Harmonic RL: A Frequency-Domain Approach to Reinforcement Learning with Application to Active Knee Prosthesis

Can Çetindağ, Robert D. McAllister, and Peyman Mohajerin Esfahani

Abstract— **We propose a frequency-domain state representation to improve the performance and reduce the computation and data requirements of reinforcement learning. This approach is tailored to tracking tasks of periodic trajectories. We apply the proposed methodology to an active knee prosthesis application. Using the high-fidelity simulator MuJoCo, we demonstrate significant performance improvements (in terms of Bellman error) for the proposed frequency-domain state representation relative to the current state-of-the-art time-domain state representation used in these applications.**

I. Introduction

Feedback control algorithms such as reinforcement learning (RL), determine control actions based on the current state of the system. This state is typically constructed via a sufficient number of previous measurements such that this state and system satisfy the Markov property. RL is particularly attractive for many engineering applications as it does not require an explicit dynamical model of the system, we can apply RL using only data generated from both real-world experiments and high-fidelity simulators. While we ideally want to include all available measurements in the state, the amount of data and computation time required to learn a suitable RL policy scales poorly with the dimension of the state. Thus, an efficient and reduced order representation of this state is important to successfully deploy these algorithms in embedded (robotics) applications.

In this work, we consider a specific class of periodic trajectory tracking problems that is particularly relevant to control of active knee prostheses. In these problems, we can select parameters for a lower level controller at each step. In active knee prosthesis applications, this lower level controller corresponds to an impedance controller (IC) with three IC parameters [1]. The system then produces a trajectory of measurements that depends on these parameters and the state of the system. The goal is to design a policy to select these parameters at each step such that the trajectory of measurements at each step follows a specified periodic reference. Since there are many measurements taken within this trajectory, we require a lower dimensional representation of these states based on some set of key features from this trajectory. In machine learning, this procedure is sometimes called feature extraction, but we use the term state representation for consistency with control theory.

A standard practice is to normalize these IC parameters according to body proportions and keep these parameters constant for a particular user. However, this overlooks the interpersonal and intra-personal variation of walking patterns. RL has been proposed as a natural method to address these limitations [2]. In particular, RL allows us to determine a suitable policy for adjusting these parameters in real-time that is tailored to each individual. Previous studies represented the state of the system using time-domain characteristics [2]–[7]. This approach is based on the extrema of the trajectory in the time domain. The state is then defined as the deviations between the observed and target extrema in terms of both time and magnitude.

However, a major drawback of this approach is that it discards all the information between the extrema. To address this drawback, this study proposes a *frequency-domain state representation* for RL, i.e. Harmonic RL. Specifically, we can approximate these trajectories via a truncated Fourier series. With this approach, the information between the peaks can be preserved by representing the state as the difference between the observed and target Fourier series coefficients. Fourier series have been used previously in RL to approximate the value function [8] and more recently have been applied as a pre-processing step in deep RL [9], [10]. To the best of our knowledge, using Fourier approximations to represent the state of the system for RL is a novel contribution.

To train and test the influence of harmonic state representation, a custom environment is created using a physics simulation software called *MuJoCo* [11]. MuJoCo was chosen for this study of active knee control because this software includes the ground reaction forces (GRF), which are crucial in human walking. The following summarizes the contributions of this study:

- **Methodology:** We propose a frequency-domain state representation for RL applied to periodic trajectory following tasks. We justify the use of this representation via the relationship between the Fourier coefficients and the original objective of periodic trajectory tracking.
- **Application:** We demonstrate the efficacy of this frequency-domain state representation via an active knee prosthesis application. Using a custom multi-body human gait simulation that includes GRF (built with MuJoCo) we demonstrate significantly better performance of the proposed frequency-domain approach relative to the previously studied time-domain approach.

The paper is structured as follows. First, Section II lays out the problem and the proposed frequency-domain state representations. Section III describes a standard RL algorithm that is used in this study (Q-learning). Finally, Section IV

This research is partially supported by the European Research Council (ERC) under grant TRUST-949796.

The authors are with the Delft Center for Systems and Control, Delft University of Technology, Delft, The Netherlands.

discusses the custom simulation environment for the active knee prosthesis application and demonstrates the benefits of the proposed approach using this simulator.

II. Problem Setting and State Representation

We consider a dynamical system in which we measure a (scalar) variable θ at T equally spaced subintervals $\delta := 1/T$. We collect these measurements at each time step $t \in \mathbb{Z} := \{0, 1, \dots\}$ in the vector x_t defined as

$$x_t := \begin{bmatrix} \theta(t) & \theta(t-\delta) & \dots & \theta(t-(T-1)\delta) \end{bmatrix}^\top$$

We are also provided with a target trajectory for θ within this subinterval:

$$\overline{x} = \begin{bmatrix} \overline{\theta}(1) & \overline{\theta}(1-\delta) & \dots & \overline{\theta}(1-(T-1)\delta) \end{bmatrix}^\top$$

We assume that these T measurements within a time interval $[t-1, t]$ contain sufficient information to define a Markov decision process with a manipulated input/action $u_t \in \mathcal{U} \subseteq \mathbb{R}^m$ and random variable $w_t \in \mathbb{R}^q$. Formally speaking, we denote the dynamics by

$$x_{t+1} = f_x(x_t, u_t, w_t) \tag{1}$$

The following are some specific characteristics of our problem, motivating the proposed harmonics solution approach:

(i) The measurements θ is periodic within the interval $[t-1, t]$, and the target trajectory $\overline{\theta}$ satisfies $\overline{\theta}(t-1) = \overline{\theta}(t)$.

(ii) At each time step $t \in \mathbb{Z}$, we can select an input u_t that is then constant across the next interval $[t, t+1)$.

(iii) The number of measurements T is much larger than the internal state dimension, and as such, it is reasonable to assume that there is a Markovian property between the pair (x_t, u_t) and the next state x_{t+1}.

This problem setting is particularly relevant in hierarchical control schemes, in which the input u_t defines parameters for a low-level control system and the goal is to track a periodic trajectory. One such application is an active knee prosthesis discussed in Section IV.

We define the squared difference between θ and $\overline{\theta}$ on the interval $[t-1, t]$ as

$$\|\theta - \overline{\theta}\|^2_{[t-1,t]} := \sum_{i=0}^{T-1} \left(\theta(t-\delta i) - \overline{\theta}(1-\delta i)\right)^2 = \|x_t - \overline{x}\|^2$$

The goal is to select u_t to minimize the squared distance between θ and $\overline{\theta}$ with minimal control effort. We define this performance via the stage cost:

$$g_x(x, u) = \frac{1}{T}\|x - \overline{x}\|^2 + u^\top R_u u \tag{2}$$

in which R_u is a positive definite matrix that weighs the cost of different inputs. Note that $1/T$ is used to normalize the cost by the number of measurements. In the ideal case, we want to find a policy $u_t = \pi_x(x_t)$ to minimize the expected value of a discounted sum of stage costs defined as

$$\min_{\pi_x} \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k g_x(x_k, u_k) \;\middle|\; u_k = \pi_x(x_k) \text{ and } (1)\right] \tag{3}$$

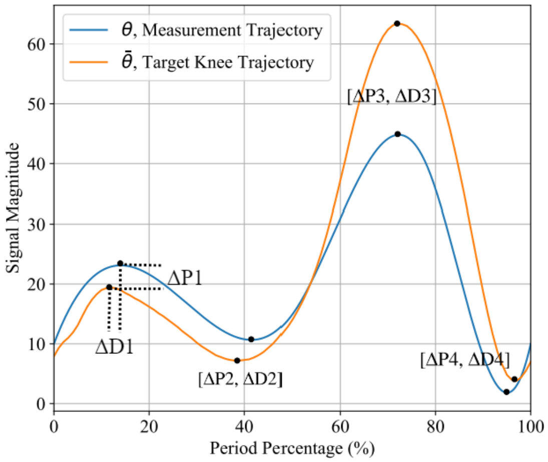

Fig. 1. Plot of θ and $\hat{\theta}$ indicating the difference in extrema that are used to define s in the time-domain feature extraction.

in which $\gamma \in (0, 1)$ is the discount factor, and $\mathbb{E}[\cdot]$ denotes expected value with respect to the random variables w_k.

The problem is that T can be very large and therefore solving (3) via, e.g., reinforcement learning (RL) or approximate dynamical programming, is very difficult. Instead, a more efficient representation of the state, defined based on key features of θ, may be sufficient to approximately define a Markov decision process and solve (3). In other words, we want a map $s_t = \phi(x_t)$ from $x_t \in \mathbb{R}^T$ to a low dimensional vector of features $s_t \in \mathbb{R}^n$ with $n << T$ that is sufficiently informative to define the dynamics

$$s_{t+1} \approx f(s_t, u_t, w_t) \tag{4}$$

in which $f(\cdot)$ is the state transition function for s_t. We also require features that are sufficiently informative to define a new stage cost $g(s, u)$ that is approximately equal to the original stage cost:

$$g_x(x, u) \approx g(\phi(x), u) \tag{5}$$

Subsequently, we discuss two approaches to define $\phi(\cdot)$ and therefore s_t. The first defines s_t via time-domain features of x_t such as extrema and is used in the application discussed in Section IV. The second is the frequency-domain approach proposed in this paper that leverages periodicity to define s_t as the Fourier coefficients of θ.

A. Time-domain state representation via extrema

One common approach to define s is to use the extrema of θ on the interval $(t-1, t]$. The difference between the observed extremum and the target extremum is encoded through the difference in value (ΔP) and location (ΔD) as shown in Figure 1. The resulting state representation is

$$s = \left[\Delta P_1, \Delta D_1, \dots, \Delta P_4, \Delta D_4\right]^T \tag{6}$$

With this state representation, the stage cost is

$$g(s, u) = s^\top R_s s + u^\top R_u u \tag{7}$$

in which the matrix $R_s \succ 0$ is used as a tuning parameter.

There are, however, a few shortcomings of this approach. First, the relevant extrema in the reference trajectory must be determined manually in the controller design step based on the application of interest. For more complicated trajectories, this step may prove difficult. Second, and more importantly, this representation ignores the behavior of the system between these extrema. Thus, there is no direct connection between the stage cost in (7) and the original stage cost $g_x(x,u)$. Instead, we are required to tune R_s until we achieve the desired controller behavior.

B. *Frequency-domain state representation via Fourier series*

Since we are considering problems in which θ and $\overline{\theta}$ are smooth and periodic, we can instead represent these trajectories via a linear combination of harmonics. Specifically, we can represent these trajectories via a (truncated) Fourier series with N harmonics:

$$\theta_N(\zeta) = \frac{1}{2}a_0 + \sum_{n=1}^{N} a_n \cos(\omega_n \zeta) + b_n \sin(\omega_n \zeta) \tag{8}$$

in which $\zeta \in [0,1]$ is a general dummy variable, $\omega_n = 2\pi n$, and the pair (a_n, b_n) are the Fourier series coefficients.[1] As demonstrated in Figure 2, periodic trajectories of interest can often be represented with only a few harmonics. In this case, $N = 6$ is sufficient. Thus, for each interval $[t-1, t]$, we have the approximations

$$\theta(t-1+\zeta) \approx \theta_{t,N}(\zeta) \qquad \overline{\theta}(\zeta) \approx \overline{\theta}_N(\zeta)$$

For the state x_t, let a_0 and $(a_n, b_n)_{n=1}^N$, denote the first N coefficients approximating the periodic signal θ from $[t-1, t]$. Let $\overline{a}_0$ and $(\overline{a}_n, \overline{b}_n)_{n=1}^N$ denote the first N coefficients from the periodic reference signal $\overline{\theta}$ from $[0,1]$. We define s as the difference between these coefficients such that

$$s = \phi(x) = \left[\Delta a_0,\ \Delta a_1,\ \Delta b_1,\ \ldots,\ \Delta a_{N-1},\ \Delta b_{N-1}\right]^\top$$

in which $\Delta a_n = a_n - \overline{a}_n$ and $\Delta b_n = b_n - \overline{b}_n$. Note that the time-domain approach in II-A requires that the relevant features are tailored to the problem of interest. However, in the frequency domain approach, we need to choose only the number of harmonics N. Thus, the frequency domain approach is easier to generalize across various applications.

Moreover, representing the state of the system via Fourier coefficients allows us to provide a straightforward approximation of the cost function in (2). We note that for sufficiently large N:

$$\|x_t - \overline{x}\|^2 = \|\theta - \overline{\theta}\|^2_{[t-1,t]} \approx \|\theta_{t,N} - \overline{\theta}_N\|^2_{[0,1]}$$

in which

$$\|\theta_{t,N} - \overline{\theta}_N\|^2_{[0,1]} = \sum_{i=0}^{T-1} \left(\theta_{t,N}(i/T) - \overline{\theta}_N(i/T)\right)^2$$

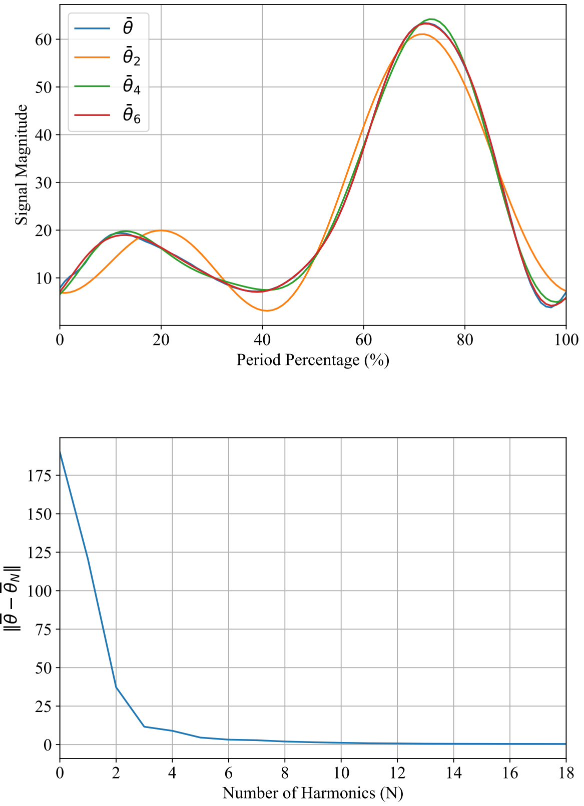

Fig. 2. Performance of the Fourier Series to approximate a smooth and periodic trajectory with the increasing number of harmonics. Top: trajectories, bottom: Approximation error $\|\overline{\theta} - \overline{\theta}_N\|_{[0,1]}$

For sufficiently large T, we can approximate this summation as the integral

$$\frac{1}{T}\|\theta_{t,N} - \overline{\theta}_N\|^2_{[0,1]} \approx \int_0^1 \left(\theta_{t,N}(\zeta) - \overline{\theta}_N(\zeta)\right)^2 d\zeta$$

From the orthogonality of the Fourier series, we have

$$\int_0^1 \left(\theta_{t,N}(\zeta) - \overline{\theta}_N(\zeta)\right)^2 d\zeta = \frac{1}{2}(a_n - \overline{a_n})^2 + \frac{1}{2}\sum_{n=1}^{N}\left((a_n - \overline{a_n})^2 + (b_n - \overline{b_n})^2\right)$$

Therefore, we have $\frac{1}{T}\|x - \overline{x}\|^2 \approx \frac{1}{2}\|s\|^2$. We thus have an approximate relationship between the norm of the frequency-domain features (s) and the difference between the realized and target trajectories for θ. Hence, we define

$$g(s,u) = \frac{1}{2}\|s\|^2 + u^\top R_u u \tag{9}$$

and note that this definition satisfies (5). With this harmonic state representation, there is now a direct connection between the stage cost in (9) and the original stage cost $g_x(\cdot)$.

[1] Note that b_0 is irrelevant to $\theta_N(\zeta)$ and ignored in subsequent discussion.

III. Q-LEARNING ALGORITHM

In this section, we briefly review the standard discounted Q-learning algorithm [12], [13], which is applied to train the Q-function in the setting of our proposed harmonic state representation. We define the Q-function

$$Q^*(s_0, u_0) := \min_{\pi} g(s_0, u_0) + \qquad (10)$$
$$\mathbb{E}\Big[\sum_{k\geq 1} \gamma^k g(s_k, u_k) \;\Big|\; u_k = \pi(s_k) \text{ and } (4)\Big]$$

Note that (10) uses the features s instead of the complete state in (3). The classical equivalent characterization of the Q-function defined in (10) is the so-called Bellman equation [14]:

$$Q^*(s, u) = g(s, u) + \gamma\, \mathbb{E}\Big[\min_{u^+\in\mathcal{U}} Q^*\big(f(s, u, w), u^+\big)\Big] \quad (11)$$

This implicit equation needs to hold for all the state-action pairs (s, u). The optimal policy π^* in the definition of Q-function Q^* in (10) can be computed via

$$\pi^*(s) = \arg\min_{u\in\mathcal{U}} Q^*(s, u). \qquad (12)$$

The alternative Bellman characterization (11) is the starting point to compute Q^* numerically. The implicit equation (11) often cannot be solved exactly for two reasons: (i) The function Q^* is an infinite dimensional object while for numerical purposes we can only restrict our search space to a finitely many parameterized approximation, and (ii) the Bellman equality (11) should (approximately) hold for all (infinitely many) pairs (s, u) whereas we can only ensure this for finitely many pairs (s_k, u_k), often in the form of an offline trajectory of the system such as

$$\mathcal{D} = \{s_k, u_k, s_k^+\}_{k=0}^{K-1}, \quad \text{where} \quad s_k^+ = f(s_k, u_k, w_k).$$

With these two limitations in mind and given the above offline dataset $\mathcal{D}$, a common practice to restrict the Q-function to a finitely parameterized function $\widehat{Q}_r$ with the parameter $r \in \mathbb{R}^n$ and approximate the Bellman equation (11) via the so-called projected Bellman equation [15]:

$$r^* = \arg\min_{r\in\mathbb{R}^n} \sum_{k=0}^{K-1} \Big(\widehat{Q}_r(s_k, u_k) - g(s_k, u_k) \qquad (13)$$
$$- \gamma \min_{u^+\in\mathcal{U}} \widehat{Q}_{r^*}(s_k^+, u^+)\Big)^2$$

A common approximation class is when the approximate Q-function is linearly parameterized as

$$\widehat{Q}_r(s, u) = \langle \Psi(s, u), r\rangle, \qquad (14)$$

where the vector $\Psi(s, u) = [\psi^{(1)}(s, u), \ldots, \psi^{(n)}(s, u)]$ is the collection of n basis functions, $r \in \mathbb{R}^n$ is the approximation parameters to be learned, and $\langle\cdot,\cdot\rangle$ is the conventional inner product between two $\mathbb{R}^n$ vectors. Note that considering the linear approximation (14) makes the right-hand side of the projected Bellman equation (13) a quadratic optimization in the parameter r. A simple iterative algorithm converging to the solution r^* in (13) can be obtained by applying the gradient descent with respect to r on the right-hand side of (13) and updating the parameter r on the left-hand side. This approach leads to the update equation

$$r^{(i+1)} = r^{(i)} - \eta^{(i)} \sum_{k=0}^{K-1} \Big(\langle\Psi_k, r^{(i)}\rangle - g_k \qquad (15)$$
$$- \gamma \min_{u^+\in\mathcal{U}} \langle\Psi(s_k^+, u^+), r^{(i)}\rangle\Big)\Psi_k$$

where we use the shorthand notation $\Psi_k = \Psi(s_k, u_k)$ and $g_k = g(s_k, u_k)$, and the scalar $\eta^{(i)}$ is the stepsize of the gradient descent which is typically set proportion to $1/i$ to ensure the convergence of the algorithm. It is worth noting that the algorithm (15) can be accelerated using momentum techniques from the optimization algorithm literature [16], which proves to be effective in our numerical experiments:

$$r^{(i+1)} = r^{(i)} - \eta^{(i)} \sum_{k=0}^{K-1} \Big(\langle\Psi_k, r^{(i)}\rangle - g_k \qquad (16)$$
$$- \gamma \min_{u^+\in\mathcal{U}} \langle\Psi(s_k^+, u^+), r^{(i)}\rangle\Big)\Psi_k$$
$$- \mu\big(r^{(i)} - r^{(i-1)}\big)$$

IV. KNEE PROSTHESIS APPLICATION AND RESULTS

As a proof of concept application, active knee prosthesis control is chosen as the case study to assess the effectiveness of the harmonic state representations. We first introduce the domain and then apply the algorithm in Section III.

A. Introduction to Active Knee Prosthesis Control

A motorized prosthetic knee is typically controlled using impedance control (IC) [1].

$$\tau = -K(\theta - (\theta_s)) - C\dot{\theta} \qquad (17)$$

where K is the stiffness coefficient, θ_s is the set-point angle and C is the damping coefficient. Together, they constitute the *IC parameters*, which are the main concern of this application. The variables θ and $\dot{\theta}$ are the angular position and velocity of the joint, respectively. The torque applied by the motor is denoted by τ.

The personalization of the IC control is an open problem in the active prosthesis domain. One approach is to normalize parameters according to body proportions, but this approach overlooks two real-life aspects. The variation between two people with similar body proportions (inter-personal variations) and the variation between two steps of an individual (intra-personal variations). This study aims to achieve adaptive and personalized IC parameters via RL.

Within the scope of this study, we focus on the *Swing Extension (SWE)* phase of the level ground walking, but the study can be expanded to every phase. SWE encapsulates the part of the walking from full flexion of the knee to full extension. Recall the characteristics listed in Section II. The first characteristic is satisfied as the objective is to recreate healthy human knee trajectory [17], which is smooth and periodic by nature. The proposed system is a discrete system, where the adjustments on the IC parameters are taken just

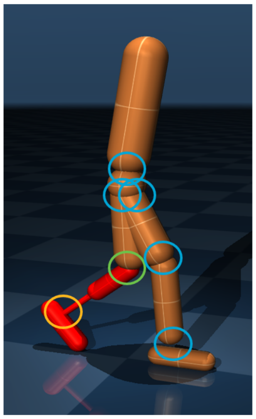

Fig. 3. The custom model created in MuJoCo. Blue circles depict the five healthy side joints, yellow circle depicts the passive ankle prosthesis, and green circle depicts the knee prosthesis aimed to control.

before the start of the SWE phase, and they are applied only throughout the following SWE phase. This fact satisfies the second characteristic. The third characteristic is assumed to be satisfied given the large sampling rate of θ, that is $200Hz$.

Level ground walking is a highly variant activity, even across able-bodied people. Typically, this variety is even more for amputees. Thus, RL is a strong candidate for learning user-specific system dynamics (f) and acting accordingly. However, one must first create a suitable environment to train and evaluate an RL algorithm.

B. Reinforcement Learning Environment

An environment dedicated to running the algorithm is one of the most fundamental components of a RL project. This environment is created by *Multi-Joint dynamics with Contact (MuJoCo)* [18], which is a physics engine that primarily focuses on multibody dynamics and accommodates most of the benchmarking environments in the field of RL [19]. One should note that this environment does not aim to perfectly replicate the walking motion but to create an environment that can be enhanced by training an RL agent.

The model (Figure 3) that this thesis utilized is a modified version of the 17 DoF Humanoid model of Gym [19], which is one of the benchmarking environments for RL applications. The Humanoid model was reduced to a 7-link model, which includes feet, shanks, thighs, and HAT (combined entity of head-arms-trunk). The proportions of different body parts are adjusted according to the average male body proportions given in [20]. All the model joints are controlled with the IC Law given in Equation 17, with the proper constraints on different types of joints (healthy joints, active knee prosthesis, passive ankle prosthesis).

Ground reaction forces (GRF) are one of the most important aspects of walking. Including GRF in the simulation is crucial to prove the approach has reciprocity in the real-life application. One of the significant contributions of this thesis is the customized RL environment that includes GRF. This was possible due to the strong contact dynamics capabilities provided by MuJoCo.

TABLE I

HYPER-PARAMETERS USED IN EXPERIMENTS.

Noise Standard Deviation	σ	0.01
Number of Harmonics	N	6
Discount Factor	γ	0.7
Learning Rate	η	0.7
Momentum Rate	μ	0.9
Batch Size	K_b	100
Target Samples	N_s	8000

The simulation environment is enhanced by a zero-mean Gaussian noise with standard deviation σ. In real-world scenarios, the training of the agent is expected to be richer in uncertainty. Thus, the investigation for non-Gaussian noise is left to clinical studies.

C. Results

In the context of this custom environment, the state of the model can be described as a combination of all the described joints. However, by the nature of the problem, in real life, the only accessible information is on the prosthetic knee. Thus, the approximate Markov state s motivated in Section II only contains the information of the prosthetic knee. Two sets of experiments are conducted to compare the performance of the two different state representation approaches (ϕ): time-domain and frequency-domain. To investigate the performance of a policy, we consider the average Bellman error (BE), which is the objective of the right-hand side of (13) evaluated on our offline dataset $\mathcal{D}$, i.e.,

$$\overline{BE} := \frac{1}{K}\sum_{k=0}^{K-1}\left(\langle\Psi_k, r^{(i)}\rangle - g_k - \gamma \min_{u^+\in\mathcal{U}}\langle\Psi(s_k^+, u^+), r^{(i)}\rangle\right)^2$$

The initial weight vector is the "naive" controller that has no adjustments on the IC parameters. With that in mind, the percentage improvement throughout the training is thus described by

$$\Delta\overline{BE}(\%) := 100 \times \frac{\overline{BE}_i - \overline{BE}_f}{\overline{BE}_i}, \tag{18}$$

where $\Delta\overline{BE}(\%)$ is the percentage change, which indicates improvement when it is positive. The variables $\overline{BE}_i$ and $\overline{BE}_f$ are the initial and final bellman error values, respectively. The hyper-parameters of the two experiments are kept the same under different state representations. These key hyper-parameters are reported in Table I.

Different state representations result in differences in the scale of the average bellman error. For a better comparison, the average bellman error is normalized for both experiments. The evolution of $\overline{BE}$ is plotted in Figure 4.

Table II presents the percentage performance improvement for both setups. The significant improvement by the harmonic state representation can be seen clearly in both Figure 4 and Table II. This suggests that frequency-domain features encode the trajectory information better and provide a more efficient learning process.

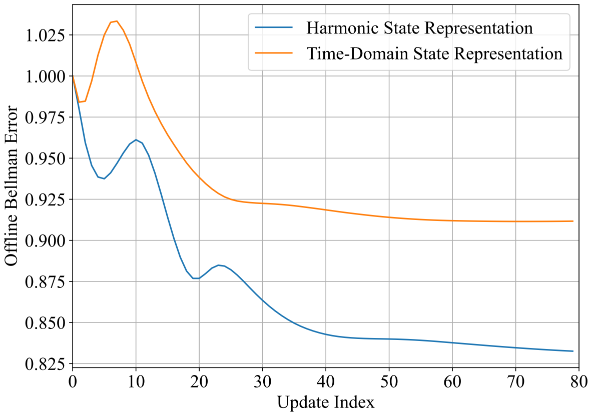

Fig. 4. Normalized average bellman error harmonic (blue) and time-domain (red) state representations.

Increasing the richness of the Q-function parameterization (e.g., with neural networks) may improve the controller performance. However, limited computational power on the prosthetics and critical response time should be taken into consideration when increasing the complexity of this parameterization.

Through simulations, the proposed RL framework has proven to be a potential solution to achieve a personal and adaptive active knee prosthesis controller by targeting the healthy human knee trajectory. However, amputee walking and able-bodied walking do not have one-to-one correspondence. Thus, the real-life improvement of the algorithm still needs to be investigated through clinical studies. A possible future direction can be extracting personal target trajectories through a musculoskeletal simulation. Having such a tool would also pave the way to use this algorithm on other ambulation modes such as ramp ascent or stair ascent.

Despite the promising performance results in the MuJoCo simulation environment, it should be noted that the algorithm converges in around 40 iterations with a batch size of 100 samples, which is approximately equivalent to 4000 steps. In real life, asking an amputee to walk 4000 steps (around 70 mins) is not realistic. Thus, the migration of this study to an actual controller should be done considering such clinical constraints. Realize that these experiments are focused on training the agents from scratch. One approach to increase the convergence speed is using a pre-trained agent as the starting point. This can be achieved by training an agent using a pool of data collected from amputees with diverse profiles.

Algorithm's real-world performance should be evaluated through controlled experiments. A recommended approach would be to conduct daily comparisons, with and without the algorithm, while maintaining a blind testing condition to prevent bias. This methodology will ensure that the impact of the algorithm is accurately assessed in practical scenarios.

TABLE II

SWE BY RL FRAMEWORK USING DIFFERENT STATE REPRESENTATIONS.

	Percentage Improvement	**Convergence**
Frequency Domain	15.70%	40 iterations
Time Domain	6.16%	30 iterations

References

[1] N. Hogan, "Impedance control: An approach to manipulation: Part ii—implementation," 1985.

[2] Y. Wen, X. Gao, J. Si, A. Brandt, M. Li, and H. Huang, "Robotic knee prosthesis real-time control using reinforcement learning with human in the loop," *Communications in Computer and Information Science*, vol. 1005, pp. 463–473, 2019. cited By 3.

[3] M. Li, X. Gao, Y. Wen, J. Si, and H. H. Huang, "Offline policy iteration based reinforcement learning controller for online robotic knee prosthesis parameter tuning," in *2019 International conference on robotics and automation (ICRA)*, pp. 2831–2837, IEEE, 2019.

[4] X. Gao, J. Si, Y. Wen, M. Li, and H. Huang, "Reinforcement learning control of robotic knee with human-in-the-loop by flexible policy iteration," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 10, pp. 5873–5887, 2021.

[5] Y. Wen, J. Si, A. Brandt, X. Gao, and H. Huang, "Online reinforcement learning control for the personalization of a robotic knee prosthesis," *IEEE Transactions on Cybernetics*, vol. 50, no. 6, pp. 2346–2356, 2020. cited By 68.

[6] M. Li, Y. Wen, X. Gao, J. Si, and H. Huang, "Toward expedited impedance tuning of a robotic prosthesis for personalized gait assistance by reinforcement learning control," *IEEE Transactions on Robotics*, vol. 38, no. 1, pp. 407–420, 2022. cited By 11.

[7] W. Liu, R. Wu, J. Si, and H. Huang, "A new robotic knee impedance control parameter optimization method facilitated by inverse reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 10882–10889, 2022.

[8] G. Konidaris, S. Osentoski, and P. Thomas, "Value function approximation in reinforcement learning using the fourier basis," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 25, pp. 380–385, 2011.

[9] D. Brellmann, D. Filliat, and G. Frehse, "Fourier features in reinforcement learning with neural networks," *Transactions on Machine Learning Research*, 2023.

[10] A. Li and D. Pathak, "Functional regularization for reinforcement learning via learned fourier features," *Advances in Neural Information Processing Systems*, vol. 34, pp. 19046–19055, 2021.

[11] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033, 2012.

[12] D. Bertsekas and J. N. Tsitsiklis, *Neuro-dynamic programming*. Athena Scientific, 1996.

[13] D. Bertsekas, *Abstract dynamic programming*. Athena Scientific, 2022.

[14] R. Bellman, "A markovian decision process," *Journal of mathematics and mechanics*, pp. 679–684, 1957.

[15] D. P. Bertsekas, "Temporal difference methods for general projected equations," *IEEE Transactions on Automatic Control*, vol. 56, no. 9, pp. 2128–2139, 2011.

[16] Y. Nesterov, "A method of solving a convex programming problem with convergence rate o (1/k** 2)," *Doklady Akademii Nauk SSSR*, vol. 269, no. 3, p. 543, 1983.

[17] G. Bovi, M. Rabuffetti, P. Mazzoleni, and M. Ferrarin, "A multiple-task gait analysis approach: kinematic, kinetic and emg reference data for healthy young and adult subjects," *Gait & posture*, vol. 33, no. 1, pp. 6–13, 2011.

[18] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *2012 IEEE/RSJ international conference on intelligent robots and systems*, pp. 5026–5033, IEEE, 2012.

[19] M. Towers, J. K. Terry, A. Kwiatkowski, J. U. Balis, G. d. Cola, T. Deleu, M. Goulão, A. Kallinteris, A. KG, M. Krimmel, R. Perez-Vicente, A. Pierré, S. Schulhoff, J. J. Tai, A. T. J. Shen, and O. G. Younis, "Gymnasium," Mar. 2023.

[20] D. A. Winter, *Biomechanics and motor control of human movement*. John wiley & sons, 2009.