Settling Time Reduction for Underactuated Walking Robots

Sotiris Apostolopoulos[1,2], Marion Leibold[1] and Martin Buss[1,2]

Abstract— **This paper introduces a novel way to improve the settling time of transitions between different walking controllers. This improvement is achieved by commanding a sequence of intermediate transitions to the target controller. As a result, the state of the system enters the domain of attraction of the target controller closer to the fixed point of the Poincaré Map. The method is applicable to any walking robot with one degree of underactuation. The problem is expressed as a Markov Decision Process and then solved with Reinforcement Learning. In order to simplify the stability analysis of underactuated walking the Hybrid Zero Dynamics framework is utilized. Another advantage of using the Hybrid Zero Dynamics is the dimensionality reduction of the state representation in the Markov Decision Process. The experimental results suggest that the proposed methodology performs better than a one-step transition for 84.34% of all the considered transitions for a simulated walking robot matching the parameters of RABBIT [1].**

I. Introduction

The settling time of a transition between periodic controllers can be defined as the time until convergence to the new limit cycle. Any effort in minimizing it, can be considered equivalent to improving the reaction time of the system to unexpected situations. Assume that an abrupt change of velocity is required, thus a new controller has to be commanded. In the case of underactuation, a transition between two different controllers will not drive the state of the system to the fixed point of the target controller, but in a region around it, called the domain of attraction. Thus, the entry point of the domain of attraction plays an important role in the rate of convergence to the limit cycle and as a consequence to the desired velocity.

One possible way to improve the settling time of such a transition is to minimize the maximum eigenvalue of the Poincaré Map, as done in [2] for a simulated running robot. This minimization led to improvement in convergence rate and stability. As discussed though in this work [2], eigenvalue optimization constitutes a difficult problem. One of the reasons behind this is the fact that the construction of the Poincaré Map requires the computation of the first order sensitivities of the discontinuous joint trajectories. The suggested way of treating such an objective is based on a two layer optimization procedure where the outer loop performs stability optimization while the inner loop optimizes an energy criterion based on the parameters delivered from the outer loop.

Another way of dealing with the problem of settling time reduction is to consider a sequence of controllers that enters the domain of attraction of the target controller closer to the fixed point, than if a one-step transition was taken as suggested in [3]. When such methods are utilized, feasibility has to be ensured such that a sequence will drive the state of the system in the domain of attraction of the target controller.

The feasibility problem has been treated in [4] and [5]. In [4], the idea of LQR-Trees is introduced, where the state space is partitioned in different regions. Each region corresponds to the domain of attraction of a LQR controller which is computed based on a conservative approach using Sum-of-Squares optimization. Thus, the system can be driven to a final state from any initial one by finding a sequence of LQR controllers. This idea is claimed to be extendible to walking robots as well. In [5] the framework of Sequential Composition Control is introduced. The idea is more generic than the LQR-Trees, since it describes how any kind of controllers can be composed into a sequence in order to accomplish a high-level plan. The feasibility condition states that a transition between different controllers is feasible only when the image of the funnel of one controller belongs to the domain of attraction of the other one.

In related work [6], the idea of Sequential Composition Control was used to define feasibility conditions for transitions between different controllers. When a transition was not feasible, a connecting controller was learnt online. Despite the fact that these ideas describe ways to create a composite controller as a concatenation of different ones, they do not take into account any optimality criteria for the way this composite controller is generated, but only focus on feasibility. In this paper we overcome this limitation by expressing the problem of controller composition as a Markov Decision Process and by introducing a reward function that takes into account optimality criteria related with the task under study.

In this paper, we study the case of underactuated walking. We assume a set of controllers and consider the problem of reducing the settling time of a transition between different controllers. Due to the underactuation, the feasibility of a transition is not pre-determined but rather has to be verified at each impact event (i.e. swing leg establishing contact with the ground). The settling time reduction is set up as a Markov Decision Process and then solved with a realization of Reinforcement Learning. A contribution of using the Hybrid Zero Dynamics framework is that the proposed methodology can be extended to any walking robot with one degree of underactuation. In addition, the state of the Markov Decision Process has lower dimensionality.

[1] Chair of Automatic Control Engineering, Technische Universität München, Theresienstr. 90, 80333 München, Germany, {sotiris.apostolopoulos, marion.leibold, mb}@tum.de

[2] TUM Institute for Advanced Study, Technische Universität München, Lichtenbergstrasse 2a, 85748 Garching, Germany

The paper is structured as follows: Sections II and III present the dynamics of underactuated walking and the Hybrid Zero Dynamics framework, respectively. In section IV the problem under study is formulated as a Markov Decision Process and the Reinforcement Learning algorithm is presented. The simulation results are presented in section V. Section VI concludes the paper.

II. Underactuated Walking

Underactuated walking is modeled as a hybrid process with two discrete states: the single support and the rigid impact. In order to explain these two phases more thoroughly, the dynamics of underactuated walking are introduced. They are based on the Lagrangian formulation and the assumption of rigid bodies. Motion is restricted in the sagittal plane, but can be extended to the 3D case [7].

A. Single Support

During the single support, the legs of the robot are labelled as ”stance” and ”swing”. The stance leg is pinned on the ground and the swing leg moves forward with an adequate foot clearance in order to become the new stance leg, concluding the single support phase. The state $\boldsymbol{x}$ of such a robot contains the joint positions $\boldsymbol{q}$ and the joint velocities $\dot{\boldsymbol{q}}$, i.e. $\boldsymbol{x} = [\boldsymbol{q}^T\ \dot{\boldsymbol{q}}^T]^T$. Thus, a robot with n degrees of freedom has a $2n$-dimensional state. Utilizing Lagrangian dynamics, the equations of motion can be expressed as

$$\boldsymbol{D}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{G}(\boldsymbol{q}) = \boldsymbol{B}\boldsymbol{u}, \tag{1}$$

where $\boldsymbol{D}(\boldsymbol{q}) \in \mathbb{R}^{n\times n}$ is the mass-inertia matrix, $\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \in \mathbb{R}^{n\times n}$ is the matrix of centrifugal and Coriolis terms, $\boldsymbol{G}(\boldsymbol{q}) \in \mathbb{R}^{n}$ summarizes the gravitational terms, $\boldsymbol{B} \in \mathbb{R}^{n\times(n-1)}$ is the input matrix and $\boldsymbol{u} \in \mathbb{R}^{n-1}$ is the vector of generalized torques. The challenge in controlling such a system is due to the fact that the input matrix $\boldsymbol{B}$ is non-square. Thus, when applying input-output feedback linearization methods, there will be dynamics which are non-observable known as *zero dynamics*. However, they have to be taken into account when designing individual walking gaits but also when concatenating different gaits in a single motion plan. Section III explains the *zero dynamics* and the methodology to design feedback controllers for underactuated walking robots in more detail.

B. Rigid Impact

The rigid impact takes place when the swing leg establishes contact with the ground. The impact is assumed to be inelastic and instantaneous. At the rigid impact, the leg previously pinned on the ground (i.e. the stance leg) loses contact with the ground and the role of the legs is switched. The impact causes a discontinuity on the joint velocities $\dot{\boldsymbol{q}}$ which can be determined by the impact map $\boldsymbol{\Delta}$ and the pre-impact joint velocities $\dot{\boldsymbol{q}}^-$. Instead of introducing an additional model for the single support with the new stance leg, we relabel (or equivalently transform) the coordinates of the robot. Formally, this can be expressed as

$$\begin{aligned} \boldsymbol{q}^+ &= \boldsymbol{R}\boldsymbol{q}^- \\ \dot{\boldsymbol{q}}^+ &= \boldsymbol{R}\boldsymbol{\Delta}(\boldsymbol{q}^-)\dot{\boldsymbol{q}}^- = \boldsymbol{\Delta}_s(\boldsymbol{q}^-)\dot{\boldsymbol{q}}^- \end{aligned} \tag{2}$$

where the plus and minus superscripts denote the post-impact and pre-impact state of the system, respectively. The relabelling matrix is denoted by $\boldsymbol{R}$ and is circular (i.e. $\boldsymbol{R}\boldsymbol{R} = \boldsymbol{I}$).

III. Hybrid Zero Dynamics of Walking

Controller synthesis for underactuated walking robots has been a topic of extensive investigation in the literature [3], [8], [9]. The main idea is to design a set of virtual holonomic constraints and enforce them by input-output feedback linearization. More specifically, the controller design can be formulated as a tracking control problem where the outputs are defined as

$$h_i(t) = q_i(t) - q_i^d(t),\ i = 1, ..., n-1, \tag{3}$$

where q_i^d is the desired trajectory corresponding to the i-th degree of freedom (DoF). The index i runs from 1 to $n-1$, since a desired trajectory cannot be enforced on the underactuated DoF q_n (without loss of generality q_n is the underactuated DoF). In walking, time t can be replaced by a monotonically increasing variable $\theta(\boldsymbol{x})$, which replaces trajectories by paths and we can rewrite $\boldsymbol{h}(t)$ as $\boldsymbol{h}(\theta(\boldsymbol{x})) = \boldsymbol{h}(\boldsymbol{x})$. This variable is usually the underactuated coordinate of the robot q_n or a function of it. In order to facilitate the design process of the walking controller, a short introduction to the Hybrid Zero Dynamics of walking is necessary. Connection with the robotic model under study can be found in Fig. 1. For more details on Hybrid Zero Dynamics, the reader is encouraged to refer to [3] or [10].

A. Control Law

The main principle in Hybrid Zero Dynamics is the introduction of a coordinate transformation, such that the

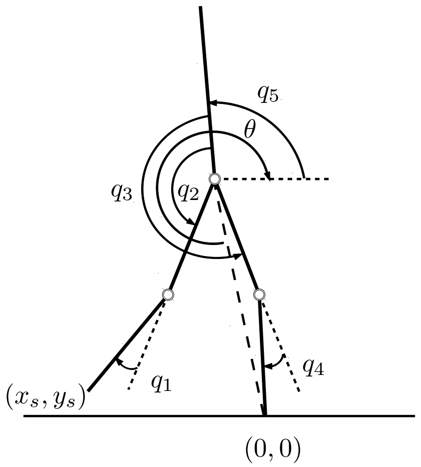

Fig. 1. Kinematic model of the biped under study. The underactuated DoF is the torso angle q_5. The Cartesian coordinates of the swing leg are denoted as (x_s, y_s). The x-axis is pointing to the right and the y-axis upwards.

outputs $\boldsymbol{h}$ and their time derivatives are zeroed and the zero dynamics are periodically stable.

Assuming that the dynamics of the robot are expressed in the state-space form

$$\begin{aligned} \dot{\boldsymbol{x}} &= \begin{bmatrix} \dot{\boldsymbol{q}} \\ \boldsymbol{D}(\boldsymbol{q})^{-1}(-\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} - \boldsymbol{G}(\boldsymbol{q}) + \boldsymbol{B}\boldsymbol{u}) \end{bmatrix} \\ &= \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x})\boldsymbol{u}, \end{aligned} \tag{4}$$

the feedback controller which zeroes the outputs $\boldsymbol{h}$ is given by

$$\boldsymbol{u}(\boldsymbol{x}) = (\boldsymbol{L_g}\boldsymbol{L_f}\boldsymbol{h}(\boldsymbol{x}))^{-1}(\boldsymbol{v}(\boldsymbol{x}) - \boldsymbol{L_f^2}\boldsymbol{h}(\boldsymbol{x})), \tag{5}$$

where the Lie derivatives are defined as

$$\boldsymbol{L_g}\boldsymbol{L_f}\boldsymbol{h}(\boldsymbol{x}) = \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{q}}\boldsymbol{D}^{-1}\boldsymbol{B}, \tag{6}$$

and

$$\boldsymbol{L_f^2}\boldsymbol{h}(\boldsymbol{x}) = \begin{bmatrix} \frac{\partial}{\partial \boldsymbol{q}}(\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{q}}\dot{\boldsymbol{q}}) & \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{q}} \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{q}} \\ \boldsymbol{D}^{-1}(-\boldsymbol{C}\dot{\boldsymbol{q}} - \boldsymbol{G}) \end{bmatrix}. \tag{7}$$

Here the arguments of the matrix and vector functions are omitted for brevity. The term $\boldsymbol{v}(\boldsymbol{x})$ in (4) is taken to be a PD term. Under the control law (5), the outputs are zeroed and the zero dynamics need to be checked for orbital stability.

B. Zero Dynamics Manifold

The Zero Dynamics Manifold is formally defined as

$$\boldsymbol{\mathcal{Z}} = \{\boldsymbol{x} | \boldsymbol{h}(\boldsymbol{x}) = 0, \boldsymbol{L_f}\boldsymbol{h}(\boldsymbol{x}) = 0\}, \tag{8}$$

where $\boldsymbol{L_f}\boldsymbol{h}(\boldsymbol{x}) = \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{q}}\dot{\boldsymbol{q}}$.

Let $\boldsymbol{x} \in \mathcal{Z}$ and define $\boldsymbol{\gamma}_0$ as the last row of the mass-inertia matrix $\boldsymbol{D}$, then the coordinates can be transformed into

$$\xi_1 = \theta, \ \xi_2 = \boldsymbol{\gamma_0}\dot{\boldsymbol{q}}. \tag{9}$$

The variable θ is shown in Fig. 1 and can be formally defined as $\theta = \boldsymbol{c}^T\boldsymbol{q} = \begin{bmatrix} -1 & 0 & -1/2 & 0 & -1 \end{bmatrix} \boldsymbol{q}$. The variable ξ_2 is the angular momentum conjugate to the underactuated DoF q_5. With this transformation the joint positions and velocities can be reconstructed by

$$\boldsymbol{q} = \boldsymbol{H}^{-1} \begin{bmatrix} \boldsymbol{q}^d \\ \xi_1 \end{bmatrix} \text{ and } \dot{\boldsymbol{q}} = \begin{bmatrix} \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{q}} \\ \boldsymbol{\gamma_0} \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ \xi_2 \end{bmatrix}, \tag{10}$$

where $\boldsymbol{H} = \begin{bmatrix} \boldsymbol{H}_0 \\ \boldsymbol{c} \end{bmatrix}$ and $\boldsymbol{H}_0 = \begin{bmatrix} \boldsymbol{I}_{n-1} & \boldsymbol{0}_{(n-1)\times 1} \end{bmatrix}$.

The remaining analysis of the Hybrid Zero Dynamics follows from [3] and is given without any proofs. A difference is made on the description of the fixed point which corresponds to the post-impact state of the robot, instead of the pre-impact one. This difference is done to facilitate the formulation of the settling time reduction as a Markov Decision Process, as will be described in section IV.

C. Orbital stability of zero dynamics

The derivatives of ξ_1 and ξ_2 can be written as

$$\begin{aligned} \dot{\xi}_1 &= \kappa_1(\xi_1)\xi_2 \\ \dot{\xi}_2 &= \kappa_2(\xi_1) \end{aligned} \tag{11}$$

where

$$\kappa_1(\xi_1) = \frac{\partial \theta}{\partial \boldsymbol{q}} \begin{bmatrix} \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{q}} \\ \gamma_0 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{12}$$

$$\kappa_2(\xi_1) = -\frac{\partial V}{\partial q_n} \tag{13}$$

In (13), V is the potential energy function of the system (4). The impact event is taken into account by the resets

$$\xi_1^+ = \theta^+ \tag{14}$$

$$\xi_2^+ = \delta_{\text{zero}}\xi_2^- \tag{15}$$

The quantity δ_{zero} accounts for the angular momentum exchange at the impact and can be computed analytically based on the dynamics of the system, i.e.

$$\delta_{\text{zero}} = \boldsymbol{\gamma}_0(\boldsymbol{q}^+)\boldsymbol{\Delta}_s(\boldsymbol{q}^-) \begin{bmatrix} \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{q}}(\boldsymbol{q}^-) \\ \boldsymbol{\gamma}_0(\boldsymbol{q}^-) \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \tag{16}$$

Since the zero dynamics manifold is 2-dimensional, the stability analysis is simplified. In this manifold, Lagrangian dynamics can be introduced and kinetic and potential energy functions can be defined as $K_{\text{zero}}(\xi_1)$ and $V_{\text{zero}}(\xi_1)$ respectively. The formal definition of these functions is

$$K_{\text{zero}}(\xi_1) = \frac{1}{2}\left(\frac{\dot{\xi}_1}{\kappa_1(\xi_1)}\right)^2 = \zeta_2 \tag{17}$$

$$V_{\text{zero}}(\xi_1) = -\int_{\theta^+}^{\xi_1} \frac{\kappa_2(\xi_1)}{\kappa_1(\xi_1)} d\xi_1. \tag{18}$$

If the kinetic energy at the beginning of the walking motion ζ_2^+ is greater than the maximum value of the potential energy $V_{\text{zero}}^{\text{MAX}}$, the numeric integration of the zero dynamics (11) will yield a periodic orbit. Formally, if

$$V_{\text{zero}}^{\text{MAX}} - \zeta_2^+ < 0, \tag{19}$$

a periodic orbit exists and if $0 < \delta_{\text{zero}}^2 < 1$, it is exponentially stable. The associated Poincaré Map is given by

$$\rho(\zeta_2^+) = \delta_{\text{zero}}^2\zeta_2^- = \delta_{\text{zero}}^2(\zeta_2^+ - V_{\text{zero}}(\theta^-)). \tag{20}$$

The fixed point of this orbit is

$$\zeta_2^* = \frac{\delta_{\text{zero}}^2}{\delta_{\text{zero}}^2 - 1}V_{\text{zero}}(\theta^-), \ \delta_{\text{zero}}^2 \neq 1 \tag{21}$$

and its domain of attraction is the set

$$D_{\text{zero}} = \left\{\zeta_2^+ > 0 | \zeta_2^+ - V_{\text{zero}}^{\text{MAX}} > 0\right\}. \tag{22}$$

Thus, the dimensionality of the system can be reduced from $2n$ to 2. This leads to a 1-dimensional Poincaré Map where the stability analysis can be conducted with analytical expressions. The same holds for the domain of attraction.

D. Transition between Walking Controllers

A transition between periodic controllers allows aperiodic walking. Assume that a transition from controller Φ_i to a controller Φ_j is required. Then the transition is feasible only if

$$\zeta_2^+ - V_{\text{zero}}^{\text{MAX}, i\to j} > 0. \tag{23}$$

If equation (23) is fulfilled, the state of the robot after the impact will be inside the domain of attraction of the periodic controller Φ_j. The quantity $V_{\text{zero}}^{i\to j}$ can be computed as in (18) where the integration interval now is from θ_i^+ to θ_j^-. That means, the joint positions at the end of the transition will be identical to those of controller Φ_j, unlike the joint velocities.

After the feasibility condition for aperiodic walking has been defined, the settling time reduction can be investigated.

IV. Learning for Settling Time Reduction

The purpose of this work is to find a sequence of controllers in order to reduce the settling time of a transition from an initial walking controller Φ_{init} to another one Φ_{target}. In order to do so, we formulate the settling time reduction as a Markov Decision Process, which we solve with Reinforcement Learning. For this work, each controller corresponds to a desired average walking velocity.

A. One-step Transition

In the one-step approach, a transition between two different walking controllers is realized by checking first if (23) holds. If this is the case, the transition is executed and then the state of the robot is inside the domain of attraction of the target controller. Otherwise, an intermediate transition is taken to the controller whose domain of attraction can be reached by the initial controller and is closest (in terms of fixed point) to the target one. If the target controller is still unreachable, this process can be repeated. In any case, once in the domain of attraction of the target controller, the convergence to the fixed point is dictated by the quantities δ_{zero} and $V_{\text{zero}}(\theta^-)$, thus it is possible that convergence requires a lot of time and steps.

B. Multi-Step Transition

In this work, this slow convergence is confronted by requiring that the transition to the target controller Φ_{target} does not have to be realized following the one-step approach, but there might be a sequence of transitions that can potentially reduce the settling time. More formally, assume that in the multi-step transition the state of the robot enters the domain of attraction of Φ_{target} at time t^* and ζ_2^* denotes the fixed point of Φ_{target}. Since the one-step transition might be executing a step at time t^*, we allow it to conclude the step and measure ζ_2^+ at time $t^+ > t^*$.

Definition 1: A composite controller is successful if and only if

$$|\zeta_2^+(t^*) - \zeta_2^*| \leq |\zeta_2^+(t^+) - \zeta_2^*|, \tag{24}$$

□

If the entry point to the domain of attraction of the target controller is closer following the multi-step transition than the one-step, the principle of optimality can be utilized and claim that convergence is achieved faster. In order to find such multi-step transitions, we use Reinforcement Learning.

C. Reinforcement Learning

Reinforcement Learning has been proposed as a semi-supervised optimization method [11] and has been extensively used in the field of Robotics [12]. An optimization problem in the context of Reinforcement Learning can be defined as a Markov Decision Process described by the tuple $P(S, T, F, r, \gamma)$, where S is the state space, T is the action space, $F : S\times T \to S$ is the state transition function, r is the reward function and $\gamma \in [0, 1]$ is a discount factor. The state transition function F returns the state $\boldsymbol{s}_{k+1}$ when applying the action τ_k at state $\boldsymbol{s}_k$. The reward function returns a scalar value r_k after such a transition. The action selection should be dictated by a policy π, such that $\pi : S \times T \to S$.

The idea behind Reinforcement Learning is to find this policy π such that the discounted sum of future rewards is maximized. This discounted sum is represented by a state-action value function $Q^\pi : S \times T \to \mathbb{R}$. In a few words, we are trying to find a policy π such that

$$Q^\pi(\boldsymbol{s}_k, \tau_k) = \mathbb{E}\left(\sum_{i=0}^{\infty} \gamma^i r_{k+i+1}\right), \tag{25}$$

where $\mathbb{E}$ the expectation operator.

Different realizations have been proposed in the literature for finding such a policy. When dealing with large state-action spaces, a practical solution is to use approximation techniques in order to learn the Q function. These techniques are continuous and assume that the state space can be represented by a sufficiently large number of basis functions. In this work, a Q-learning method is adopted with linear parametrization such that $Q(\boldsymbol{s}_k, \tau_k) = \boldsymbol{\phi}^T(\boldsymbol{s}_k, \tau_k)\boldsymbol{\beta}$ and ϵ-greedy action selection, where a random action is taken with probability ϵ and an optimal one with probability $1 - \epsilon$. In this linear parametrization of Q, the parameter vector to be learnt is $\boldsymbol{\beta}$ and the parametrization of the state is given by the vector $\boldsymbol{\phi} = [\phi_1, \phi_2, ..., \phi_L]^T$, where each ϕ_i corresponds to a basis function and L is the total number of them.

In the Reinforcement Learning framework, the state-action space has to be defined for the Settling Time Reduction:

- The **state space** $S = [\zeta_{2,\text{min}}^+, \zeta_{2,\text{max}}^+] \times \{1, ..., \text{card}(\Phi)\}$. For this representation, ζ_2^+ is already defined and card denotes the cardinality of the set of controllers Φ. The discrete set $\{1, ..., \text{card}(\Phi)\}$ describes the domain of attraction where the state of the system currently belongs. The limits of ζ_2^+ are determined by the set of controllers Φ.
- The **action space** $T = \{1, ..., \text{card}(\Phi)\}$ corresponds to the domain of attraction where we want to drive the state of the system.

Note that if the state ζ_2^+ was replaced with the velocity of the robot, the integration of the equations of motion would be necessary and requires a considerable amount of time. When utilizing the Hybrid Zero Dynamics framework though, an integration is unnecessary. Additionally, according to (23), the information of the current domain of attraction is necessary to evaluate the feasibility of a transition and the subsequent value of ζ_2^+ as well, which is given by

$$\zeta_2^+(k+1) = \delta^2_{\text{zero},i\to j}(\zeta_2^+(k) - V_{\text{zero}}^{i\to j}) \tag{26}$$

For each transition, a different parameter vector $\boldsymbol{\beta}$ is learnt. An outline of the Reinforcement Learning algorithm is presented in Algorithm 1. Once the learning is concluded, actions are selected in a greedy way according to

$$\tau \leftarrow \arg\max_{\bar{\tau}} \boldsymbol{\phi}^T(\boldsymbol{s}, \bar{\tau})\boldsymbol{\beta}$$

Details regarding the convergence proof for this algorithm can be found in [13] (Ch. 3.4).

D. Reward Function for Settling Time Reduction

Assuming that the fixed point $\zeta_{2,g}^*$ corresponds to the target controller Φ_g, the **reward function** r is chosen as

$$r(\zeta_2^+, i, j) = \exp(-\lambda|\zeta_2^+(k+2) - \zeta_{2,g}^*|), \tag{27}$$

where

$$\zeta_2^+(k+1) = \delta^2_{\text{zero},i\to j}(\zeta_2^+ - V_{\text{zero}}^{i\to j}) \tag{28a}$$
$$\zeta_2^+(k+2) = \delta^2_{\text{zero},j\to g}(\zeta_2^+(k+1) - V_{\text{zero}}^{j\to g}) \tag{28b}$$

The justification behind this reward function follows from the fact that the target controller Φ_g is pre-determined. The reward function accounts for the distance between the fixed point of the target controller $\zeta_{2,g}^*$ and the value of ζ_2^+, if a transition from Φ_i to Φ_j and then to the target controller Φ_g is taken. In a few words, we are interested in how the transition from the domain of attraction of Φ_i to that of Φ_g is influenced by an intermediate transition to the domain of attraction of Φ_j. If the transition is not feasible, the reward is equal to -1.

V. Experimental Evaluation

This section presents the evaluation of the learning scheme proposed in the previous section. The parameters of the dynamic model adopted for this paper match those of the robot RABBIT [1]. The set of controllers Φ is populated by 81 periodic controllers corresponding to average desired velocities ranging from 0.7 to 1.5 m/s with a step of 0.01 m/s. Each controller is determined by an optimization procedure as described in [3] where the desired walking velocity was imposed as an equality constraint. The one-step transitions between them are precomputed and stored. The parameter λ in the reward function (27) is 0.2. Regarding the learning algorithm itself, the discount factor γ is 0.7 and the possibility of taking a random action ϵ is 30%. The learning procedure lasts for 30000 epochs, while each epoch lasts for 40 episodes. The aforementioned parameters were experimentally chosen.

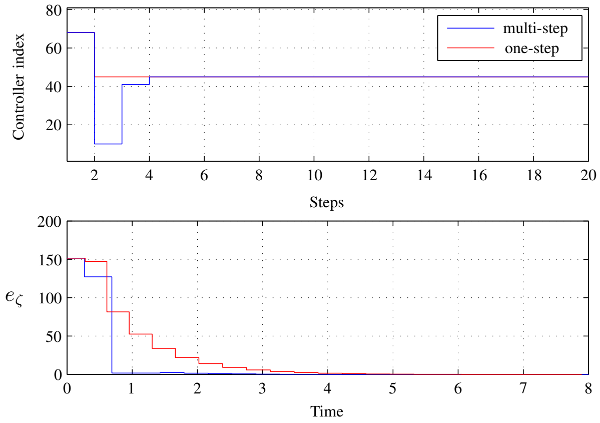

Fig. 2. The multi-step policy and the error convergence for a transition from a velocity of 1.37 m/s to that of 1.14 m/s ($\Phi_{68} \to \Phi_{45}$).

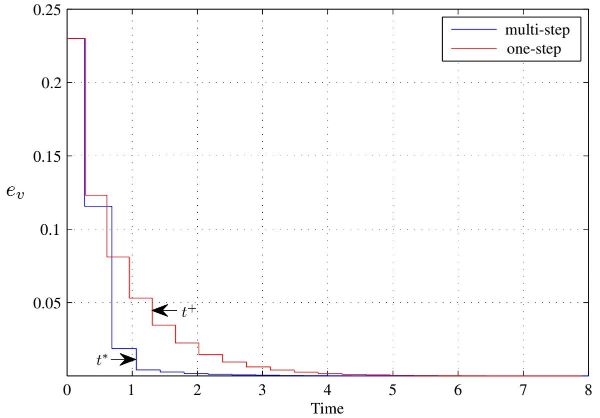

Fig. 3. The convergence of the velocity error for the transition $\Phi_{68} \to \Phi_{45}$

A. Learning a single transition

This subsection presents the experimental results obtained for learning a single transition from a velocity of 1.37 m/s to that of 1.14 m/s. As illustrated in Fig. 2, the multi-step policy tries to decelerate the robot by commanding it to walk with a much lower velocity than the target one. Then, converges to the desired one after two more steps. The validity of our approach can also be justified by plotting the convergence of the velocity error $e_v = |\bar{v} - v_{\text{target}}|$, where $\bar{v}$ is the average velocity of the robot during a step and v_{target} is the target velocity. As shown in Fig. 3, there is a high correlation between the way the velocity and ζ_2^+ converge to their desired values.

The superiority of the multi-step transition is clearly highlighted when the error $e_\zeta = |\zeta^+ - \zeta^*|$ is taken into account. The multi-step policy enters the domain of attraction of Φ_{45} at time $t^* \approx 1.1$ s (see Fig. 3) with an error $e_\zeta = 1.85$, while at time $t^+ \approx 1.3$ s (see Fig. 3) the one-step transition has an error $e_\zeta = 34$. The error e_v following the multi-step policy becomes negligible in approximately 2.2 s, while for the one-step it requires approximately 3.8 s.

Algorithm 1 Q-Learning with linear parametrization

```
for all transitions $\Phi_i \rightarrow \Phi_j$ do
    for all epochs do
        Initialize learning rate $\eta_0$
        Initialize $\epsilon$
        Initialize state $\boldsymbol{s}_0 = \left[\zeta^*_{2,i}, i\right]$
        Initialize randomly the parameter vector $\boldsymbol{\beta}$
        for all episodes do
            $\tau_k \leftarrow \begin{cases} \text{uniform random action in } T \text{ with probability } \epsilon \\ \arg\max_{\bar{\tau}} \boldsymbol{\phi}^T(\boldsymbol{s}_k, \bar{\tau})\boldsymbol{\beta}_k \text{ with probability } 1-\epsilon \end{cases}$
            Apply $\tau_k$, measure $\boldsymbol{s}_{k+1}$ and reward $r_{k+1}$
            $\boldsymbol{\beta}_{k+1} \leftarrow \boldsymbol{\beta}_k + \eta_k \left[ r_{k+1} + \gamma \max_{\tau'}(\boldsymbol{\phi}^T(\boldsymbol{s}_{k+1}, \tau')\boldsymbol{\beta}_k) - \boldsymbol{\phi}^T(\boldsymbol{s}_k, \tau_k)\boldsymbol{\beta}_k \right] \boldsymbol{\phi}(\boldsymbol{s}_k, \tau_k)$
            Reduce learning rate $\eta$ and $\epsilon$
        end for
    end for
end for
```

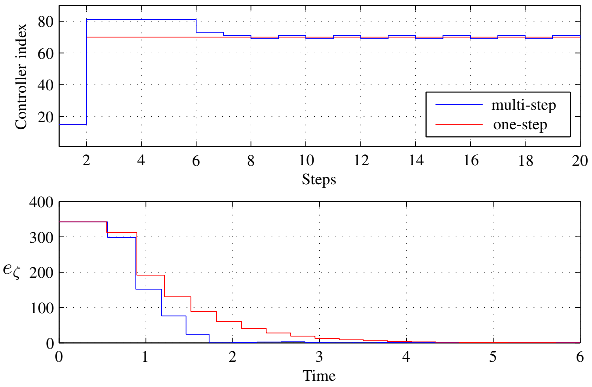

Fig. 4. An exemplary oscillating policy. The desired transition is from 0.84 m/s to 1.39 m/s.

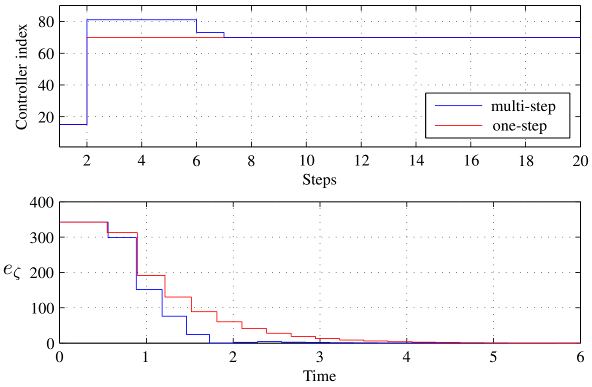

Fig. 5. The solution to the problem of the non-stationary policy. When the pattern 71 → 69 → 71 is discovered starting at the sixth step, the policy is fixed to controller Φ_{70}.

B. Dealing with non-stationary policies

When it comes to Reinforcement Learning, one might end up dealing with policies that oscillate around the desired terminal state or in our case the controller index that corresponds to the desired target velocity. Once the policies for all the transitions are learnt, a post-processing procedure is initiated to detect such repeating patterns. An example of such a policy can be seen in Fig. 4 for a transition from a velocity of 0.84 m/s to that of 1.39 m/s. As shown there, the policy utilizes the controller which gives the largest velocity and then gradually decelerates the robot to the desired velocity, but does not eventually reach it, rather it oscillates between 1.38 m/s and 1.40 m/s. Since the target velocity is known, once these policies are detected, the pattern can be removed by fixing the action τ to the controller corresponding to the target velocity as illustrated in Fig. 5.

C. Overall performance

When evaluated on all possible transitions $\Phi_i \rightarrow \Phi_j$, $i \neq j$, the proposed methodology has a success rate of 84.34%, meaning that 84.34% of the overall transitions are performed faster with this framework. For the remaining 15.66% of the transitions, the one-step approach can be utilized, since it is known that it will perform better. There is always the possibility to fine tune the policies that perform worse than the one-step approach, but it is desired to have a uniform framework. Fig. 6 presents the overall performance of the proposed methodology.

Finally, Fig. 7 gives a "heat" map, which shows how much better the proposed methodology can perform in comparison to the one-step approach. The "heat" corresponds to the value $\Delta e_\zeta = |\zeta_2^+(t^*) - \zeta_2^*| - |\zeta_2^+(t^+) - \zeta_2^*|$.

It is evident that the proposed methodology cannot outperform the one-step approach in cases where a transition is taken between "neighbouring" controllers, i.e. transitions close to the secondary diagonal. For these cases, a sequence of controllers is not expected to offer much, since the fixed points of these controllers are close with each other. On the other hand, learning the one-step transition for these cases depends strongly on the randomly selected actions at the beginning of each epoch.

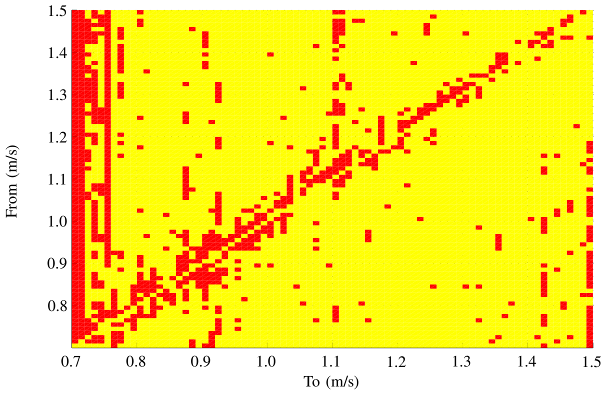

Fig. 6. The overall score for the proposed methodology. Yellow denotes that the multi-step policy performs better than the one-step approach, while red suggests the opposite. The periodic transitions (secondary diagonal) are not taken into account.

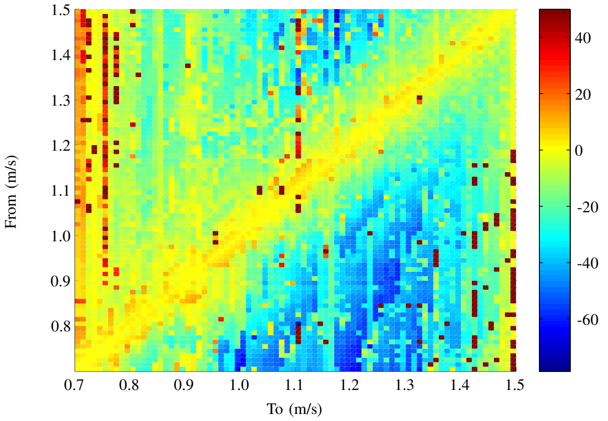

Fig. 7. A "heat" map showing how much better the proposed methodology performs in comparison to the one-step approach. The evaluation criterion is Δe_ζ.

The second big class where the one-step approach has better performance, comprises transitions where the target controller index is close to the limits (1 or 81). In that case, a multi-step transition cannot perform better than a one-step transition, since a possible deceleration or acceleration below and above the target velocity is not possible.

VI. Conclusion

This paper proposes a methodology for reducing the settling time of transitions between periodic controllers. This framework is applicable to any walking machine with one degree of underactuation since it utilizes the Hybrid Zero Dynamics of the system. The problem is expressed as a Markov Decision Process and solved with Reinforcement Learning. Using the Hybrid Zero Dynamics assists both in simplifying the controller design and reducing the state representation for the Reinforcement Learning formulation. The experimental results demonstrate that the proposed framework can perform better for 84.34% of 6480 transitions for a biped walking robot matching the parameters of RABBIT. In the future the utilization of different cost functions will be investigated in order not only to increase the success rate but also to achieve better error differences.

VII. Acknowledgement

This work is supported in part within the ERC Advanced Grant SHRINE Agreement No. 267877 (www.shrine-project.eu) and in part by the Technische Universität München - Institute for Advanced Study (www.tum-ias.de), funded by the German Excellence Initiative.

References

[1] C. Chevallereau, G. Abba, Y. Aoustin, F. Plestan, E. Westervelt, C. Canudas-de Wit, and J. Grizzle, "RABBIT: a testbed for advanced control theory," *IEEE Control Systems Magazine*, pp. 57–79, 2003.

[2] K. D. Mombaur, R. W. Longman, H. G. Bock, and J. P. Schloeder, "Open-loop stable running," *Robotica*, vol. 23, pp. 21–33, 2005.

[3] E. Westervelt, J. Grizzle, and D. Koditschek, "Hybrid zero dynamics of planar biped walkers," *IEEE Transactions on Automatic Control*, vol. 48, no. 1, pp. 42–56, 2003.

[4] R. Tedrake, I. R. Manchester, M. Tobenkin, and J. W. Roberts, "LQR-trees: Feedback Motion Planning via Sums-of-Squares Verification," *The International Journal of Robotics Research*, vol. 29, pp. 1038–1052, 2010.

[5] R. Burridge, A. Rizzi, and D. Koditschek, "Sequential composition of dynamically dexterous robot behaviors," *The International Journal of Robotics Research*, vol. 18, pp. 534–555, 1999.

[6] E. Najafi, G. Lopes, and R. Babuska, "Reinforcement learning for sequential composition control," in *IEEE 52nd Annual Conference on Decision and Control*, 2013, pp. 7265–7270.

[7] B. Buss, A. Ramezani, K. Hamed, B. Griffin, K. Galloway, and J. Grizzle, "Preliminary walking experiments with underactuated 3D bipedal robot MARLO," in *IEEE International Conference on Intelligent Robots and Systems*, Sept 2014, pp. 2529–2536.

[8] D. Djoudi, C. Chevallereau, and Y. Aoustin, "Optimal reference motions for walking of a biped robot," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2005, pp. 2002–2007.

[9] A. Shiriaev, J. Perram, and C. Canudas-de Wit, "Constructive tool for orbital stabilization of underactuated nonlinear systems: Virtual constraints approach," *IEEE Transactions on Automatic Control*, vol. 50, no. 8, pp. 1164–1176, 2005.

[10] E. Westervelt, J. Grizzle, C. Chevallereau, J. H. Choi, and B. Morris, *Feedback control of dynamic bipedal robot locomotion*, ser. Control and automation. Boca Raton: CRC Press, 2007.

[11] R. S. Sutton and A. G. Barto, *Introduction to Reinforcement Learning*, 1st ed. Cambridge, MA, USA: MIT Press, 1998.

[12] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," *The International Journal of Robotics Research*, vol. 32, pp. 1238–1274, 2013.

[13] L. Busoniu, R. Babuska, B. De Schutter, and D. Ernst, *Reinforcement learning and dynamic programming using function approximators*. CRC press, 2010.