose of adding compliancy in the robot [23]. Fig. 4 shows that we have a closed $PD_{feedback}$ loop in parallel with feedforward fILC. The following subsections will delve more in depth into the specifics of feedforward fILC.

### A. Functional Iterative Learning Control

In functional Iterative Learning Control (fILC) [28] a linear non-square continuous system

$$\dot{x}_j = Ax_j + Bu_j, y_j = Cx_j, \tag{26}$$

is assumed, with iteration index $j$, $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times l}$, $C \in \mathbb{R}^{m\times n}$, $x_j \in \mathbb{R}^n$, $u_j \in \mathbb{R}^l$ , $y_j \in \mathbb{R}^m$, is considered. For an underactuated system, $l < m$ can be assumed. Furthermore, $o$ number of time instances $T^1, \dots, T^o$ is defined. The goal with fILC is to learn the control input $u_j(t)$, where $j$ implies the iteration index, such that

$$\lim_{j\to\infty} y(T^k) = \bar{y}^k, \forall \in 1 \dots o, \tag{27}$$

where $\bar{y}^k$ denotes the reference output. The control input $u_j(t)$

$$u_j(t) = \pi(t)\alpha_j, \pi = [\pi^1 \dots \pi^o] \in \mathbb{R}^{l\times mo}, \tag{28}$$

is split into two components, where $\pi(t)$ is time dependent and consists of a combination of basis functions $\pi^i(t) \in \mathbb{R}^{l\times m}$ and $\alpha_j \in \mathbb{R}^{l\times mo}$ is the learning vector. The closed form solution of (27) is

$$y_j(t) = Ce^{At}x(0) + C\int_0^t e^{At-\tau}Bu_j(\tau)d\tau. \tag{29}$$

Substitution of (28) into (29) at the *i*-th time instance yields

$$y_j(T^i) = Ce^{AT^i}x(0) + (\int_0^{T^i} Ce^{AT^i-\tau}B\pi(\tau)d\tau)\alpha_j. \tag{30}$$

(29) can subsequently be transformed in the super-vector notation,

$$Y_j = d_j + H\alpha_j, \tag{31}$$

where $Y_j$ denotes all output responses at iteration $j$. $d_j$ and $H$ are the free- and forced response, respectively. The error $E_j$ between the tracked reference output $\bar{Y}$ and measurement $Y_j$ can be multiplied by a learning rate $L \in \mathbb{R}^{mo\times mo}$ to determine the vector $\alpha_{j+1}$ in the next iteration, as given by

$$\alpha_{j+1} = \alpha_j + LE_j = \alpha_j + L(\bar{Y} - Y_j). \tag{32}$$

### B. Basis Function Selection

Della Santina *et al.* [28] found out that, any choice of $\pi$ can be chosen if $det(H) \neq 0$. Therefore, the following three types of basis functions are chosen [15]:

- Gaussian:

$$\pi^i(t) = A_1\frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{1}{2}(\frac{t-\mu^i}{\sigma_1})^2} \tag{33}$$

- Sinusoid:

$$\pi^i(t) = A_2(1 + sin(\omega(t - b^i)) \tag{34}$$

- Radial Gaussian:

$$\pi^i(t) = A_3e^{-\frac{1}{2}(\frac{t-\mu^i}{\sigma_3^2})^2} \tag{35}$$

Here, $\pi^i$ is distributed evenly over the stance phase period, i.e. $[n_{TD}, n_{TO})$. To prevent coupling of control inputs, $\pi^i$ is diagonally inserted into $\pi(t)$.

### C. Quadratically Optimal Learning Rate

An optimal learning rate can be used for (32) if a more accurate knowledge of the plant is known [34]. (36) shows a quadratic next-iteration cost criterion called Q-ILC, where $Q_{LQ} \in \mathbb{R}^{mo\times mo}$ penalizes the error between the reference and measured output and $S_{LQ} \in \mathbb{R}^{mo\times mo}$ penalizes large changes in the control action [24]. The H-matrix can be obtained without deriving an explicit model by exciting the system through $\pi^i$ and recording the forced response at $o$ time instances. Minimizing the cost criterion

$$J_{j+1}(u_{j+1}) = ||E_j||^2_{Q_{LQ}} + ||\alpha_j - \alpha_{j-1}||^2_{S_{LQ}}, \tag{36}$$

results in an optimal learning rate

$$L = (H^TQH + S)^{-1}H^TQ. \tag{37}$$

### D. Continuous fILC Learning

According to the T-SLIP model the Center of Mass (CoM) of the robot starts at apex with an initial horizontal speed. In practice however this is not possible, therefore an alternative approach called continuous fILC learning is used [15]. In this approach fILC starts with $\alpha = 0$ and in stance phase. The trajectory is followed in an iterative manner without resetting to the initial condition after each iteration. In practice this means that at the last state of iteration 1, the system state is not reset to the initial state, but is instead used as the first state of iteration 2.

### E. Control Input to Robot

The structure of control input to the robot is graphically depicted in Fig. 4. Here, a PD feedback law is added to the fILC feedforward torque, as given by

$$u_{j+1}(t) = \underbrace{K_P(\theta_{ref}(t) - \theta_{meas}(t)) + K_D(-\dot{\theta}_{meas}(t))}_{PD_{feedback}} + \underbrace{u_{fILC_{j+1}}}_{\tau_{feedforward}}(t). \tag{38}$$

The PD feedback term in (38) works by multiplying a proportional $K_P \in \mathbb{R}^{n_j\times n_j}$ and derivative gain $K_D \in \mathbb{R}^{n_j\times n_j}$ with the error between the reference and measured motor side angles. The $\alpha_{j+1}$ term in the fILC control law is multiplied with $\pi(t)$ as given by (32). Here, $\bar{Y}$ is chosen to be the position references of both the motor- and link side trajectories, therefore $\bar{Y} = [\theta_2, \dots, \theta_5, q_2, \dots, q_5]$. The variables are the same for $Y_j$, as we can measure both motor- and link side positions with encoders. The feedforward fILC term, in the next iteration, is calculated as given by (28) and added to the PD feedback term, as given by (38).

TABLE II
INPUT FOR THE OPTIMIZATION FRAMEWORK.

| Parameter | Value |
|---|---|
| $m \quad [kg]$ | 5.5 |
| $J \quad [kgm^2]$ | $9.24 * 10^{-3}$ |
| $k_2 \quad [Nm/rad]$ | 45.5*2 |
| $k_3 \quad [Nm/rad]$ | 45.5*2 |
| $k_4 \quad [Nm/rad]$ | 45.5*2 |
| $k_5 \quad [Nm/rad]$ | 45.5*2 |
| $l_1 \quad [m]$ | 0.19 |
| $l_2 \quad [m]$ | 0.19 |
| $l_3 \quad [m]$ | 0.23 |
| $I_2 \quad [kg/m^2]$ | $2.1 * 10^{-3}$ |
| $I_3 \quad [kg/m^2]$ | $2.1 * 10^{-3}$ |
| $I_4 \quad [kg/m^2]$ | $2.1 * 10^{-3}$ |
| $I_5 \quad [kg/m^2]$ | $2.1 * 10^{-3}$ |
| $nodes \quad [-]$ | 300 |

## IV. SIMULATION RESULTS

This section will focus on the results obtained from simulation. First the setup of the simulation and the optimal results from the Trajectory Optimization (TO) are explained. Then, different hyperparameters of functional Iterative Learning Control (fILC) are compared to each other in the the subsequent subsections.

### A. Setup

The trajectory optimization is calculated using CasADi [35] and Ipopt as solver [36]. Furthermore, ROS2 is used as a framework to develop robotic software [37] and Gazebo [38] allows users to simulate their robot in a virtual environment. In the real experiment the prototype robot 'MULINEX' is to be used and a physical description, Unified Robot Description Format (URDF), of MULINEX is used in Gazebo. Using inverse kinematics the Cartesian coordinates are mapped to link side angles, which are used as trajectory reference for the robot. Note that we deliberately use only the link side reference angles in simulation as the URDF of MULINEX has no elasticity, i.e. the body and its links are rigid. The intention is to use both reference motor- and link side trajectories in the real experiments.

### B. Trajectory Optimization Results

In this section we show the results of the Nonlinear Programming (NLP) problem that is explained in Section II-I. Here, we also compare two cases in terms of CoT, one where the robot is assumed to have flexible SEA joints and the other for rigid joints. In the first case we can use the physical parameters of the MULINEX robot from Tab. II. This is the input into the NLP problem that is explained in Section II-I. Using the Ipopt solver we obtain the following results; $CoT = 0.957$, $\theta_{opt} = 16.85°$ and $l_{TD} = 0.3[m]$. Fig. 5(a) shows the Cartesian trajectory that the robot follows according to the results of the NLP. The red arrows denote the Ground Reaction Forces (GRF) and the red and orange points are the flight-stance event switches and virtual foot positions, respectively. Fig. 5(b) depicts the Cartesian trajectory of its position on the left and velocities on the right. For the second rigid joints case we assume a high joint stiffness ($K \rightarrow \infty$). Therefore, $\theta \rightarrow q$ and $K(q - \theta) \rightarrow$ finite value, so that we obtain the equivalent rigid model if we sum up both motor- and link equations in (3) [20]. For the second case we get the following results; $CoT = 1.42$, $\theta_{opt} = 16.85°$ and $l_{TD} = 0.3[m]$. The results of both cases are again highlighted in Tab. III, and from this it can be concluded that with series elastic actuation the quadruped is more energy efficient compared to a rigid quadruped.

TABLE III
COMPARISON OF ENERGY EFFICIENCY FOR A RIGID AND ELASTIC MODEL.

| | Elasticity | Rigid |
|---|---|---|
| $CoT$ | 0.96 | 1.42 |

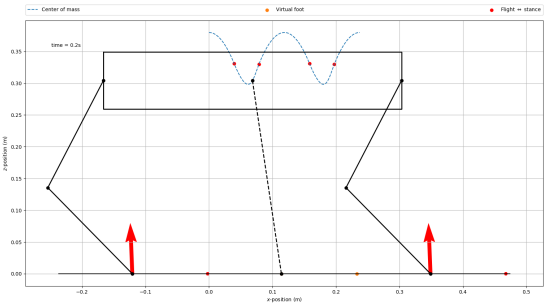
### C. Functional Iterative Learning Control Results

This section focuses on the verification of using fILC on a quadruped with the TO results from the previous section. To show the contribution of fILC more clearly we opt for an objective function that also focuses on reaching a higher jumping height. Therefore, the objective function is changed to

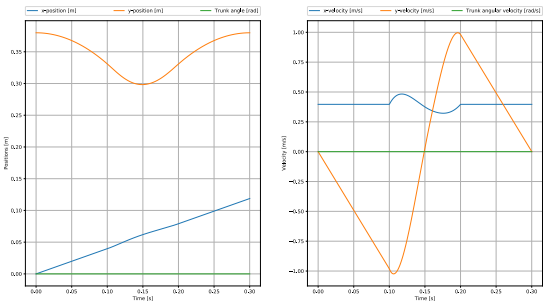
$$objective = CoT + w_1(\zeta[7:11,0]_{final} - \zeta[7:11,0]_{initial}), \tag{39}$$

where $w_1 = 1e4$ is a weight value, $\zeta[7:11,0]_{final}$ is the final values of the motor side angles obtained from TO and $\zeta[7:11,0]_{initial} = [1.64, -2.39, -2.39, 1.64]$ the initial motor side angles that is chosen as input into the TO. Furthermore, in ROS2 and in the real experiments we are not able to measure the base coordinates of the robot, therefore we can not verify and validate the fILC contribution in terms of CoT. Instead, we chose to use the Mean Average Error (MAE) of the positions of each joint to measure the performance of the robot. With these changes in mind, the output of this NLP results in a $CoT_{mech} = 2.86[-]$, $\theta_{TD} = 0[rad]$ and $l_{TD} = 0.3[m]$. Fig. 5(c) and Fig. 5(d) again show the Cartesian trajectory and Cartesian positions and velocities, respectively. These results from the 'planner' are used to test the performance of the fILC controller in ROS2. As mentioned before in Section III-B, the basis functions are distributed evenly over the stance phase, i.e. $t = 0.1 - 0.2[s]$. The values of the basis function are set to a maximum torque value of $\tau = 10[Nm]$, which are used to determine the H-matrix for calculating the optimal learning rate (37). Here, the time instances are chosen to be at the following nodes: $T^i = [0, 50, 100, 125, 150, 175, 199, 200, 250, 300]$. Note that the MAE only measures the tracking performance only at the time instances, not the whole timespan of the trajectory. The frequency of the control loop in ROS2 is set at $f = 500[Hz]$, because this will be the maximum frequency that the MULINEX robot can handle in the experiment. The following sections will compare the hyperparameters of the fILC against one another.

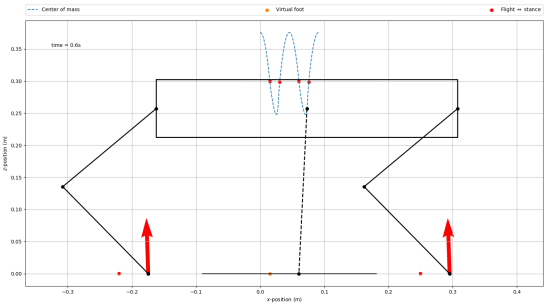
*1) Control Input Comparison:* Two control input strategies are compared, which are using only $PD_{feedback}$ versus using $PD_{feedback}$ and the $\tau_{feedforward}$ term that is generated by



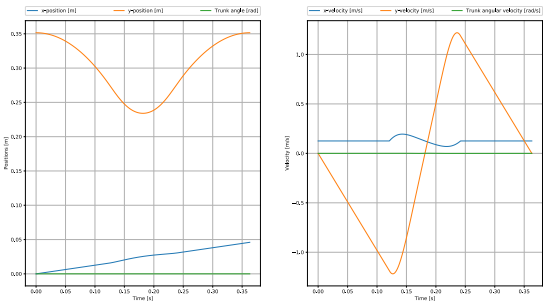
(a) Cartesian trajectory, $objective = CoT$



(b) Cartesian positions and velocities, $objective = CoT$

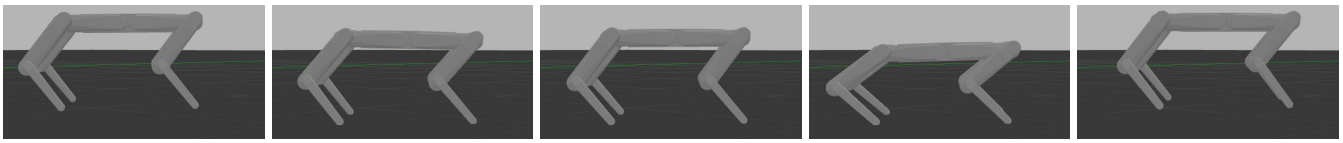


(c) Cartesian trajectory, $objective = CoT + w_1(\zeta[7:11,0]_{final} - \zeta[7:11,0]_{initial})$



(d) Cartesian positions and velocities, $objective = CoT + w_1(\zeta[7:11,0]_{final} - \zeta[7:11,0]_{initial})$

Fig. 5. This figure depicts the Cartesian trajectory obtained from trajectory optimization in (a) and (c). The red arrows here indicate the ground reaction forces. The Cartesian positions and velocities from trajectory optimization are illustrated in (b) and (d).



(a) Flight phase (b) TD event (c) Stance phase (d) TO event (e) Flight phase

Fig. 6. Snapshots of the pronking motion of the robot in the Gazebo simulator.

fILC. Fig.6 shows the motion of the jump in the Gazebo environment. The motion of the robot starts in flight phase Fig.6(a), which occurs from $t = 0.0 - 0.1[s]$. The robot then touches the ground during the TD event Fig.6(b), as described before in Section II-F. After this event the robot goes into stance phase from $t = 0.1 - 0.2[s]$, as shown in Fig.6(c). The robot prepares for take-off during the TO event and goes into flight phase again from $t = 0.2 - 0.3[s]$ as shown in Fig.6(d) and Fig.6(e), respectively. Fig. 7 and Fig. 8 show the results of the comparison between the two control strategies. From simulation, it is observed that with both methods the robot is able to perform a successful pronking motion. Fig. 7 shows the reference trajectory for the actuated joints seen from the right hand side. Note that the black crosses here are the time instances that are chosen in the previous subsection. The first iteration, given in blue, only uses PD feedback control, without feedforward fILC. After 150 iterations it is observed that with fILC the trajectory tracking is more accurate. The lower two images in Fig. 7 show the torque control input and learning weights of $\alpha$, respectively. Fig. 8 shows the comparison of both control input strategies in terms of MAE, which shows that with only $PD_{feedback}$ the same MAE value of $0.048[rad]$ is achieved for all iterations. This is as expected, because with $PD_{feedback}$ only it is not able to 'learn' in iteration domain. The control input that takes the fILC feedforward term into account is able to converge into a lower MAE. The nature of fILC is also present in the first couple iteration, in which the MAE lowers monotonously and converges into a stable value of $0.0158[rad]$.

*2) Basis Function Comparison:* The performance of three types of basis function described before in (33)-(35) are compared to each other. The parameters of the basis function are chosen as follows: $A_1 = 0.25$, $\sigma_1^2 = 0.0001$, $A_2 = 5.0$, $\omega = 2\pi\frac{1}{T_{stance}}$, $A_3 = 10.0$ and $\sigma_3^2 = 0.01$. The $PD_{feedback}$ gains are set to $K_P = 50I_{n_j \times n_j}$ and $K_D = 0.4I_{n_j \times n_j}$. The other fILC parameters are fixed at $Q = 1.0I_{n_j \times n_j}$ and $S = 1.0I_{n_j \times n_j}$, with initial value of $\alpha = 0$. The results are illustrated in Fig. 9 and Fig. 10. Fig. 9 shows the amount

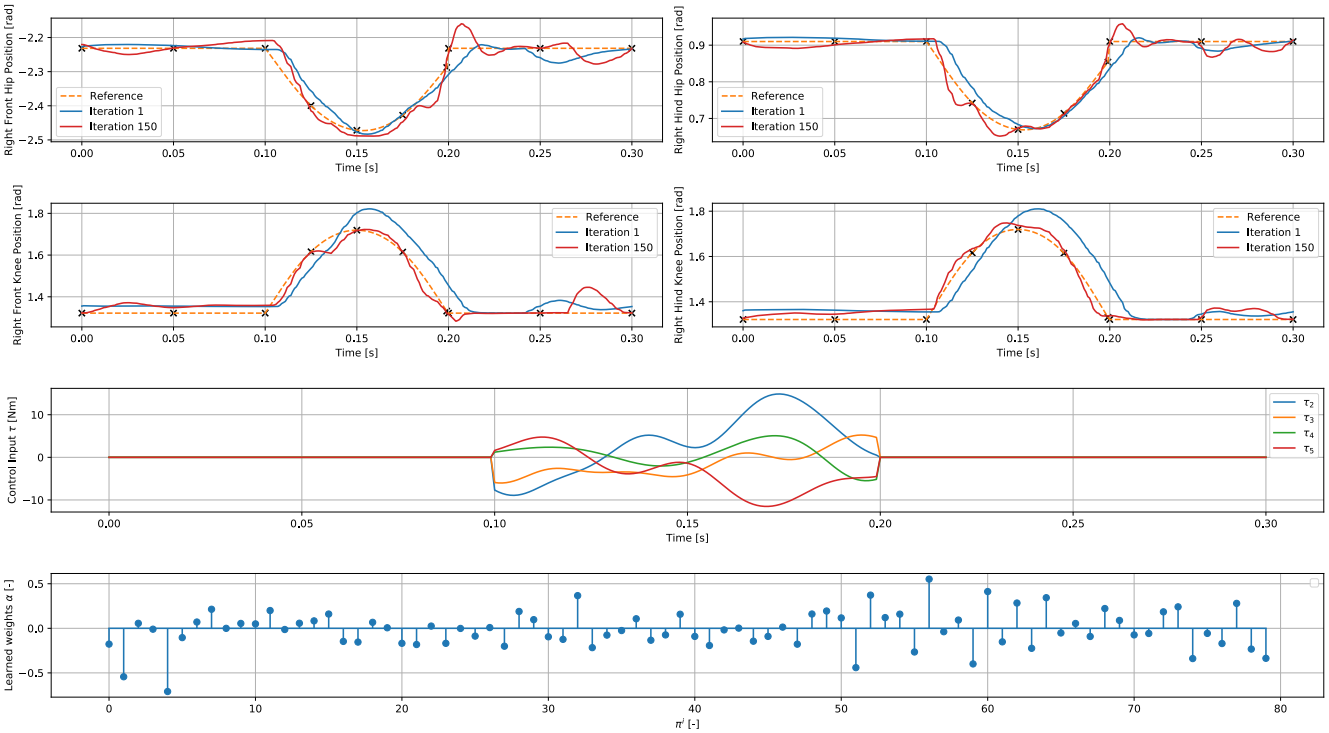



Fig. 7. Comparison of PD feedback and PD feedback + fILC feedforward, in Joint space. The upper four images show the reference tracking performance of the joints seen from the right hand side of the robot. Note that in iteration 1 there is no feedforward term as $\alpha = 0$, therefore iteration 1 is equivalent to using only PD feedback. In orange the reference trajectories that are obtained from TO are given. The black crosses indicate the tracked time instances. The torque control input and the learning weights of $\alpha$ at iteration 150 are shown in the lower two graphs.

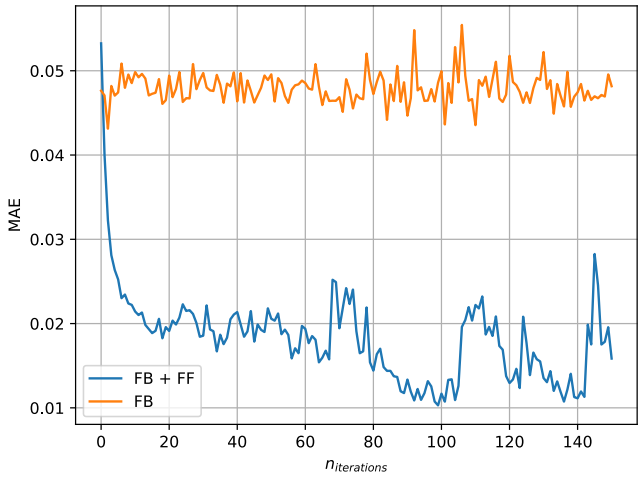



Fig. 8. Comparison of PD feedback and PD feedback + fILC feedforward.

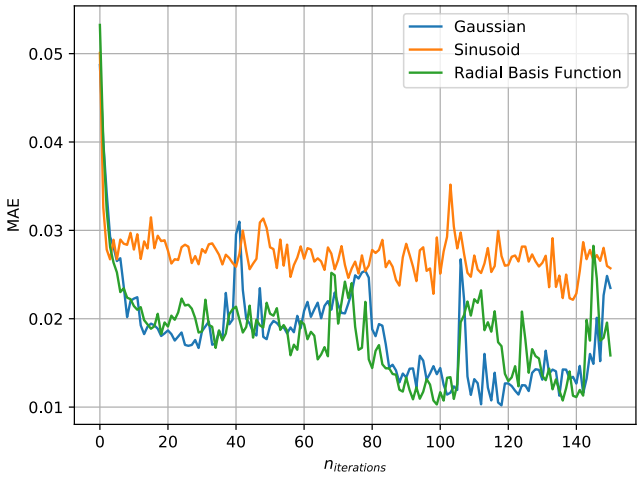
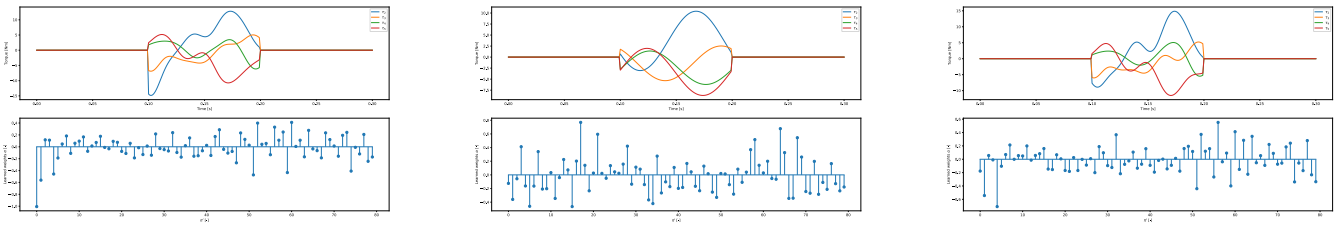



Fig. 9. Comparison between Gaussian, Sinusoid and Radial Gaussian basis functions.

of iterations ($n_{iterations}$) and Mean Average Error (MAE) on the horizontal and vertical axes, respectively. Fig. 10 depicts the fILC feedforward torques of all three basis functions and the distribution of $\alpha$ of all $\pi^i$. From simulation it is obtained that with all three basis functions the robot is able to perform the pronking motion. Fig. 9 shows that the Gaussian and Radial Gaussian basis functions perform better compared to the Sinusoid function in terms of MAE, but also contain larger variation in MAE, which is noticeable at iteration numbers 80 to 140. All three of them show fast convergence in the first couple of iterations and the MAE at $n_{iteration} = 150$ are $0.023$, $0.028$ and $0.015[rad]$ for Gaussian, Sinusoid and Radial Gaussian, respectively.

*3) Learning Rate Comparison:* Comparison of parameters of the learning rate, as described by (37), is done in this section. Bristow *et al.* mention that $Q$ can influence the performance, rate of convergence at robustness of the system



(a) Gaussian (b) Sinusoid (c) Radial Gaussian

Fig. 10. Comparison of Gaussian, Sinusoid and Radial basis functions. The upper images show the torque inputs to the robot and the lower images show the $\alpha$ distribution across all $\pi^i$.

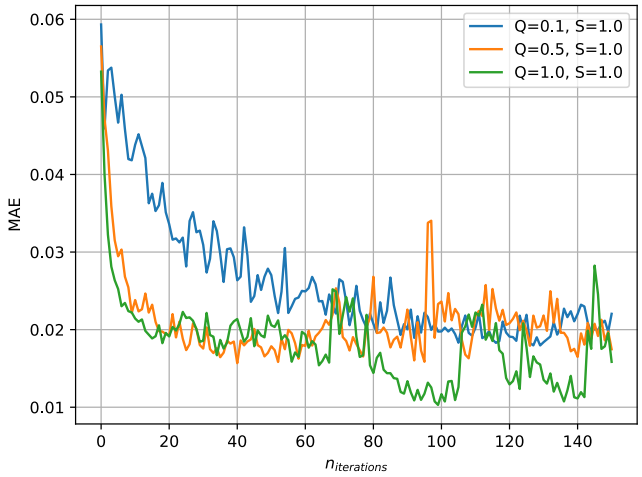



Fig. 11. Comparison between different values for Q in the learning rate equation.

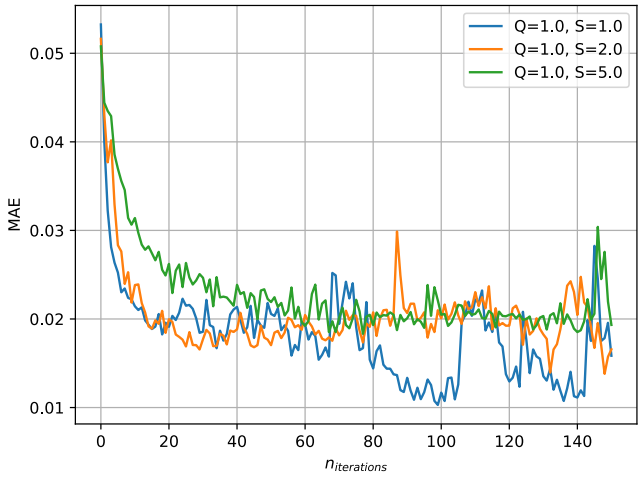



Fig. 12. Comparison between different values for S in the learning rate equation.

[24]. In Fig. 11, $S$ is fixed at $1.0I_{n_j \times n_j}$ at three values for $Q$, which are 0.1, 0.5 and 1.0. This figure shows that by increasing $Q$ the rate of convergence is faster as well. However, larger penalty factors comes at risk of stability, as can be seen from $Q = 0.5, n_{iter} = 100$ and $Q = 1.0, n_{iter} = 105$, where MAE spikes are observed. Fig. 12 shows a fixed variable for $Q$, and $S$ has a range between 1.0, 2.0 and 5.0. Bristow *et al.* mention that the value of $S$ does not have an influence on the asymptotic error value, which can also be seen from Fig. 12, where the MAE for all three lines have the same value at $n_{iterations} = 150$. Furthermore, it can be concluded that a higher value for $S$ yields in less variation of the MAE. This can be seen when comparing $Q = 1.0, S = 1.0$ with $Q = 1.0, S = 5.0$, where the latter one monotonously decreases faster compared to the former.

*4) Stiffness Comparison:* Della Santina *et al.* [39] mention that there is a inverse relationship between stiffness and tracking performance. In the context of SEA's a higher $PD_{feedback}$ is equivalent to adding a second spring in parallel, which means the system acts more in a rigid manner. However, the trade-off is that tracking performance is reduced. Fig. 13 confirms this hypothesis, where we show the results of three $PD_{feedback}$ values of $K_P = 30, 50$ and 70. Indeed, a higher stiffness of $K_P = 70$ results in a lower MAE compared $K_P = 30$. From simulation, it is obtained that with $K_P = 30$ the robot is able to jump higher in vertical direction, but not in the positive horizontal direction. With $K_P = 50$ and 70 the robot is able to successfully perform the forward pronking motion.

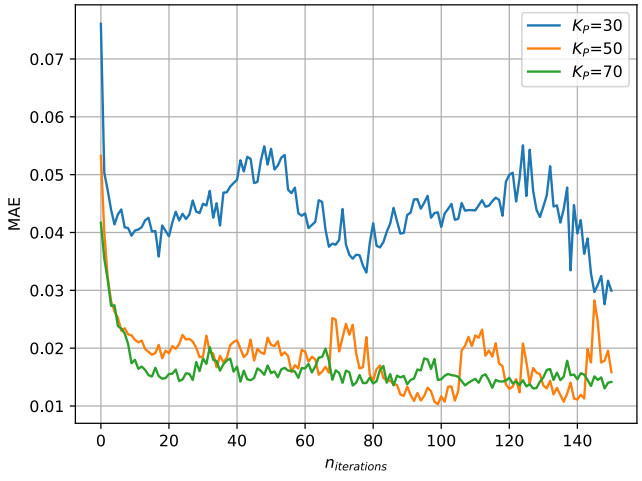



Fig. 13. Comparison between different values for $PD_{feedback}$.

Fig. 14. Snapshots of the rough terrain experiment in the Gazebo simulator. The uneven ground is set at a height of 1 cm.

### D. Rough Terrain

To test the robustness of the whole framework we put the robot in a rough environment as shown in Fig. 14. In this environment we add uneven ground, with a maximum height of 1 cm. From Fig. 14 it is also observed that the robot is able to successfully traverse through the rough terrain. Fig. 15 shows that at iteration 1 the tracking error is large for the right front- and hind knees between $t = 0.15 - 0.20[s]$. After 150 iterations, fILC manages again to reduce this tracking error. From Fig. 16 it is observed that the MAE drops down to a value of 0.02 $[rad]$.

## V. Experimental Results

This section elaborates on the experimental results obtained during the tests of the robot. Again, the setup will be explained first. Subsequently, the experiments will be discussed afterwards.

### A. Setup

The MULINEX prototype robot, as illustrated by Fig. 17, is to be used in this experiment. MULINEX is a low cost 8-DoF quadrupedal robot with modular Series Elastic Actuators (SEA), developed by Research Center E. Piaggio at University of Pisa [11]. With SEA's embedded into the system its purpose is to perform highly dynamic tasks, such as jumping.

### B. Indoor Experiment

As mentioned before in Section IV-A, both motor and link side reference trajectories are used in the real experiment. The link side positions are measured using the second encoders which are located on the link side position of the actuator. The Radial Gaussian type of basis function for the controller is used here with $Q = 0.01$, $S = 2.0$, $K_P = 50$ and $K_P = 0.4$. Fig. 17 shows snapshots of the indoor experiment. During the experiment it is observed that the robot successfully achieves a higher hopping height. Another observation is that the robot tends to hopping backwards in negative x-direction. Fig. 18 shows the tracking performance for both motor- and link right side joints. Fig. 19 illustrates the control input and learning weights $\alpha$ at 150-th iteration. From Fig. 18 it can be observed that the link side measurements have a sensor bias for all joints. If we compare the first iteration with iteration 150, we observe worse tracking for the motor references and a slightly better tracking performance for the link references. Fig. 20 shows that the MAE is able to converge at a slow rate. Spikes are observed at $n_{iteration} = 50, 120$ and $158$, which can be explained by the fact that the laptop is sometimes not able to send the commanding joint angles to the robot due to wireless connectivity issues. The overall magnitude of the MAE is larger compared to the simulation results. This could be due to multiple reasons. The first reason is that the precise stiffness of the belt is unknown, as another version of the EM-ACT actuator is used compared to the EM-ACT actuator written in [11]. The second reason is that the robot tends to move in negative x-direction, as we only control the robot in Joint coordinates. We do not have any information of the Cartesian state of the robot so we can not force the controller to go in positive x-direction. The third possible reason is that during the experiment the temperature of the motors in the hind legs quickly increases compared to the front legs. These temperatures quickly rose to 60° Celsius, after which the performance of these motors decrease significantly. A fourth possible reason for the relatively poor tracking performance is that there is a certain sensor bias in the link measurements, as shown in the lower four images of Fig. 18.

### C. Outdoor Experiment

The robustness of the framework is also tested in an outside environment, where the robot is put on a slope in front of the Engineering Faculty of University of Pisa, as illustrated in Fig. 24. The idea behind putting the robot on the slope is that weight vector now has a component in positive x-direction, such that the robot converges more quickly into the forward hopping motion. Furthermore, to put less stress on the motors of the hind legs the trunk angle $\alpha$ is fixed at $-5.0[deg]$. The results of the optimization trajectory is as follows: $CoT_{mech} = 4.01$, $\theta_{TD} = 0[deg]$, $l_{TD} = 0.28[m]$. During the experiment the robot is able to perform the pronking motion in positive x-direction, as shown in Fig. 21. Fig. 22 however shows that the overall tracking performance is poor for the first iteration. When we look at the results for iteration 50 we can see a slight improvement for all motor- and link measurements, however there is still a significant discrepancy between the reference and measured trajectories. The bias is also observed again in the link measurements. Fig. 24 shows that the MAE decreases in few iterations to an error of $0.0765[rad]$.

## VI. Conclusion & Future Work

In this work, we have combined three ideas into one framework. The first idea is to create a pronking motion for a quadrupedal robot using a reduced order model that takes the elasticity of the joints into account. With a template-anchor approach the elasticity of the joints, which are placed in series, are embedded into a Trunk-Spring Loaded Inverted Pendulum (T-SLIP) model. This model provides a periodic pronking motion of a quadrupedal robot. The second idea is the use of functional Iterative Learning Control (fILC) as a feedforward controller to preserve the elasticity of the quadruped. We show

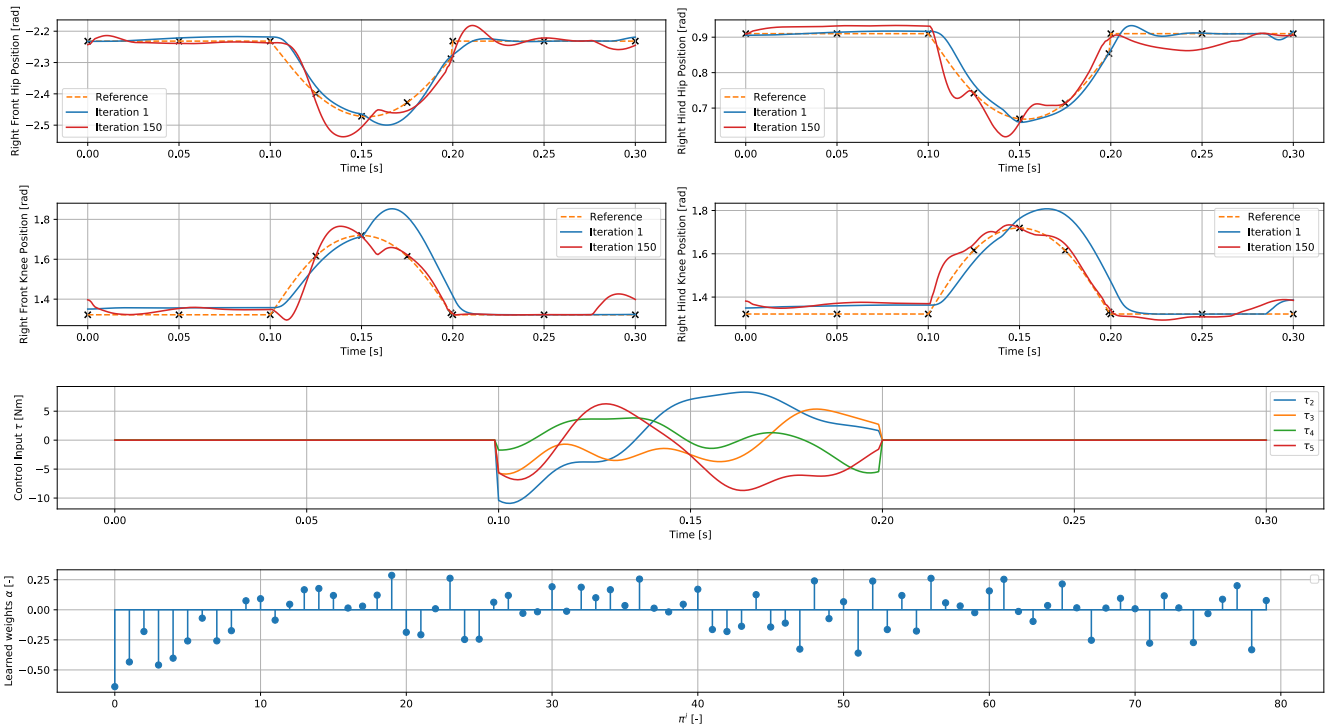



Fig. 15. Rough terrain simulation results. The upper four images show the reference tracking performance of the joints seen from the right hand side of the robot. In orange the reference trajectories that are obtained from TO are given. The black crosses indicate the tracked time instances. The torque control input and the learning weights of $\alpha$ at iteration 150 are shown in the lower two graphs.

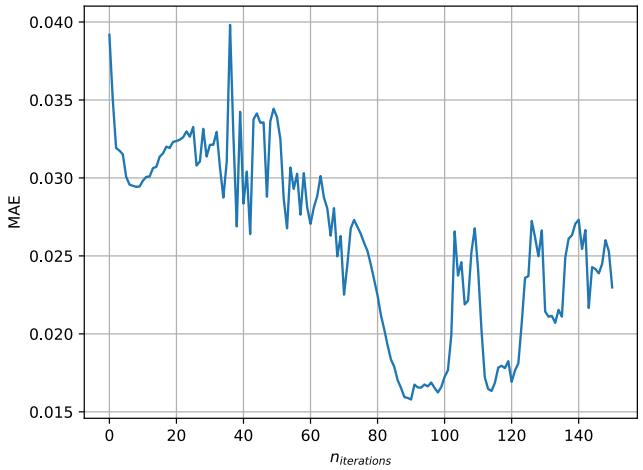



Fig. 16. This figure depicts the MAE against the number of iterations when testing the robot in a rough simulated environment.

that with fILC the trajectory error can be reduced in a short amount of iterations. These two ideas are validated in a newly developed series elastic actuated 8-DoF quadrupedal robot called 'MULINEX'. Here, the Trajectory Optimization (TO) is done offline and all the calculations from fILC can be done online.

From simulation it is observed that combining the first two ideas results in a successful pronking motion. Here, multiple hyperparameters such as the type of basis function, learning rate and stiffness are tested against one another to determine the best initialization parameters to be used during the real experiment. The three ideas combined are then validated on both in- and outdoor experiments. The indoor experiment shows that the robot is able to perform a pronking motion, and with fILC the MAE is able to converge monotonously. In the outdoor experiment, where the robot is put on a slope, we observe that the robot is able to pronk in forwards direction. The MAE converges to a lower value however there is still a large discrepancy between the performance of the robot in simulation compared to the real experiments.

Future work can improve the performance of the pronking motion by first getting the robot into the correct initial flight position and then performing the pronking motion. This was already attempted by combining the work of Gori *et al.* [17] with this thesis work. Gori's paper focuses on gait adaption using TO, in which the quadrupedal robot is able to perform a jumping motion from stand still. By connecting Gori's work with this thesis work the robot is already in the first flight phase after which the T-SLIP model performs the pronking gait. Another way to improve the pronking motion is to replace the ball feet with bioinspired adaptive feet to decrease the amount of slippage when performing the jumping task [40]. Some preliminary tests with these adaptive feet were done, however due to time constraints we were not able to perform a successful pronking motion with the adaptive feet.

Another possibility of improving the work in this thesis is to be able to measure the base states of the robot by means of an IMU. By giving both Cartesian and Joint reference signals

(a) Flight phase (b) TD event (c) Stance phase (d) TO event (e) Flight phase

Fig. 17. Snapshots of the indoor experiment with the MULINEX robot.

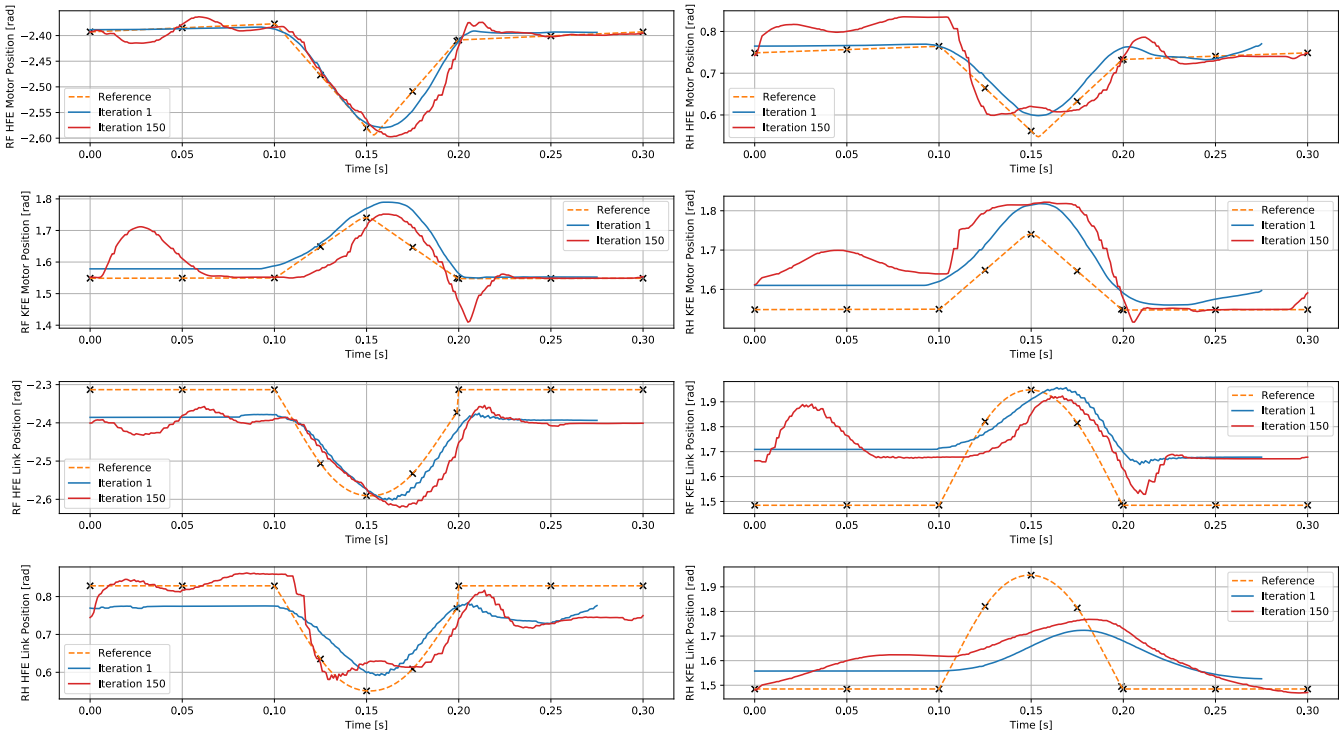



Fig. 18. Tracking performance for motor- and link positions for the indoor experiment. The eight images show the reference tracking performance of the motor- and link trajectories seen from the right hand side of the robot. In orange the reference trajectories that are obtained from trajectory optimization are given. The black crosses indicate the tracked time instances. The acronyms on the y-axes are as follows; RF HFE = Right Front Hip, RF KFE = Right Front Knee, RH HFE = Right Hind Hip, RH KFE = Right Hind Knee.

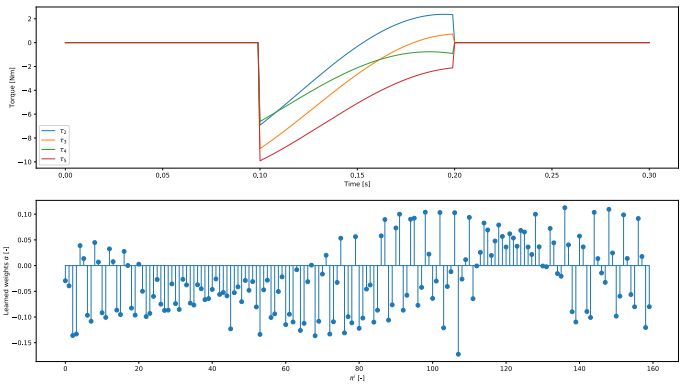


Fig. 19. Control input torques and $\alpha$ weight distribution for the indoor experiment.

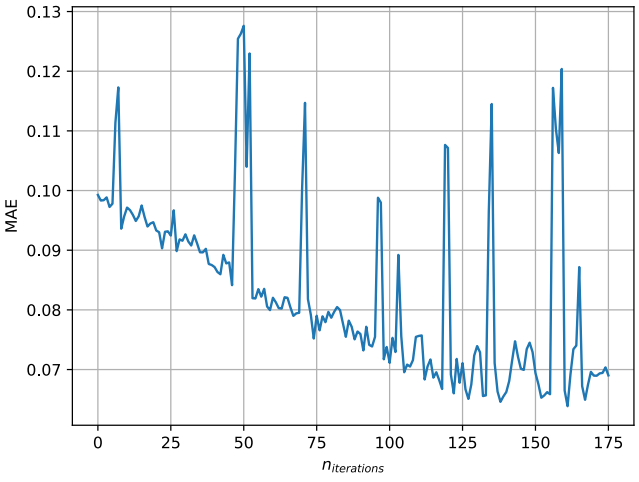



Fig. 20. MAE plotted against the number of iterations for the indoor experiment.

(a) Flight phase (b) TD event (c) Stance phase (d) TO event (e) Flight phase

Fig. 21. Snapshots of the outdoor experiment with the MULINEX robot.

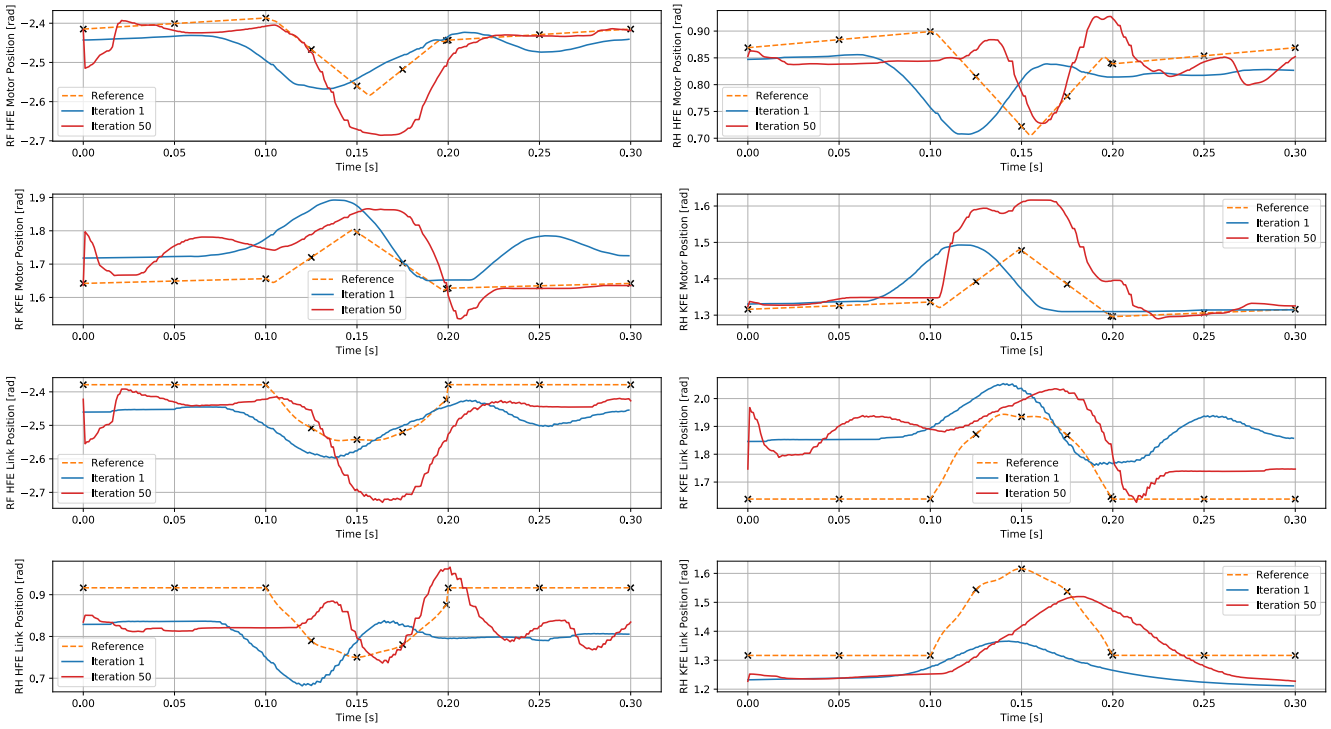



Fig. 22. Tracking performance for motor- and link positions for the outdoor experiment. The eight images show the reference tracking performance of the motor- and link trajectories seen from the right hand side of the robot. In orange the reference trajectories that are obtained from trajectory optimization are given. The black crosses indicate the tracked time instances. The acronyms on the y-axes are as follows; RF HFE = Right Front Hip, RF KFE = Right Front Knee, RH HFE = Right Hind Hip, RH KFE = Right Hind Knee.

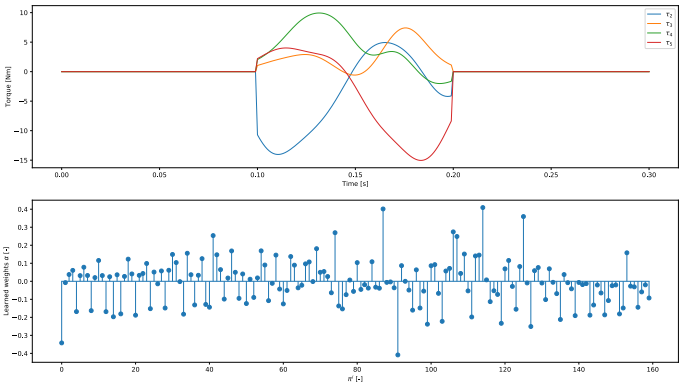



Fig. 23. Control input torques and $\alpha$ weight distribution for the outdoor experiment.

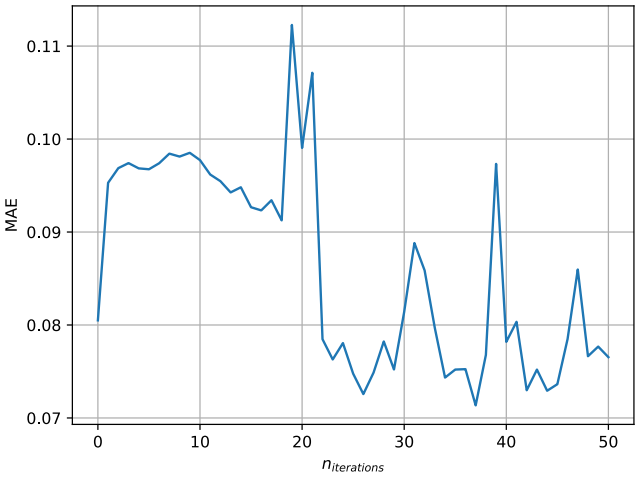



Fig. 24. MAE plotted against the number of iterations for the outdoor experiment.

to the robot it should be able to perform a succesful forward pronking motion. Other possibilities of improvement is to extend the T-SLIP model to include lateral motion of the robot or to use other types of gait in combination with fILC. At last, it would be also interesting to combine Reinforcement Learning (RL) with ILC. The main drawback of only using ILC is that is restricted to only one repetitive task. Combining RL and ILC can have the advantage that RL can achieve generalisation of multiple tasks whilst convergence is guaranteed with ILC. De Santis *et al.* [41] propose this idea for a 2-DoF SEA leg robot following a minimal jerk trajectory. De Santis' work could be combined with this work to achieve multiple jumping tasks in combination with fILC on a quadrupedal robot.

## REFERENCES


[1] R. Alley, T. Berntsen, N. L. Bindoff, Z. Chen, A. Chidthaisong, P. Friedlingstein, J. Gregory, G. Hegerl, M. Heimann, B. Hewitson, B. Hoskins, F. Joos, J. Jouzel, V. Kattsov, U. Lohmann, M. Manning, T. Matsuno, M. Molina, N. Nicholls, J. Overpeck, D. Qin, G. Raga, V. Ramaswamy, J. Ren, M. Rusticucci, S. Solomon, R. Somerville, T. F. Stocker, P. Stott, R. J. Stouffer, P. Whetton, R. A. Wood, D. Wratt, J. Arblaster, G. Brasseur, J. H. Christensen, K. Denman, D. W. Fahey, P. Forster, E. Jansen, P. D. Jones, R. Knutti, H. Le Treut, P. Lemke, G. Meehl, P. Mote, D. Randall, D. A. Stone, K. E. Trenberth, J. Willebrand, and F. Zwiers, "Climate Change 2007: The Physical Science Basis," tech. rep., IPCC, 2007.

[2] A. Costello, M. Abbas, A. Allen, S. Ball, S. Bell, R. Bellamy, S. Friel, N. Groce, A. Johnson, M. Kett, M. Lee, C. Levy, M. Maslin, D. McCoy, B. McGuire, H. Montgomery, D. Napier, C. Pagel, J. Patel, J. A. P. de Oliveira, N. Redclift, H. Rees, D. Rogger, J. Scott, J. Stephenson, J. Twigg, J. Wolff, and C. Patterson, "Managing the health effects of climate change," *The Lancet*, vol. 373, no. 9676, 2009.

[3] F. Angelini, P. Angelini, C. Angiolini, S. Bagella, F. Bonomo, M. Caccianiga, C. D. Santina, D. Gigante, M. Hutter, T. Nanayakkara, P. Remagnino, D. Torricelli, and M. Garabini, "Robotic Monitoring of Habitats: the Natural Intelligence Approach," *IEEE Access*, 2023.

[4] G. A. Cavagna, N. C. Heglund, and C. R. Taylor, "Mechanical work in terrestrial locomotion: two basic mechanisms for minimizing energy expenditure," *American Journal of Physiology - Regulatory Integrative and Comparative Physiology*, vol. 2, no. 3, 1977.

[5] R. M. N. Alexander, "Tendon elasticity and muscle function," *Comparative Biochemistry and Physiology - A Molecular and Integrative Physiology*, vol. 133, no. 4, 2002.

[6] R. M. Alexander, "Energy-saving mechanisms in walking and running.," 1991.

[7] J. Chen, Z. Liang, Y. Zhu, C. Liu, L. Zhang, L. Hao, and J. Zhao, "Towards the exploitation of physical compliance in segmented and electrically actuated robotic legs: A review focused on elastic mechanisms," *Sensors (Switzerland)*, vol. 19, 12 2019.

[8] P. Arm, R. Zenkl, P. Barton, L. Beglinger, A. Dietsche, L. Ferrazzini, E. Hampp, J. Hinder, C. Huber, D. Schaufelberger, F. Schmitt, B. Sun, B. Stolz, H. Kolvenbach, and M. Hutter, "SpaceBok: A dynamic legged robot for space exploration," in *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2019-May, 2019.

[9] M. Hutter, C. Gehring, D. Jud, A. Lauber, C. D. Bellicoso, V. Tsounis, J. Hwangbo, K. Bodie, P. Fankhauser, M. Bloesch, R. Diethelm, S. Bachmann, A. Melzer, and M. Hoepflinger, "ANYmal - A highly mobile and dynamic quadrupedal robot," in *IEEE International Conference on Intelligent Robots and Systems*, vol. 2016-November, 2016.

[10] J. Ding, P. Posthoorn, V. Atanassov, F. Boekel, J. Kober, and C. D. Santina, "Quadrupedal Locomotion With Parallel Compliance: E-Go Design, Modeling, and Control," *IEEE/ASME Transactions on Mechatronics*, vol. 29, pp. 2839–2848, 8 2024.

[11] R. k. M. Gopanunni, L. Martignetti, F. Iotti, A. Ranjan, F. Angelini, and M. Garabini, "EM-Act: A Modular Series Elastic Actuator for Dynamic Robots," *IEEE Open Journal of the Industrial Electronics Society*, pp. 1–14, 2024.

[12] M. A. Sharbafi and A. Seyfarth, *Bioinspired Legged Locomotion: Models, Concepts, Control and Applications*. Butterworth-Heinemann Inc, 2017.

[13] S. Collins, A. Ruina, R. Tedrake, and M. Wisse, "Efficient bipedal robots based on passive-dynamic walkers," *Science*, vol. 307, no. 5712, 2005.

[14] A. V. Vivas, A. Cherubini, M. Garabini, P. Salaris, and A. Bicchi, "Minimizing Energy Consumption of Elastic Robots in Repetitive Tasks," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 53, no. 8, 2023.

[15] J. Ding, M. A. L. Sels, F. Angelini, J. Kober, and C. D. Santina, "Robust Jumping with an Articulated Soft Quadruped Via Trajectory Optimization and Iterative Learning," *IEEE Robotics and Automation Letters*, vol. 9, no. 1, 2024.

[16] S. Gilroy, D. Lau, L. Yang, E. Izaguirre, K. Biermayer, A. Xiao, M. Sun, A. Agrawal, J. Zeng, Z. Li, and K. Sreenath, "Autonomous Navigation for Quadrupedal Robots with Optimized Jumping through Constrained Obstacles," in *2021 IEEE 17th International Conference on Automation Science and Engineering (CASE)*, pp. 2132–2139, IEEE, 8 2021.

[17] P. Gori, M. Pierallini, F. Angelini, and M. Garabini, "Gait Adaptation and Iterative Control: a Switched System Optimization Framework for Quadrupedal Robots." 2024.

[18] R. Blickhan, "The spring-mass model for running and hopping," *Journal of Biomechanics*, vol. 22, no. 11-12, 1989.

[19] J. Ding, V. Atanassov, E. Panichi, J. Kober, and C. D. Santina, "Robust Quadrupedal Jumping With Impact-Aware Landing: Exploiting Parallel Elasticity," *IEEE Transactions on Robotics*, vol. 40, pp. 3212–3231, 2024.

[20] B. Siciliano and O. Khatib, *Springer handbook of robotics*. Springer, 2016.

[21] C. Nguyen, L. Bao, and Q. Nguyen, "Continuous Jumping for Legged Robots on Stepping Stones via Trajectory Optimization and Model Predictive Control," in *Proceedings of the IEEE Conference on Decision and Control*, vol. 2022-December, 2022.

[22] Q. Nguyen, M. J. Powell, B. Katz, J. D. Carlo, and S. Kim, "Optimized jumping on the MIT cheetah 3 robot," in *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2019-May, 2019.

[23] C. Della Santina, M. Bianchi, G. Grioli, F. Angelini, M. Catalano, M. Garabini, and A. Bicchi, "Controlling Soft Robots: Balancing Feedback and Feedforward Elements," *IEEE Robotics and Automation Magazine*, vol. 24, pp. 75–83, 9 2017.

[24] D. A. Bristow, M. Tharayil, and A. G. Alleyne, "Survey Of Iterative Learning Control: A Learning-Based Method for High-Performance Tracking Control," *IEEE Control Systems*, vol. 26, no. 3, 2006.

[25] M. Pierallini, F. Angelini, R. Mengacci, A. Palleschi, A. Bicchi, and M. Garabini, "Iterative Learning Control for Compliant Underactuated Arms," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 53, no. 6, 2023.

[26] M. Pierallini, F. Angelini, A. Bicchi, and M. Garabini, "Swing-Up of Underactuated Compliant Arms Via Iterative Learning Control," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, 2022.

[27] S. P. Chhatoi, M. Pierallini, F. Angelini, C. Mastalli, and M. Garabini, "Optimal Control for Articulated Soft Robots," *IEEE Transactions on Robotics*, vol. 39, no. 5, 2023.

[28] C. Della Santina and F. Angelini, "Iterative Learning in Functional Space for Non-Square Linear Systems," in *Proceedings of the IEEE Conference on Decision and Control*, vol. 2021-December, pp. 5858–5863, Institute of Electrical and Electronics Engineers Inc., 2021.

[29] S. Drost, P. Pustina, F. Angelini, A. De Luca, G. Smit, and C. Della Santina, "Experimental Validation of Functional Iterative Learning Control on a One-Link Flexible Arm," in *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2023-May, 2023.

[30] M. P. Kelly, "Transcription Methods for Trajectory Optimization: a beginners tutorial," tech. rep., Cornell University, 7 2017.

[31] M. A. Sharbafi, C. Maufroy, H. M. Maus, A. Seyfarth, M. N. Ahmadabadi, and M. J. Yazdanpanah, "Controllers for robust hopping with upright trunk based on the Virtual Pendulum concept," in *IEEE International Conference on Intelligent Robots and Systems*, 2012.

[32] M. W. Spong, "Modeling and control of elastic joint robots," *Journal of Dynamic Systems, Measurement and Control, Transactions of the ASME*, vol. 109, no. 4, 1987.

[33] A. De Luca, "Robots with Flexible Joints: Modeling and Control Control of Soft and Articulated Elastic Robots," tech. rep., Sapienza Università Di Roma, Rome, 5 2023.

[34] N. Amann, D. H. Owens, and E. Rogers, "Iterative learning control for discrete-time systems with exponential rate of convergence," *IEE Proceedings: Control Theory and Applications*, vol. 143, no. 2, 1996.

[35] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi: a software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, pp. 1–36, 3 2019.

[36] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, pp. 25–57, 3 2006.
[37] S. Macenski, T. Foote, B. Gerkey, C. Lalancette, and W. Woodall, "Robot Operating System 2: Design, architecture, and uses in the wild," *Science Robotics*, vol. 7, 5 2022.
[38] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, pp. 2149–2154, IEEE, 2004.
[39] C. Della Santina, M. G. Catalano, and A. Bicchi, "Soft Robots," in *Encyclopedia of Robotics*, pp. 1–15, Springer Berlin Heidelberg, 2021.
[40] A. Ranjan, F. Angelini, T. Nanayakkara, and M. Garabini, "Design Guidelines for Bioinspired Adaptive Foot for Stable Interaction with the Environment," *IEEE/ASME Transactions on Mechatronics*, vol. 29, no. 2, 2024.
[41] Y. De Santis, M. Pierallini, F. Angelini, and M. Garibini, "Design and Implementation of a Novel Control Strategy Combining Reinforcement Learning and Iterative Learning Control." 2024.

## APPENDIX A

The stance dynamics can be derived using T-SLIP and reduced Spong model. We first take the Cartesian coordinates $\xi = [x, z, \alpha]$. With assumption A2 and A3 we can state that:

$$T_{link} = \frac{1}{2}m(\dot{x}^2 + \dot{z}^2) + \frac{1}{2}J\dot{\alpha}^2 \quad \text{(Cartesian space)}, \tag{40}$$

$$V_{link} = mgz \qquad \text{(Cartesian space)}. \tag{41}$$

With assumption A2 and A4 we can state that:

$$T_{motor} = \frac{1}{2}I_2\dot{\theta_2}^2 + \frac{1}{2}I_3\dot{\theta_3}^2 + \frac{1}{2}I_4\dot{\theta_4}^2 + \frac{1}{2}I_5\dot{\theta_5}^2 \tag{42}$$

$$\begin{aligned} V_{motor} = &+\frac{1}{2}K_2(\theta_2 - q_2)^2 + \frac{1}{2}K_3(\theta_3 - q_3)^2 \\ &+\frac{1}{2}K_3(\theta_4 - q_4)^2 + \frac{1}{2}K_3(\theta_5 - q_5)^2. \end{aligned} \tag{43}$$

The Lagrangian formulation can be derived as follows:

$$L = T - V = (T_{link} + T_{motor}) + (V_{link} + V_{motor}), \tag{44}$$

and the Euler-Langragian equations are shown in (45) and (46).

$$\frac{d}{dt}(\frac{\partial L}{\partial \dot{\xi}})^T - (\frac{\partial L}{\partial \xi})^T = 0 \tag{45}$$

$$\frac{d}{dt}(\frac{\partial L}{\partial \dot{\theta}})^T - (\frac{\partial L}{\partial \theta})^T = \tau_m \tag{46}$$

Substitution of (40)-(44) into the Euler-Lagrangian derivation (45)-(46) yields

$$\begin{aligned} &\ddot{\xi} = M(\xi)^{-1}(-G(\xi) + J_h^T(\xi)K(\theta - q)) \quad \text{(Cartesian space)}, \\ &\ddot{\theta} = B^{-1}(\tau_m - K(\theta - q)) \qquad \text{(Joint space)}. \end{aligned} \tag{47}$$