UCLouvain

École polytechnique de Louvain

Offline Human-in-the-Loop Reinforcement Learning for Personalized Tuning of a Powered Ankle-Foot Prosthesis

Author: **Raphaël VASSART**
Supervisors: **Eric PIETTE, Renaud RONSSE**
Readers: **Luana MARSANO DA COSTA NUNES, Quentin CAPPART**
Academic year 2025–2026
Master [120] in Computer Science and Engineering

Contents

Acknowledgments

Since childhood, I have always been interested in computer science, robotics, and, more recently, artificial intelligence. A few years ago, science and medicine saved my brother's life by curing him from cancer. This experience deeply strengthened my respect for the medical field and made me realize how meaningful scientific and technological progress can be when it directly improves and saves people's lives. In that sense, this master's thesis topic felt like an obvious choice. Being able to combine artificial intelligence with the medical field, and more specifically with rehabilitation robotics, was exactly the kind of project I had hoped to work on.

This manuscript marks the end of a long year of continuous and demanding work. However, none of this work could have been carried out alone.

Let me start from the beginning. I would like to sincerely thank my supervisors, Professor Eric Piette and Professor Renaud Ronsse, for their guidance throughout this year, through our many meetings, discussions, and feedback sessions. Your advice was always valuable, and your involvement in this project was greatly appreciated. I also hope that my sometimes rather dense slides were at least somewhat useful to you.

I would also like to thank you, Luana. I had the chance to collaborate with you on this project and to benefit from your expertise in biomechanics and robotics. Thank you for your many ideas, suggestions, and comments during our meetings and discussions. Thank you also for reassuring and helping me in the moments when I felt overwhelmed by the amount of work. Finally, I am grateful that you supervised the data collection campaign with the ELSA prosthesis. Even if the prosthesis was not always as cooperative as we had hoped, this campaign remained an essential step in the completion of this thesis.

Also, thank you for the computational resources that have been provided by the Consortium des Equipements de Calcul Intensif (CECI), funded by the Fonds de la Recherche Scientifique de Belgique (F.R.S.-FNRS) under Grant No. 2.5020.11 and by the Walloon Region. Without them, I would never have been able to run so many experiments and tests on my model in such a short space of time.

On a lighter note, I would also like to thank ChatGPT and Claude AI for existing, and for sparing me many long and painful sessions of text reduction, reformulation during the final hours of writing this report. They were also use for code debugging purpose during the year to assist me when fixing my code. This use was carried out in accordance with the regulations and guidelines established by EPL.

Finally, I would, of course, also like to thank my family, my parents and my two brothers, for their constant support and encouragement throughout this year.

1

Introduction

Powered lower-limb prostheses can now inject positive mechanical work during push-off, partially closing the energetic gap of passive devices. Yet at UCLouvain, where the Efficient Lockable Spring Ankle (ELSA) was developed, this mechanical sophistication does not yet translate into a personalized walking experience. The control parameters that shape stiffness, damping, and assistance timing must still be hand-tuned by someone else than the user in a time-consuming, subjective and hard-to-scale process. This work required the integration of two domains: *reinforcement learning* (RL) and *powered ankle-foot prosthesis control*. The state-of-the-art review then revealed a clear gap: existing RL approaches for prosthesis tuning succeed at reproducing reference gait kinematics but rarely ask the user whether the resulting walk feels comfortable, well-timed, or trustworthy. Filling that gap, by embedding structured *subjective feedback* in an offline, small-data tuning framework for a powered ankle, became the horizon this thesis sets out to explore.

The proposed **Human-in-the-Loop Reinforcement Learning (HITL-RL)** framework adds a learning layer above ELSA's existing controller. It observes biomechanical gait deviations, integrates the user's comfort and perceived-assistance scores, and learns offline how to recommend bounded updates of three interpretable control parameters. Around it, a dedicated data-collection protocol was designed in collaboration with a PhD researcher, a modular Python pipeline was implemented end-to-end, and a structured ablation campaign was run. The contribution is therefore not a deployed clinical controller, but an offline proof of concept and evaluation pipeline for studying these two research questions:

1. **Integration.** Can a HITL-RL learning layer *coexist* with ELSA's existing controller while preserving its safety guarantees and clinical interpretability?

2. **Value of human feedback.** In the small-data offline regime of a real prosthesis experiment, does subjective user feedback bring *measurable learning value* over a purely biomechanical reward?

The remainder of this report answers these questions, from the theoretical and technological foundations (Chapters 2, 3, 4) to the proposed framework (Chapter 5), its implementation (Chapter 6), the experimental protocol (Chapter 7), and the analysis of the learned behavior (Chapter 8).

2

Background

This chapter gathers the conceptual material on which the rest of the report relies: the walking-biomechanics vocabulary used to describe the prosthesis behavior (Section 2.1), the reinforcement-learning scaffold that frames sequential decision-making and its offline variant (Section 2.2), and the extension that folds human feedback into this scaffold as a distinct learning signal (Section 2.3).

2.1 The Human Gait Cycle

This thesis addresses the personalization of a powered ankle-foot prosthesis. The discussion therefore rests on a small set of walking-biomechanics concepts, which this section introduces.

A *gait cycle* is the period between two successive heel strikes of the same foot, conventionally normalized between 0 and 100 % [1]. It splits into a *stance phase* (≈ 0–60 %), during which the foot remains in contact with the ground, and a *swing phase* (≈ 60–100 %), during which it is lifted and brought forward (Figure 2.1). Four events structure the cycle:

- *heel strike* (HS), at 0 %, when the heel first contacts the ground;
- the *maximal plantarflexion angle* (MPA), reached shortly after HS when the foot becomes flat on the ground;
- *push-off*, the propulsive end-of-stance sub-phase during which the ankle generates positive mechanical work;
- *toe-off*, marking the transition into swing.

Ankle motion in the sagittal plane is described by a single angle: *dorsiflexion* denotes upward rotation of the foot (positive angle), *plantarflexion* the opposite (negative angle) [1]. The natural trajectory of a healthy adult on level ground exhibits three characteristic deflections (cf. Figure 2.1): a small plantarflexion drop just after heel strike (reaching MPA), a progressive dorsiflexion during rollover as the body rotates over the supporting foot, and a pronounced plantarflexion peak during push-off. The peak-to-peak amplitude defines the joint's range of motion (RoM), typically around 30° in level-ground walking.

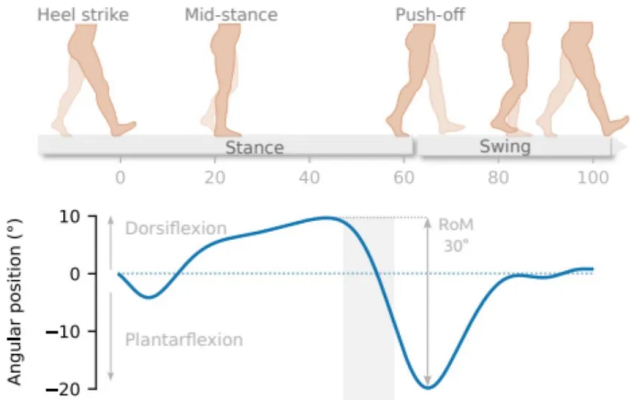


Figure 2.1: The human gait cycle at the ankle. *Top:* successive events (heel strike, mid-stance, push-off, toe-off) within the stance and swing phases of the normalized cycle. *Bottom:* typical sagittal-plane ankle trajectory of a healthy adult, showing the post-heel-strike plantarflexion drop (MPA), the dorsiflexion during rollover, and the push-off plantarflexion peak. The peak-to-peak amplitude defines the range of motion (RoM, $\approx 30^{\circ}$) [2].

These gait events are not only used here as biomechanical landmarks. They will later serve as reproducible extraction points for the gait features used in the reinforcement learning state representation.

2.2 Reinforcement Learning Foundations

Reinforcement learning (RL) provides the conceptual and algorithmic vocabulary on which the rest of this report relies. This section unfolds it in a self-contained manner: from the formalism that defines what a sequential decision-making problem is (Section 2.2.1), to the value-based reasoning that allows policies to be evaluated and improved (Section 2.2.2), to the adaptations required when exact solutions become intractable (Section 2.2.3), and finally to the offline regime in which the agent learns from a fixed dataset rather than from interaction (Section 2.2.4).

2.2.1 Formalizing sequential decision-making

Many real-world problems share a common structure: an agent must repeatedly choose an action, observe its effect, and use that experience to choose better next time. Personalizing a prosthesis controller is one such problem, but so is steering a car, treating a patient, or playing a game. Reinforcement learning is the branch of machine learning concerned with sequential decision-making problems, where an agent learns to *choose actions from their consequences over time* [3, 4]. Historically, RL can be seen as the meeting point between trial-and-error learning [5] and dynamic programming [6], but the relevant viewpoint in this thesis is primarily operational: RL provides a *formal language* to describe what the prosthesis-tuning agent observes, what it can safely modify, and how the quality of these modifications is evaluated.

To reason about such problems formally, one needs a language that captures both the choices available to the agent and the way the world responds. The standard one is the *Markov Decision Process* (MDP): a tuple $(\mathcal{S}, \mathcal{A}, p, \gamma)$ where $\mathcal{S}$, the *states*, is the set of possible

situations the agent can find itself in, $\mathcal{A}$, *the actions*, the set of available decisions, $p(s' \mid s, a)$ the probability that taking action a in state s leads to state s', and $\gamma \in [0, 1]$ a number called the *discount factor* whose role will be clarified shortly [7]. The defining assumption of this formalism is the *Markov property*: the present state contains all the information needed to predict the future [3, 6]. In practice, a realistic and reasonable representation of the states often does not allow for this, for example, a person walking with a prosthesis adapts, fatigues, and shifts balance over time. But it provides a tractable scaffold within which approximate solutions can be searched for.

Inside this formalism, an agent's behavior is described by a *policy* π, a rule mapping each state to the action to take there [3]. At each time step t, after taking action A_t in state S_t, the environment transitions to a new state S_{t+1} and emits a scalar *reward* R_{t+1} measuring how desirable the transition was (cf. Figure 2.2). The agent's objective is not to maximize any single reward but the discounted sum of all future rewards, called the *return*:

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots \tag{2.1}$$

The factor γ controls the trade-off between immediate and future gains: values close to 1 make the agent patient and long-sighted, while smaller values emphasise short-term outcomes [7].

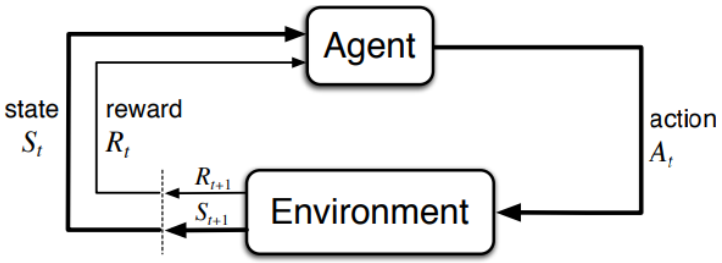


Figure 2.2: The agent-environment interaction in a Markov decision process [3]

The MDP framework tells us *what* the agent observes, *what* it can do, and *what* it is trying to maximise. It does not yet tell us *how* a good policy is found.

2.2.2 Evaluating and improving decisions

Because rewards are delayed, evaluating a policy is harder than evaluating a single decision. RL addresses this through value functions. The *state-value* $v_\pi(s)$ is the expected return when starting from state s and following π thereafter [3, 6]:

$$v_\pi(s) = \mathbb{E}_\pi \left[G_t \,\middle|\, S_t = s \right], \tag{2.2}$$

Conditioning additionally on the first action taken yields the *action-value function* $q_\pi(s, a)$, which answers the slightly different question: *what is the expected return of doing action a in state s, then continuing with π?*

$$q_\pi(s, a) = \mathbb{E}_\pi \left[G_t \,\middle|\, S_t = s, A_t = a \right]. \tag{2.3}$$

This second form turns out to be more directly useful, because it allows the policy to be improved one action at a time.

Both functions satisfy a self-consistency relation: the value of an action equals the immediate reward expected when taking it, plus the discounted value of what follows under the same policy. Formalised on the action-value, this is the *Bellman expectation equation* [3, 6]:

$$q_\pi(s,a) = \mathbb{E}_\pi\left[R_{t+1} + \gamma\, q_\pi(S_{t+1}, A_{t+1}) \middle| S_t = s, A_t = a \right]. \tag{2.4}$$

Every value-based RL algorithm is, at its core, a procedure for finding a q-function that satisfies this relation as closely as possible on the data at hand [8].

When the same self-consistency holds not for some policy but for the *best* policy among all possible ones, the expectation over the next action is replaced by a maximum, giving the *Bellman optimality equation*:

$$q^\star(s,a) = \mathbb{E}\left[R_{t+1} + \gamma \max_{a'} q^\star(S_{t+1}, a') \middle| S_t = s, A_t = a \right]. \tag{2.5}$$

The corresponding *greedy policy* selects, at every state, the action that achieves this maximum:

$$\pi^\star(s) = \arg\max_a \; q^\star(s,a). \tag{2.6}$$

This is the key idea that turns value estimation into policy construction: if one can estimate q accurately, acting greedily with respect to it yields a policy at least as good as the one used for the estimation.

This observation directly yields *Policy Iteration* [9]. Starting from any initial policy, two steps are alternated: *policy evaluation* estimates q_π for the current policy, then *policy improvement* replaces π by the greedy policy with respect to the new q_π. In the ideal finite-state and finite-action case with a known model, policy iteration is guaranteed to improve the policy until convergence [3].

Few real problems satisfy these idealised conditions. Sutton and Barto introduced the name *Generalized Policy Iteration* (GPI) for the broader family of algorithms that follow the same evaluation-improvement skeleton but in which evaluation may be approximate, improvement may be partial, and the two may even be interleaved at every transition rather than alternated cleanly [3]. Most modern RL methods are instances of GPI in this loose sense.

2.2.3 When exact solutions are out of reach

When the state or action spaces are continuous, the number of distinct (s,a) pairs becomes effectively infinite and the lookup-table representation of q_π is no longer feasible. The standard remedy, called *function approximation*, is to replace the exact q-function with a *parametric model* $\hat{q}(s,a;\theta)$. It is a compact mathematical object controlled by a small number of parameters θ that maps any (s,a) to a value [3, 10]. Choosing this object has far-reaching consequences.

With function approximation, the Bellman equation can no longer be satisfied exactly: the learned value function is restricted by the chosen function class and therefore retains an approximation error [8, 11]. As a result, approximate policy-iteration methods should

not be interpreted as guaranteeing exact optimality, but rather as seeking a good solution within the representational limits of the chosen approximation. This trade-off is the price paid for applying RL to continuous state-action spaces with finite data.

The evaluation step itself must be adapted accordingly. Since $\hat{q}$ cannot satisfy the Bellman equation pointwise, the parameters θ are chosen instead to *minimise its violation*, measured by the squared *Bellman residual* averaged over the available transitions [11, 12]:

$$\min_{\theta} \sum_{k} \Big(\hat{q}(s_k, a_k; \theta) - r_k - \gamma\, \hat{q}(s_{k+1}, \pi(s_{k+1}); \theta) \Big)^2. \tag{2.7}$$

Each summand compares what the approximator predicts at (s_k, a_k) with what it predicts one step ahead, augmented by the observed reward r_k. Minimizing their sum thus amounts to fitting $\hat{q}$ as a regression on the dataset. This formulation yields the *Least-Squares Policy Iteration* family [12], pairing a regression-based evaluation step with a greedy improvement step that selects, at each state, the action minimizing the freshly fitted $\hat{q}$.

Approximation, however, does not solve every difficulty: it still assumes that the agent can interact freely with the system to gather the transitions needed to fit $\hat{q}$. In safety-critical applications, this assumption breaks too, and a final adaptation is required.

2.2.4 Learning from a fixed dataset: the offline setting

An RL algorithm is called *on-policy* if it learns about the same policy that generated its data, and *off-policy* if the two policies differ [3]. The latter family allows learning from transitions produced by any behaviour, including transitions collected long before the algorithm is even chosen, and opens the door to a regime that goes one step further: learning entirely without interaction.

Offline reinforcement learning refers to settings in which the agent never interacts with the system during training but learns from a fixed dataset of transitions

$$\mathcal{D} = \{(s_k, a_k, r_k, s'_k)\}_{k=1}^{N}, \tag{2.8}$$

collected beforehand under some unknown behaviour policy [13]. This regime matters whenever interaction is expensive, slow, or unsafe. Collecting transitions from a human user requires the user to actually perform the task each time, and online exploration in safety-critical systems may expose the agent or its environment to unstable intermediate behaviours. In such cases, training and inspecting a policy entirely offline becomes a methodological requirement, not a preference.

Offline RL is not, however, a free lunch. Without interaction, the algorithm cannot *try* an action it has not seen. It can only reason about unseen actions through what its function approximator extrapolates. Two structural difficulties follow. The first is *distribution shift*: if the learned policy recommends actions far from those present in the dataset, the $\hat{q}$ values at those actions are extrapolations that may be unreliable yet appear attractive to the optimiser [13, 14]. The second is *support coverage*: regions of the state-action space not represented in the dataset are essentially invisible to the algorithm, so the learned policy can be trusted only within, or close to, what the dataset has exposed.

Offline RL includes many algorithmic families, from least-squares policy-iteration methods to modern deep RL approaches that explicitly control out-of-distribution actions [13].

This thesis adopts the former direction: a lightweight value-based method, better suited to the small-data and interpretability constraints of prosthesis tuning. The four ingredients introduced in this section, MDPs, value functions, function approximation, and offline data, therefore provide the theoretical scaffold for the chosen algorithm of the framework used later (cf. Chapter 5).

2.3 Human-in-the-loop Reinforcement Learning

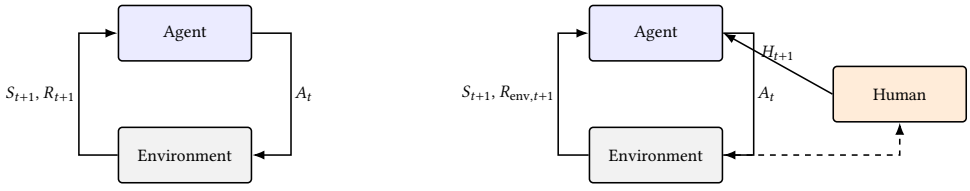
The reinforcement learning framework of the previous section assumes that the scalar reward delivered by the environment fully captures the objective of the task. For many real-world problems this assumption is overly optimistic. The reward may be sparse, hard to specify, or fundamentally unable to encode subjective criteria such as comfort, perceived stability, or naturalness [15, 16]. In such settings, relying only on environmental rewards can lead to slow convergence or suboptimal behaviors [17]. *Human-in-the-Loop Reinforcement Learning* (HITL-RL) addresses this gap by treating the human not as a passive end-user but as an additional source of information that shapes reward estimation, value updates, or action selection [15, 18, 19]. This section unfolds the formalism HITL-RL adds on top of the classical MDP scaffold (Section 2.3.1), surveys the main forms human feedback can take (Section 2.3.2), and motivates why this paradigm fits assistive robotics particularly well (Section 2.3.3).

2.3.1 Embedding human feedback in the MDP/GPI scaffold

A classical MDP $(\mathcal{S}, \mathcal{A}, p, r, \gamma)$ provides no entry point for information that does not flow through the environmental reward r. HITL-RL extends this picture by adding a human channel H_t that travels alongside the standard transition signals at each step [15]:

$$(S_t, A_t) \longrightarrow (R_{\text{env},t+1}, H_{t+1}, S_{t+1}). \tag{2.9}$$

The content of H_{t+1} is intentionally left abstract at this stage. It may be a scalar approval signal [18, 20], a comparison between two trajectories [17, 21], a demonstration [22–24], or a correction [25]. Section 2.3.2 returns to this taxonomy and details how each form is concretely folded into the learning loop. Figure 2.3 contrasts this augmented loop with the standard agent-environment diagram of Section 2.2.1.



(a) Classical MDP *(b) HITL-RL*

Figure 2.3: From classical reinforcement learning to human-in-the-loop reinforcement learning. (a) In the standard MDP, the agent and the environment exchange actions and reward-augmented state signals. (b) In HITL-RL, a human supervisor delivers an additional feedback signal H_{t+1} that the agent folds into reward estimation, value updates, or policy improvement. The dashed link represents the human's access to the system, which may be concurrent with learning (on-policy) or precede it on a batch of recorded trajectories (off-policy / offline).

Conceptually, this addition does not break the Generalized Policy Iteration template introduced in Section 2.2.2. It enriches the inputs that drive its two steps. The policy-evaluation step can now estimate a value function under a reward that incorporates *human information*, while the policy-improvement step can additionally constrain or steer the greedy update through *human advice* or *safety corrections* [18, 25]. In this sense, HITL-RL is best understood not as a new algorithmic family but as an extension of GPI whose evaluation and improvement steps both admit a human contribution.

The distinction between on-policy and off-policy learning introduced in Section 2.2.4 carries directly over. As represented by the dashed link in Figure 2.3, a human can either rate behavior generated by the current policy, or supply feedback once on a batch of previously executed trajectories. *Off-policy variants* are particularly relevant whenever each interaction is expensive, slow, or unsafe, since they allow the algorithm to reason about a fixed set of previously collected human-evaluated transitions without exposing the user to intermediate untrained policies.

2.3.2 Forms of human feedback

The literature has explored several concrete ways for H_t to enter the learning loop, summarized in Figure 2.4 [15, 19]. The two paragraphs below detail the families that have received the most attention, evaluative reward shaping and preference-based learning. A third paragraph briefly lists the remaining modalities in order to give a complete picture of the HITL landscape.

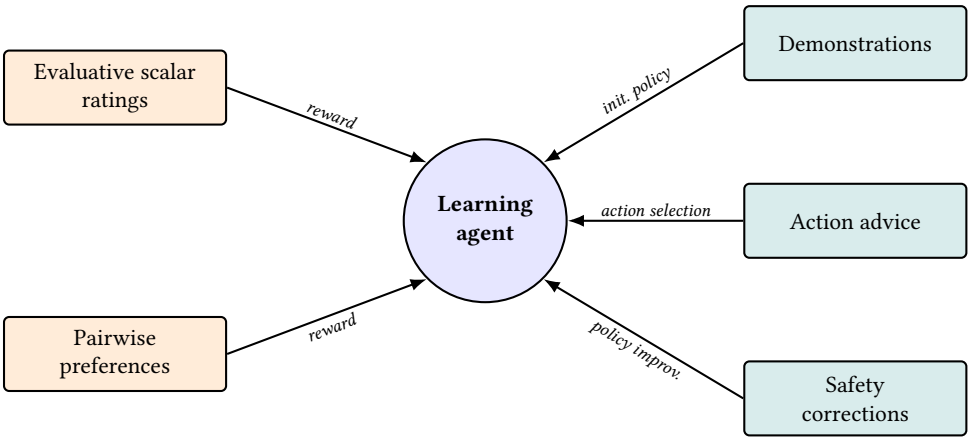


Figure 2.4: The main forms of human feedback explored in HITL-RL, organised by their point of entry in the learning loop. Reward-shaping signals (orange) modify the cost-to-go through Eqs. (2.10) and (2.11), while demonstrations, advice, and corrections (green) act on the policy or its admissible action set.

Evaluative reward shaping. The earliest form of human feedback explored in interactive RL was a *scalar approval signal* delivered after observing a state-action pair [20, 26]. Frameworks such as *TAMER* [20] showed that an agent could learn meaningfully from such ratings alone, but they also exposed the structural difficulties that come with scalar human input. Ratings tend to be *sparse*, since humans cannot reasonably score every tran-

sition. They are *noisy*, since two evaluators (or the same evaluator at two different times) may disagree on what counts as a good action. And they are subject to *credit-assignment ambiguity* whenever the action that triggered the feedback is not the one most recently executed [27].

These difficulties motivated the now-standard practice of combining human approval with a well-defined environmental reward rather than relying on it as the sole learning signal [26]. The most common form folds the human term directly into the reward as a convex combination of an environmental component and a human component:

$$r'(s,a) = (1-\lambda)\, r_{\text{env}}(s,a) + \lambda\, r_{\text{human}}(s,a), \tag{2.10}$$

where $\lambda \in [0,1]$ controls how much weight the human signal carries [18]. The formulation is conceptually simple. But the trade-offs it requires, in particular the relative scale, density, and reliability of the two terms, are non-trivial in practice. Excessively weighting an inconsistent human signal can destabilize learning, while ignoring it altogether negates the very motivation for keeping the human in the loop. In this hybrid regime, the human term acts as a corrective bias that informs the agent on dimensions the environmental reward cannot capture, while the environmental reward provides the structural backbone of the optimization.

Preference-based reinforcement learning A second family of HITL-RL methods asks humans to compare behaviors rather than score them independently. Instead of assigning an absolute rating to a single trajectory, the user *indicates which of two trajectories is preferred.* This can be easier and more reliable for subjective criteria such as comfort, naturalness, or perceived assistance, where no obvious numerical unit exists [17, 21]. A common formulation learns a reward model $\hat{r}_\phi$ from *pairwise preferences.* If trajectory τ_i is preferred to trajectory τ_j, the probability of this preference can be modeled with a *Bradley-Terry distribution*:

$$P(\tau_i \succ \tau_j) = \frac{\exp\left(\sum_t \hat{r}_\phi(s_t^i, a_t^i)\right)}{\exp\left(\sum_t \hat{r}_\phi(s_t^i, a_t^i)\right) + \exp\left(\sum_t \hat{r}_\phi(s_t^j, a_t^j)\right)}. \tag{2.11}$$

The learned reward can then be used in place of, or together with, an environmental reward during policy optimization [17, 21]. In the context of prosthesis tuning, this idea is conceptually relevant because a user may find it easier to compare two assistance profiles than to assign stable absolute scores across different sessions.

The same general principle, *learning from human preferences*, has also been scaled far beyond robotics in *reinforcement learning from human feedback* (RLHF), where language models are trained to align their outputs with human judgments [28, 29]. These large-scale systems differ strongly from small-data physical human-robot interaction, but they illustrate a broader point that is central to HITL-RL: human judgments can provide useful learning signals when the desired behavior is difficult to specify with a hand-designed reward. More recent methods such as *Direct Preference Optimization* (DPO) further simplify parts of the RLHF pipeline by fitting directly from preference data [30]. For assistive robotics, such methods are not directly transferable, but they motivate future work on richer and better-conditioned forms of user feedback [31].

Other forms. Beyond reward shaping and pairwise preferences, the literature also considers *demonstrations* used for policy initialization or imitation learning [22, 23, 32], *action advice* that biases the agent's choices without directly changing the reward [18], *safety corrections* that prevent or redirect unsafe behavior [25], and dynamic action-space *extensions* [33]. As Figure 2.4 illustrates, these modalities differ not only in their nature but in the point of the learning loop they touch. Thus, the choice of feedback type is itself a design decision shaped by what is feasible to collect from the user and by which step of the loop it ultimately enters. These modalities are not used in the present framework, which focuses on block-level scalar ratings. They nevertheless remain relevant design options for future versions of human-in-the-loop prosthesis tuning, especially if the framework evolves toward online adaptation.

2.3.3 Toward physical human-robot interaction

A natural application domain for HITL-RL lies in *physical Human-Robot Interaction* (pHRI), the sub-field of robotics that studies systems in which a human and a robot share a mechanical interface and exchange forces, motion, or energy in a sustained manner [34]. This is conceptually distinct from *cognitive* or *social* HRI, where the exchange operates through informational channels such as speech, gaze, or expressive gesture. In cognitive interaction the robot's behavior is observed and interpreted, whereas in physical interaction it is *felt*. Examples span collaborative manipulators, rehabilitation exoskeletons, haptic interfaces, and powered prostheses.

These systems share three properties that make pHRI a natural application domain for HITL-RL: continuous parameter spaces that are difficult to tune from first principles, interactions that are costly or unsafe to explore extensively online, and success criteria that combine objective biomechanical signals with subjective dimensions such as comfort, effort, and trust [15]. HITL-RL is relevant in this context because it can combine sensor-based measurements with human feedback, while remaining compatible with offline or off-policy learning when direct exploration is limited. This combination is particularly important for lower-limb prosthesis tuning, where control decisions must be data-efficient, physically safe, and aligned with both measurable gait behavior and the user's perception of the device. The next chapter therefore reviews how reinforcement learning has been applied to lower-limb robotic prostheses, a particularly demanding instance of this challenge.

3

Reinforcement Learning for Robotic Prosthesis Impedance Tuning

Personalizing the controller behavior of powered lower-limb prostheses has emerged as a central challenge in rehabilitation robotics. Despite rich mechanical capabilities and adaptable controllers, deployment still relies on *manual tuning by expert prosthetists or experimenters*. This is slow, subjective, and hard to scale, and it worsens as parameter spaces grow. Two research directions address this problem. One improves tuning through *optimization procedures* carried out with the *user physically in the loop*, often using physiological or biomechanical outcome measures. The other formulates *tuning as a reinforcement-learning problem*. In both cases, however, the human is most often present as part of the walking dynamics or as the source of measured performance outcomes, not as a structured source of subjective feedback entering the learning objective. This distinction is central to the positioning of this thesis: the gap addressed here is not merely the absence of human-in-the-loop experiments, but the absence of explicit user feedback as a *learning signal* for prosthesis tuning.

3.1 Impedance Control and Human-in-the-Loop Optimization

Powered prostheses are often controlled through *mechanical impedance*. It describes how a system opposes an imposed motion, by relating the torque it returns to the displacement and the velocity of that motion [35]. Typically, a linear impedance control law reduces to a stiffness term, proportional to the deviation from an equilibrium angle, and a damping term, proportional to the angular velocity. A prosthesis controlled with this approach will thus generate a joint torque within each gait phase that follows

$$\tau = k(\theta - \theta_{ref}) + d\,\dot{\theta}, \tag{3.1}$$

with k the virtual stiffness, d the virtual damping, and θ_{ref} the virtual equilibrium angle. Tuning a prosthesis means choosing these parameters within a *finite-state impedance*

controller (FSM-IC), formalized in early robotic designs around 2008 and 2009 [36, 37]. These parameters are set per gait phase to reproduce able-bodied normative kinematics [38], which restores basic locomotion but depends heavily on clinician expertise and scales poorly, since modern powered knee prostheses may need more than a dozen parameters per locomotion mode [38, 39].

A parallel line keeps the user inside the optimization loop. Zhang et al. reduced natural walking metabolic cost in 2017 through iterative parameter optimization of a assistive device driven by physiological measurements [40]. Ding et al. obtained comparable gains in 2018 on a soft exosuit [41]. These results show the value of optimizing against the response of the actual wearer. They also use few parameters and steady-state cost evaluation, which does not fit high-dimensional phase-dependent impedance controllers. These limitations motivated reinforcement learning approaches as it is able to tune high-dimensional sequential decisions without requiring explicit identification of the human-prosthesis dynamics, while the optimization line had already shown that the user response is itself a valuable signal.

3.2 Reinforcement Learning for Impedance Tuning

Reinforcement learning recasts parameter adjustment as a sequential decision-making problem, learning a control policy from interaction data without a model of the human-prosthesis dynamics. Feasibility for knee prosthesis impedance tuning was first established in 2017 through a simulation study based on adaptive dynamic programming [42]. Related work then tuned several impedance parameters in real time on the real system [43, 44]. Online learning, however, exposes the user to long exploration and to poorly tuned intermediate behaviors, with limited guarantees on stability or suboptimality. These concerns moved the field toward more data-efficient and structured policy-iteration schemes.

3.2.1 Offline and Structured Policy Iteration

The following developments come mainly from one research group and form a coherent line of work on data-efficient impedance tuning for powered knee prostheses. In the reviewed literature, no directly comparable RL-based impedance tuning study has been identified for powered ankle-foot prostheses. The knee-prosthesis literature is therefore considered here as the closest available experimental context for single-joint prosthesis parameter tuning, while keeping in mind that the biomechanical role of the ankle, especially during rollover and push-off, is substantially different from knee flexion-extension. This line of work is not presented as a complete survey of all RL approaches to prosthesis control, but as the most directly relevant progression from online adaptive dynamic programming toward offline and structured policy iteration. It provides the closest precedent for learning personalized parameter updates from limited human-prosthesis data, shaping the direction taken in the framework developed for the thesis.

Offline policy iteration (OPI), introduced in 2019, was a major step in this direction [45]. It separates learning into two stages, shown in Figure 3.1. An offline phase reuses previously collected gait data to approximate a near-optimal policy. A short online phase then refines the impedance parameters with limited exploration. Near-normative knee kinematics were reached within a few updates, with a large cut in tuning time over online-only methods.

This two-stage structure became the backbone of the later literature.

Two refinements followed, from the same research group. *Flexible Policy Iteration* (FPI), in 2020, added experience replay to reuse past data during policy evaluation, with system-level guarantees on value-function convergence, stability, and suboptimality [46]. Such guarantees matter in safety-critical rehabilitation, where uncontrolled updates can compromise balance or comfort.

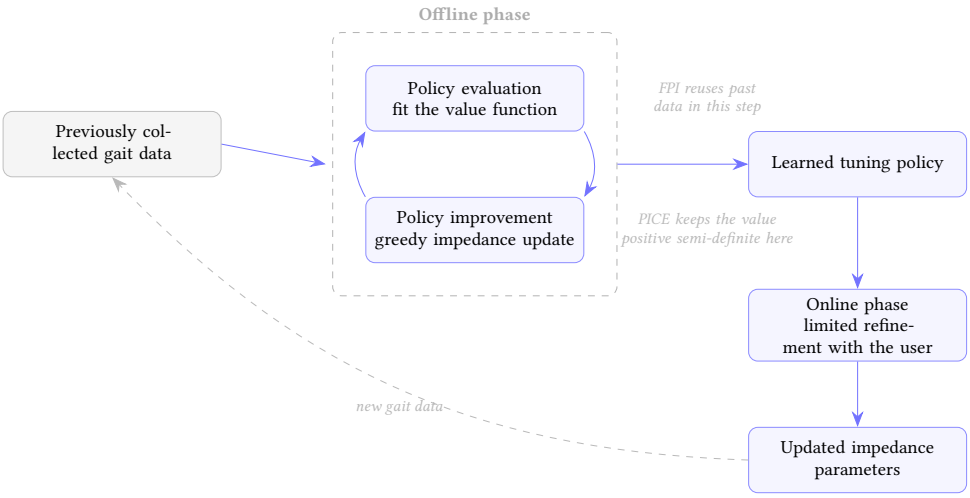


Figure 3.1: Shared two-stage backbone of offline policy iteration for prosthesis tuning [45]. An offline phase alternates policy evaluation and policy improvement on previously collected gait data. A short online phase then refines the impedance parameters with the user.

Policy Iteration with Constraint Embedded (PICE), in 2021, kept the value positive semi-definite during evaluation through a projected Bellman equation, with online and offline implementations [47]. Human-subject tests reported faster convergence and robustness across users and tasks. Figure 3.2 summarizes this evolution.

The reference itself was then questioned. Early formulations tracked fixed trajectories, but human gait is adaptive and a time-invariant template can hinder limb coordination. In 2022, the same line coordinated the prosthesis with the *intact limb*, re-framing personalization as adaptive coordination rather than static tracking [48]. A reference is still needed. It is simply better used as a flexible anchor than as a rigid target.

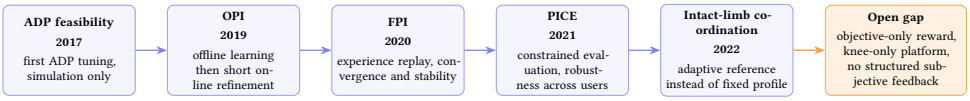


Figure 3.2: Evolution of RL for robotic knee prosthesis impedance tuning and the remaining gap.

3.2.2 Open Challenges and Research Positioning

Reinforcement learning can tune high-dimensional impedance controllers without system identification, yet three challenges remain.

First, the objective remains almost *purely biomechanical*. Most methods optimize measurable quantities such as kinematic tracking error, impedance-tracking performance, or coordination with the intact limb. Comfort, perceived stability, perceived assistance, and gait naturalness rarely enter the learning objective explicitly. This is an important distinction because several existing formulations are described as human-in-the-loop: the human wearer is indeed physically present in the tuning loop, and their gait dynamics shape the data collected by the algorithm. However, the wearer is not usually treated as a structured source of subjective feedback that directly shapes the reward or policy update. In other words, prior work is often human-in-the-loop in the experimental sense, but not human-feedback-driven in the learning sense. Integrating explicit subjective feedback into the objective therefore remains largely open for prosthesis control.

Second, the experimental scope is narrow. The whole lineage above targets the robotic knee. None of these formulations has been transposed to a *powered ankle*, whose limited torque sensing and decisive push-off timing reshape both the state description and the safety constraints a learning layer must respect.

Third, the definition of the biomechanical reference itself remains open. Most tracking-based objectives rely on an external norm, often derived from average able-bodied gait. Such a reference provides a useful and interpretable anchor, as in this thesis, but it may not represent an attainable or desirable target for every individual user. Future formulations may therefore need to reduce their dependence on fixed normative references and move toward more user-specific objectives.

Finally, clinical translation is still limited. It requires balancing exploration with safety, fast convergence, and robustness to inter-user variability. Recent frameworks improved theoretical grounding and data efficiency, but long-term validation in realistic conditions stays scarce, and the small-data regime typical of real prosthesis campaigns remains demanding.

These problems frame two research questions. The first is whether a learning layer using structured subjective feedback can be added to an existing impedance controller without weakening safety and interpretability. The second is whether that feedback brings measurable value in a small-data offline regime. ELSA, the powered ankle developed at UCLouvain, offers exactly the structured and interpretable experimental platform such a study requires.

4

The ELSA Prosthesis as Experimental Platform

The state of the art identified a precise gap. Structured reinforcement learning for prosthesis tuning has matured on the robotic knee, under purely biomechanical objectives. It has never been transposed to a powered ankle. This chapter introduces the platform on which that transposition is attempted. The description is focused on what a learning layer needs, namely the control level it acts on, the parameters it moves, and the manual process it replaces [2, 49].

4.1 From Powered Prostheses to ELSA

Powered lower-limb prostheses improve on passive feet by actively injecting positive mechanical work at push-off, which partially closes the energetic deficit caused by amputation [50]. ELSA, the *Efficient Lockable Spring Ankle* developed at UCLouvain, is one such device (cf. Figure 4.1). It is a compact, foot-size powered ankle-foot prosthesis that reproduces key ankle dynamics across locomotion tasks [2, 49]. For the present thesis, the key property of ELSA is not the detailed actuator design [51], but the fact that its controller exposes a small set of bounded, interpretable parameters through which ankle behavior can be personalized.

Figure 4.1: Full prototype of ELSA 3.1 with the foot cover and pylon.

4.2 Hierarchical 3-Layer Control Architecture

ELSA relies on a *hierarchical three-layer control architecture* [52], as detailed in Figure 4.2.

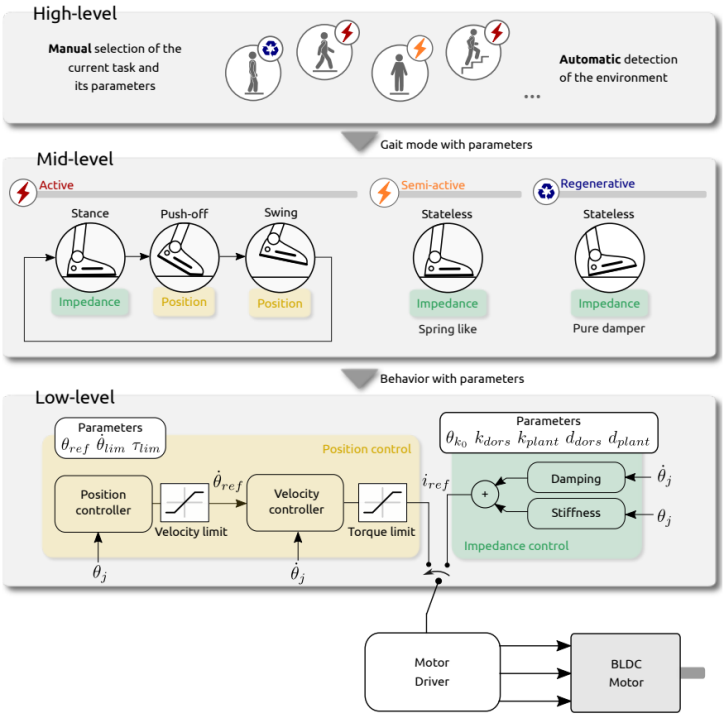


Figure 4.2: 3-level Hierarchical Control Architecture of ELSA prosthesis [2].

The low level regulates motor current and tracks the torque or impedance commands sent by the layers above, ensuring actuator stability and safe current limits. This work does not modify it. The high level runs a rule-based detector that recognizes the current locomotion task (level walking, slopes, stairs, sit-to-stand) and selects the matching gait mode [53]. It switches between modes but does not personalize the impedance parameters inside a mode. That missing personalization is the gap this thesis addresses. This makes the mid-level controller the appropriate intervention point for learning: it is high enough to preserve low-level safety and low enough to modify the gait behavior felt by the user.

4.2.1 Mid-Level Finite-State Impedance Control

The mid-level controller follows a finite-state impedance scheme, the dominant paradigm in powered prosthetics [54]. The gait cycle is split into discrete phases (cf. Section 2.1). Within each phase the ankle torque follows the finite-state impedance law already introduced and developed in the Section 3.1 (cf. Eq. (3.1)).

Table 4.1 lists the bounded, clinically interpretable parameters of the mid-level controller. The damping terms d_{plant} and d_{dors} shape the foot resistance during plantarflexion and dorsiflexion, which affects the response felt just after heel strike and during rollover. The stiffness terms k_{plant} and k_{dors} modulate the perceived rigidity, mainly for standing and the rollover phase. Push-off is governed by θ_{trig}, τ_{lim}, and θ_{tar}, which set when assistance is triggered, how strong it is, and the target ankle angle at the end of push-off.

Var.	Parameter	Range	Units
d_{plant}	Damping in plantarflexion	(0; 0.5)	Nm/(°/s)
d_{dors}	Damping in dorsiflexion	(0; 0.5)	Nm/(°/s)
k_{plant}	Stiffness modulation, plantarflexion	(0; 7)	Nm/°
k_{dors}	Stiffness modulation, dorsiflexion	(0; 7)	Nm/°
θ_{trig}	Push-off trigger angle	(3; 11)	°
τ_{lim}	Push-off assistance level (torque limit)	(10; 60)	Nm
θ_{tar}	Target angle during push-off	(−10; −20)	°

Table 4.1: Tunable ELSA control parameters and their admissible ranges.

4.3 The Manual Tuning Bottleneck and the Case for HITL-RL

ELSA is still tuned by hand through a web dashboard, shown in Figure 4.3. A prosthetist adjusts the parameters, observes the resulting gait, asks the user for feedback, and repeats. The procedure works in a controlled laboratory, but it is time-consuming, subjective, and hard to scale across users, tasks, and environments. This manual loop already contains the ingredients of the proposed RL formulation: an observed gait response, a parameter update, and subjective user feedback. What is missing is a formal mechanism to record these elements as transitions and use them to learn a tuning policy.

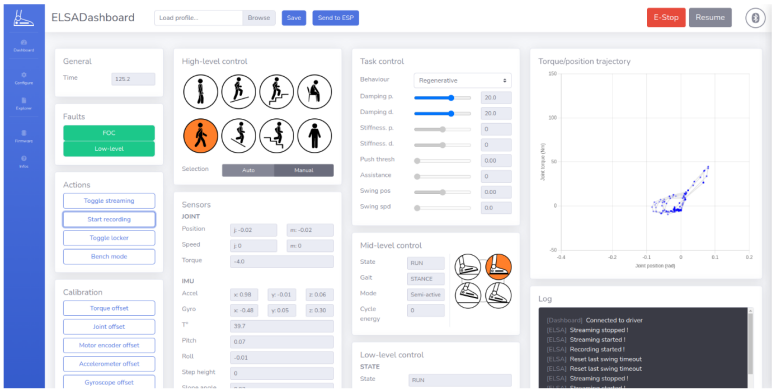


Figure 4.3: ELSA Manual Tuning Web Dashboard. Parameters are adjusted manually via the sliders on the Task Control panel.

Overall, two properties make ELSA well suited to a formal human-in-the-loop approach. The FSM-IC exposes a structured, bounded parameter space that is far smaller and more interpretable than direct torque control. And each parameter has a clinical meaning, so parameter updates stay explainable and easy to bound safely. This thesis does not replace ELSA's controller. It adds an offline learning layer above the mid-level FSM-IC, working inside that bounded space so the current safety guarantees and clinical interpretability are kept by construction. This defines the platform and the constraints under which the framework is designed.

5

A HITL-RL Framework for ELSA Personalization

The previous chapter identified the main control bottleneck of ELSA: despite its structured hierarchical controller, the prosthesis still requires *manual parameter tuning*. In practice, this process already includes a human in the loop, as the experimenter adjusts parameters, observes gait, asks for feedback, and iterates. However, this feedback remains *unstructured* and is *not directly integrated* into an algorithmic optimization process.

This chapter turns this observation into a *Human-in-the-Loop Reinforcement Learning* (HITL-RL) formulation. The objective is not to replace the existing ELSA controller or to learn direct motor commands. Instead, the proposed framework adds a learning layer above the mid-level controller. This layer observes compact gait features, integrates human feedback, and learns how selected prosthesis parameters should be updated from one walking condition to the next. The following sections define the scope of this first framework, the tunable parameter space, and the adaptation of the classical Markov Decision Process formulation to human-prosthesis tuning.

5.1 Motivation, Scope, and Tunable Parameters

ELSA relies on a finite-state impedance controller, where the gait cycle is divided into phases and ankle behavior is shaped through interpretable parameters such as damping, stiffness, assistance timing, and torque limits (cf. Table 4.1) [2, 49]. This structure is well suited to learning-assisted personalization: rather than acting directly on motor current or torque commands, the learning algorithm operates on biomechanically meaningful control parameters. The existing controller architecture is therefore preserved, while data-driven adaptation becomes possible.

This first version of the framework is deliberately restricted to *level-ground walking* (LGW). Although ELSA can operate across several locomotion modes [2, 53], learning across all modes at once would increase the dimensionality of the problem and make it harder to separate the effect of the learning policy from task transitions and natural gait variability. Focusing on level-ground walking keeps the problem experimentally tractable while still addressing the central challenge of personalized prosthesis tuning.

Another key design choice is to formulate the problem *offline*. As argued generically in Section 2.3.1, off-policy and offline variants of HITL-RL are particularly relevant when each interaction is costly or unsafe, a constraint that applies directly to a human-prosthesis system. Every transition requires a user to walk with the device under a given configuration, and unsafe intermediate policies may affect comfort, stability, or fatigue. This follows previous work on robotic prosthesis tuning, where offline policy iteration was used to improve data efficiency and reduce online exploration with human users [45, 47]. The proposed framework is therefore designed as an *offline-trained tuning policy*, while keeping a structure that could later be integrated into an online adaptation loop.

Within this restricted scope, the first version of the framework acts on a deliberately reduced action space. Among the many ELSA parameters (see Section 4.2.1), the remaining ones are kept fixed to limit dimensionality and establish a controlled baseline before extending the framework to a larger parameter space. Three parameters were selected through a joint engineering assessment by the thesis author and the prosthesis's biomechanical experts. The selection criterion was that each parameter had to exert a direct and perceptible influence on level-ground gait, remain interpretable for prosthetists and biomechanists, and carry no normative target value, these parameters are optimized globally across the interaction rather than driven toward a prescribed reference. The three selected parameters are:

$$d_{\text{plant}}, \quad \theta_{\text{trig}}, \quad \tau_{\text{lim}}$$

These respectively denote *plantarflexion damping*, *push-off trigger angle*, and the *motor torque limit during push-off*, as introduced in Section 4.2.1. The parameter d_{plant} governs energy dissipation during the foot response in plantarflexion, especially after heel strike and during the transition toward push-off. A lower damping value allows faster plantarflexion, whereas a higher value slows the foot motion and may give the user the impression of a firmer heel response. The parameter θ_{trig} determines when push-off assistance is triggered, and therefore affects whether propulsion feels well timed, delayed, or premature. Finally, τ_{lim} limits the maximum torque delivered during push-off, controlling the intensity of assistance and potentially affecting energy consumption, step symmetry, and gait naturalness.

Together, these three parameters cover functional aspects of gait that are both relevant to level-ground walking and readily perceptible to the user: early-stance foot response, assistance timing, and assistance magnitude. They are also interpretable for prosthetists and biomechanists, which is important for a HITL-RL system intended to remain understandable and clinically reasonable.

5.2 From a Classical MDP to Offline Human-Prosthesis Tuning

The MDP formalism introduced in Section 2.2.1 must be adapted to the specific constraints of human-prosthesis experiments.

The main adaptation concerns the *decision time scale*. The RL agent does not act at the low-level frequency of the prosthesis. Instead, decisions are made at the level of walking blocks. A block corresponds to a short walking condition with a fixed number of steps,

during which the prosthesis uses a fixed parameter vector p_k. Two consecutive blocks then define one parameter-transition event.

Thus, the decision index k represents a tuning block rather than an instant in the continuous sensor stream. At block k, the user walks with parameter configuration p_k, from which a compact gait state s_k is extracted. The action a_k is the parameter update leading to the next configuration. After this update, the next block provides a successor state s_{k+1}, and the transition receives a reward r_k based on gait behavior and human feedback.

The offline transition dataset can therefore be written as:

$$\mathcal{D} = \{(s_k, a_k, r_k, s_{k+1})\}_{k=1}^{N}. \tag{5.1}$$

Each element summarizes one observed interaction between a parameter update and its effect on the human-prosthesis system. This remains close to the standard MDP structure, but it should not be interpreted as a complete model of human walking dynamics. The true state includes hidden factors such as adaptation, fatigue, balance strategy, comfort, and step-to-step variability. The proposed state is therefore a compact approximation: it retains the gait features considered most relevant for tuning the selected parameters, while accepting that the process is only approximately Markovian.

This distinction is important. The MDP formulation does not claim that the human-ELSA system is fully observable or perfectly stationary. Rather, it provides a structured scaffold to define what the agent observes, what it can change, and how the quality of a parameter update is evaluated. The next sections instantiate these components through the state representation, action space, reward function, and learning algorithm used to obtain the tuning policy.

5.3 State Representation: Compact Biomechanical Deviations

The previous section defined a transition as the effect of a parameter update between two walking blocks. The next step is to define what information the learning algorithm receives at each block. Although one could use full ankle trajectories, motor signals, or step-to-step dynamics, this would be poorly suited to the present setting: the dataset is small, and the learned policy must remain interpretable for both AI and biomechanics experts.

For this reason, the state is not defined as a raw time series. Each walking block is instead summarized by a compact vector of *biomechanical features* extracted from the gait cycle. Following previous RL approaches for prosthesis tuning, the state is built from *deviations* between measured gait features and reference gait behavior, rather than from the complete sensor trajectory [47, 48]:

$$s_k = \begin{bmatrix} f_k^1 - f_{\text{ref}}^1 \\ f_k^2 - f_{\text{ref}}^2 \\ \vdots \\ f_k^n - f_{\text{ref}}^n \end{bmatrix}, \tag{5.2}$$

where f_k^i is the i-th feature measured during walking block k, and f_{ref}^i its corresponding reference value. This representation is directly interpretable, since each component measures

how the current prosthetic gait differs from a reference pattern. It also anchors the learning problem in biomechanics: the agent observes not only which parameter configuration was used, but how this configuration affected specific gait events.

5.3.1 State-Design Principles

The state representation was designed according to four principles. First, it must be *compact*, because each real transition requires a human user to walk with the prosthesis, making high-dimensional observations more prone to overfitting and harder to interpret. Second, it must be *biomechanically meaningful*: each feature should correspond to a gait quantity that can be discussed by human experts, such as plantarflexion after heel strike, push-off timing, or assistance during propulsion. Third, it must be *connected* to the tunable parameters d_{plant}, θ_{trig}, and τ_{lim}, so that the observations reflect the parts of the gait cycle where these parameters are expected to act. Finally, it must be compatible with the experimental pipeline, meaning that all selected features must be extractable from the available prosthesis logs and from reference datasets that could be used to define normative behavior.

5.3.2 Gait-Event-Based Features

Feature extraction is organized around key events of the ankle trajectory during level-ground walking showed in Figure 5.1 :

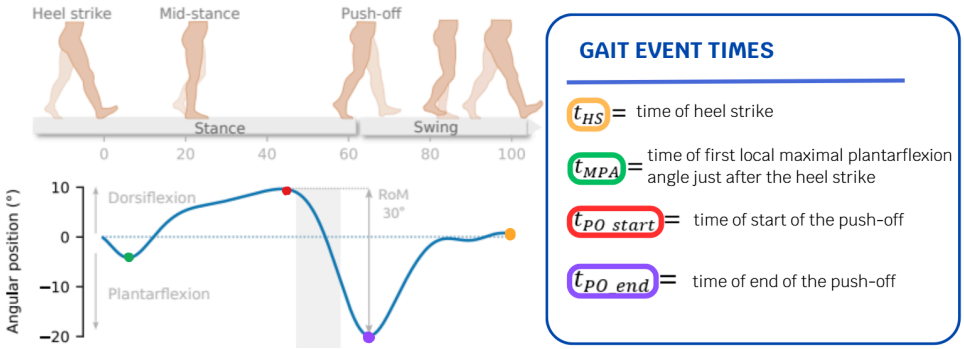


Figure 5.1: Key events of the ankle trajectory during LGW on which the feature states are based.

where t_{HS} is the heel strike timing, t_{MPA} the time of the first local maximum in plantarflexion after heel strike, and $t_{\text{PO,start}}$ and $t_{\text{PO,end}}$ the start timing and end timing of push-off. These events divide the gait cycle into phases directly related to the selected parameters. The resulting state is composed of the six features listed in Table 5.1. The first four features are derived from ankle-angle telemetry and its associated timing information, while the push-off torque and energy features rely on torque and power estimates reconstructed from the recorded control or actuation signals. Their reliability must therefore be checked during preprocessing, especially for cycles where event detection or signal reconstruction is uncertain.

Feature	Description
$f_{1k} = \theta_k(t_{\text{MPA}}) - \theta_k(t_{\text{HS}})$	Plantarflexion drop occurring after heel strike.
$f_{2k} = t_{\text{PO,start}} - t_{\text{MPA}}$	Delay between maximal plantarflexion and the beginning of push-off.
$f_{3k} = \max_{t \in [t_{\text{PO,start}}, t_{\text{PO,end}}]} \lvert\dot{\theta}_k(t)\rvert$	Maximum ankle angular velocity during push-off, used to characterize the dynamical intensity of the motion.
$f_{4k} = t_{\text{PO,end}} - t_{\text{PO,start}}$	Duration of the push-off phase.
$f_{5k} = \max_{t \in [t_{\text{PO,start}}, t_{\text{PO,end}}]} \tau_k(t)$	Maximum assistance torque delivered during push-off, derived from control signals.
$f_{6k} = \int_{t_{\text{PO,start}}}^{t_{\text{PO,end}}} P_k(t)\, dt$	Mechanical energy generated during push-off, computed from the estimated mechanical power over the push-off window.

Table 5.1: State features extracted from gait cycle signals.

5.3.3 Normative References

Defining the state as a deviation requires reference values f_{ref}^i. These references should describe plausible level-ground walking behavior while providing enough biomechanical guidance for learning.

This thesis uses the dataset published by Moreira et al. [55], which contains lower-limb kinematic, kinetic, and EMG recordings from young healthy participants walking on a flat surface at controlled speeds. It is well suited to this work for three reasons: it is specific to locomotion, it provides processed gait-cycle-normalized signals from which event-based quantities can be easily extracted, and it includes several controlled walking speeds, allowing the selected references to remain consistent with the experimental protocol.

The reference values are not used as rigid trajectories that the prosthesis must reproduce exactly. This does not remove the dependence on external normative data, but it limits it to global, event-based gait descriptors rather than imposing full trajectory matching. They provide a biomechanical anchor: a meaningful reference frame in which the algorithm can evaluate whether a given configuration moves the user closer to, or farther from, a typical level-ground gait pattern. The objective is therefore not to force every user toward an identical normative gait, but to give the state and reward a physiologically interpretable reference.

5.4 Action Space: Safe Incremental Updates

Once the state describes the current gait behavior, the next MDP component is the action. In this framework, an action is not a low-level motor command, since torque generation remains handled by the existing ELSA controller. Instead, it represents the *parameter*

update (p) proposed by the learning policy between two consecutive walking blocks:

$$a_k = p_{k+1} - p_k = \begin{bmatrix} d_{\text{plant},k+1} - d_{\text{plant},k} \\ \theta_{\text{trig},k+1} - \theta_{\text{trig},k} \\ \tau_{\text{lim},k+1} - \tau_{\text{lim},k} \end{bmatrix} = \begin{bmatrix} \Delta d_{\text{plant},k} \\ \Delta \theta_{\text{trig},k} \\ \Delta \tau_{\text{lim},k} \end{bmatrix} \in \mathbb{R}^3. \tag{5.3}$$

This follows previous RL work on prosthesis impedance tuning, where the policy learns parameter adjustments rather than absolute impedance values [45, 47]. The agent therefore does not define a complete controller configuration from scratch, but only decides how the current one should be modified.

This incremental formulation is useful for both learning and safety. It preserves the sequential nature of prosthesis tuning, where a prosthetist typically observes the current behavior and applies corrections rather than replacing all parameters at once. It also avoids abrupt changes in assistance timing or intensity, which could feel unstable or uncomfortable for the user. The updated parameter vector is therefore constrained to remain inside predefined admissible bounds described in Table 5.2. This notion of safety should be understood in a constrained-control sense: the learning policy is prevented from proposing parameter values outside the experimentally admissible operating ranges of ELSA, but this does not guarantee that every bounded configuration is comfortable, stable, or well adapted to a given user.

$$p_{\min} \leq p_{k+1} \leq p_{\max}.$$

Parameter	$p_{\min}$	$p_{\max}$	Units
d_{plant}	0	0.5	Nm/(°/s)
θ_{trig}	3	11	°
τ_{lim}	10	60	Nm

Table 5.2: Predefined admissible bounds for the ELSA control parameters.

For a given current configuration p_k, the *admissible action space* becomes:

$$p_{\min} - p_k \leq a_k \leq p_{\max} - p_k. \tag{5.4}$$

The action bounds therefore depend on the current parameter values. A parameter close to its maximum cannot be increased much further, and a parameter close to its minimum cannot be decreased beyond the experimentally validated range. In practice, updates can be written as:

$$p_{k+1} = \text{clip}(p_k + a_k, p_{\min}, p_{\max}), \tag{5.5}$$

where clipping ensures that all recommendations remain inside the predefined operating ranges of the prosthesis.

These bounds are not arbitrary numerical limits. They correspond to experimentally validated and meaningful operating ranges of the selected ELSA parameters, established during the development and experimental characterization of the prosthesis [2]. They encode prior knowledge from the hardware, controller, and previous user experiments.

Finally, bounded incremental actions make the learned policy more interpretable. Recommendations such as "*increase the trigger angle*" or "*reduce the torque limit*" can be

directly related to prosthesis behavior and discussed with clinicians or biomechanists. They also prepare the reward formulation of the next section: since each action has a magnitude, the framework can penalize overly large updates and favor smoother tuning, encouraging gait improvement without unnecessary parameter changes.

5.5 Reward Design: Combining Biomechanics and Human Feedback

The reward function is where the biomechanical and human-centered objectives of the framework are combined. Previous RL approaches for prosthesis impedance tuning mainly relied on measurable quantities, such as deviations from reference gait features and penalties on large parameter changes [45, 47]. This is a natural starting point, because it gives the learning algorithm a clear mechanical direction: reduce deviations from a target gait while avoiding unnecessary control effort.

However, prosthesis tuning is not only a tracking problem. As discussed in Section 2.3.3, configurations that are biomechanically plausible can still feel uncomfortable or poorly adapted to the user, and these subjective dimensions cannot be inferred from onboard sensors alone. Integrating human feedback into the reward therefore supports personalization and directly addresses one of the central questions of this thesis: whether subjective feedback provides useful information for offline prosthesis tuning beyond biomechanical references alone.

The proposed reward instantiates the *evaluative reward shaping* formulation introduced in Section 2.3.2 (Eq. (2.10)) by combining a prosthesis-centered term, based on gait deviations and action magnitude, with a human-feedback term, based on comfort and perceived assistance:

$$r_k = (1-\lambda_h) r_k^{\text{prosthesis}} + \lambda_h r_k^{\text{human}}, \tag{5.6}$$

where $\lambda_h \in [0,1]$ controls the relative contribution of human feedback.

5.5.1 Prosthesis-centered reward

The prosthesis-centered reward is defined as:

$$r_k^{\text{prosthesis}} = -\sum_{i=1}^{n} w_i \left(\tilde{s}_k^i\right)^2 - a_k^\top R_a a_k. \tag{5.7}$$

The first term penalizes normalized deviations from the reference gait features ($\tilde{s}_k^i$) introduced in Section 5.3. The normalization allows features expressed in different units to contribute on a comparable scale, while the weights w_i define their relative importance. For instance, push-off timing can be emphasized more strongly if it is considered central to the tuning objective. The normalization details are set out in Section 6.4.

The second term penalizes large parameter updates. The matrix R_a sets how costly it is to modify each tuned parameter, encouraging smoother policies and discouraging aggressive or unnecessary changes in prosthesis behavior. It therefore acts as a safety-oriented regularization of the action space.

5.5.2 Human-feedback reward

The human-feedback reward is defined as:

$$r_k^{\text{human}} = \alpha \tilde{C}_k + (1-\alpha)\tilde{A}_k, \tag{5.8}$$

where $\tilde{C}_k$ and $\tilde{A}_k$ are the normalized comfort and perceived assistance scores reported for walking block k (see details of the normalization process in Section 6.3). The coefficient $\alpha \in [0,1]$ controls their balance: values close to one prioritize comfort, while values close to zero prioritize assistance.

This term is deliberately simple. The goal is not to build a complete model of human preference, which would be unrealistic with the limited feedback available, but to introduce a structured and interpretable subjective signal. Comfort captures whether the configuration feels acceptable to the user, while perceived assistance captures whether the prosthesis feels helpful during walking. Both are relevant for tuning and cannot be fully inferred from ankle kinematics alone.

5.5.3 HITL Reward Shaping

The hybrid reward can be interpreted through λ_h, which here plays the role of the general weighting coefficient introduced in Section 2.3.2: $\lambda_h = 0$ recovers a purely biomechanical reward, $\lambda_h = 1$ a purely human-driven one, and intermediate values define the HITL setting targeted in this thesis. This design is intentionally conservative. The structural difficulties of scalar human input identified in Section 2.3.2, sparsity, noise, and credit-assignment ambiguity, make it unwise to treat human feedback as perfect ground truth. Conversely, the biomechanical reference is not assumed to fully define optimal walking, since a normative gait pattern may not match what a specific user finds comfortable or useful.

In this sense, the reward is also an experimental tool. By varying λ_h, α, the feature weights w_i, and the action-cost matrix R_a, the framework can assess which reward components actually shape the learned policy. This is essential for evaluating the role of human feedback in the small-data, offline setting considered in this thesis.

5.6 Offline Policy Iteration for Learning the Tuning Policy

The final component of the framework is the learning algorithm that transforms the offline transition dataset into a tuning policy. This thesis uses *Offline Policy Iteration* (OPI), adapted from previous work on robotic prosthesis parameter tuning [45, 47]. The goal here is not to rederive the full algorithm, but to explain why it is suitable for ELSA and how it is adapted to the proposed HITL formulation.

OPI matches the practical constraints of this application. The dataset is limited because each transition requires a human user to walk with the prosthesis under a given configuration. Direct online learning would also expose the user to intermediate policies that may be poorly tuned. At the same time, the action space is continuous but low-dimensional and bounded, since the agent only updates three interpretable control parameters. These properties make OPI more appropriate than deep RL for this first framework: it is lightweight, data-efficient, and compatible with offline training before any deployment on the prosthesis.

5.6.1 From Reward Maximization to Cost Minimization

The framework is expressed in terms of reward, as is standard in reinforcement learning. However, the OPI formulation used here follows the prosthesis tuning literature and is written as a *cost-minimization problem* [45, 47]. The algorithm therefore searches for actions that minimize an immediate cost and the future costs expected under the current policy.

A direct conversion $U_k = -r_k$ would preserve the reward ordering, but could produce negative immediate costs. This is undesirable in the OPI formulation used here, which assumes a positive quadratic cost structure. To preserve the ordering of transitions while ensuring positive costs, a constant shift is applied:

$$U_k = c_{\text{shift}} - r_k, \tag{5.9}$$

with

$$c_{\text{shift}} = \max_{j \in \mathcal{D}} r_j + \varepsilon, \tag{5.10}$$

where $\mathcal{D}$ is the offline transition dataset and $\varepsilon > 0$ a small margin. The best observed reward then receives a cost slightly above zero, while lower rewards receive larger costs. The transformation does not change which transitions are preferable. It only makes the reward compatible with OPI cost minimization.

5.6.2 Quadratic Approximation of the Action-Value Function

Once the instantaneous cost U_k is defined, OPI learns an action-value function $Q(s, a)$ (Section 2.2.1), here interpreted as a *cost-to-go*: the long-term cost of applying action a in state s and following the current policy thereafter. Because the true human-prosthesis dynamics are unknown, this cost-to-go cannot be computed exactly and is replaced by a parametric approximation (Section 2.2.3). OPI uses a quadratic form:

$$\hat{Q}(s, a) = \mu(s, a)^\top S \mu(s, a), \tag{5.11}$$

where $\mu(s, a)$ contains the state and action variables, and S is the learned quadratic coefficient matrix. Concretely, S encodes how much each state deviation, each parameter update, and each state-action interaction contributes to the estimated long-term cost. This follows the OPI literature, where a quadratic approximation keeps policy evaluation tractable and policy improvement solvable as a quadratic optimization problem [45, 47].

The quadratic form is simple but expressive: it can penalize state deviations, penalize actions, and represent how the effect of an action depends on the current state. In block form:

$$\hat{Q}(s, a) = s^\top S_{ss} s + 2 s^\top S_{sa} a + a^\top S_{aa} a. \tag{5.12}$$

The first term ($s^\top S_{ss} s$) represents the cost of the current biomechanical state, the last term ($a^\top S_{aa} a$) the cost of changing the parameters, and the middle term ($2 s^\top S_{sa} a$) the state-action interaction. This interaction is essential: it allows the policy to learn, for example, that increasing the trigger angle may help when push-off starts too early, but not when it is already delayed.

5.6.3 Policy Evaluation and Policy Improvement

OPI is a direct instantiation of the policy iteration loop of Section 2.2.2, with the approximate evaluation step of Section 2.2.3 and a cost-minimization convention in place of reward maximization. During policy evaluation, the current policy is fixed and S is fitted so that the quadratic approximation satisfies the Bellman relation (Eq. 2.4) translated to costs:

$$\hat{Q}(s_k, a_k) \approx U_k + \gamma\, \hat{Q}\big(s_{k+1}, \pi(s_{k+1})\big), \tag{5.13}$$

where $\gamma \in [0, 1]$ now controls the importance of future costs. As introduced in Section 2.2.3, S is obtained by minimizing the squared Bellman residual on the dataset (Eq. 2.7, written here on costs).

In the original OPI/PICE formulation [45, 47], the full matrix S is constrained to be positive semi-definite (PSD), ensuring a globally non-negative quadratic value function. In this thesis, the constraint is restricted to the action-action block:

$$S_{aa} \succeq 0, \tag{5.14}$$

This keeps policy evaluation convex while avoiding an unnecessarily strong constraint on the full state-action surface.

The reason for restricting the PSD constraint to S_{aa} appears in the policy improvement step. Following the greedy step of policy iteration (Eq. 2.6), transposed to cost minimization under a bounded action set:

$$\pi_{\text{new}}(s) = \arg\min_{a \in \mathcal{A}(s)} \hat{Q}(s, a), \tag{5.15}$$

the state s is fixed. So, from Eq. (5.12), $s^\top S_{ss} s$ is constant and the linear state-action term $2s^\top S_{sa} a$ only shifts the optimum. The curvature of the action optimization is therefore entirely controlled by $a^\top S_{aa} a$, and constraining $S_{aa} \succeq 0$ alone is sufficient to make policy improvement convex in the action.

The admissible set $\mathcal{A}(s)$ corresponds to the bounded parameter updates defined in Section 5.4. Policy improvement is thus a bounded quadratic program: for each gait state, the policy selects the safe parameter update predicted to minimize long-term cost. By constraining only S_{aa}, the model preserves the convexity needed for action selection while keeping more freedom to learn the state-action interactions in S_{sa}, which are precisely what allow parameter updates to adapt to gait features.

The evaluation and improvement steps are repeated until the cost approximation and policy stabilize. The output is a greedy tuning policy mapping each gait state to an incremental update of the ELSA parameters. OPI is therefore not used as a black-box controller, but as a structured offline method for learning an interpretable first policy from limited data. It preserves the existing ELSA controller, uses bounded updates, and offers a practical compromise between learning capacity, data efficiency, and experimental safety.

5.7 Conceptual Summary of the Full Framework

Figure 5.2 and Table 5.3 summarize the proposed HITL-RL framework. During each walking block, the existing ELSA controller drives the prosthesis under a fixed parameter vector. The resulting gait signals and user feedback are then transformed into the three core MDP

components of the framework: compact biomechanical states, bounded parameter-update actions, and a hybrid reward combining gait deviations with subjective feedback. After conversion of this reward into a positive cost, the resulting transitions form the offline dataset used by OPI to learn an interpretable bounded tuning policy.

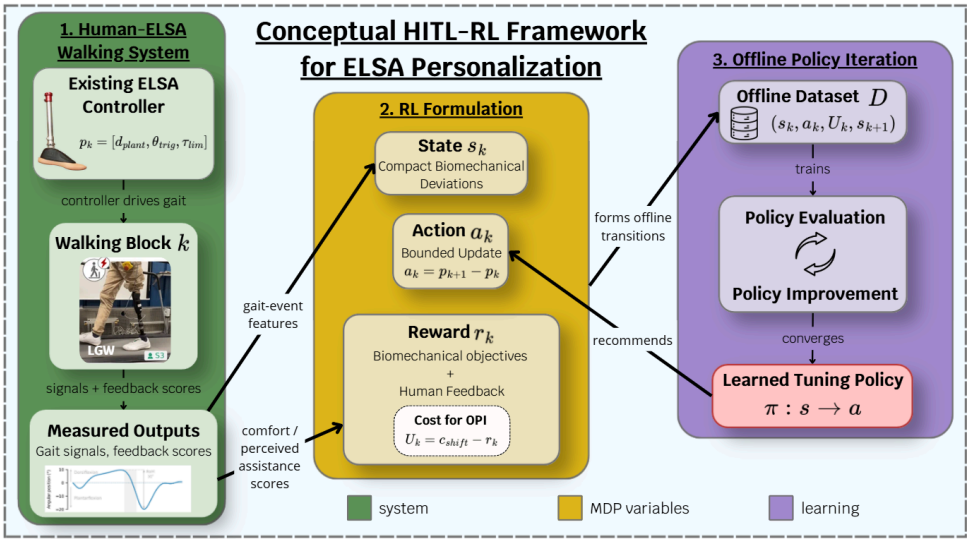


Figure 5.2: Conceptual overview of the proposed HITL-RL framework for ELSA personalization. The human-ELSA walking system generates gait signals and subjective feedback, which are mapped to state, action, and reward variables. These transitions form the offline dataset used by OPI to learn a bounded tuning policy.

This chapter has therefore defined the framework at the conceptual level: what the agent observes, what it can change, how each transition is evaluated, and how these elements are combined to learn parameter-update recommendations. The next chapter describes how this formulation was implemented in practice.

RL component	Instantiation in this thesis
Agent	Offline tuning policy
Environment	Human-ELSA walking system
Decision level	Walking block / parameter transition
State	Deviations from reference gait features
Action	Bounded incremental parameter update
Reward	Biomechanical objective + human feedback
Learning algorithm	Offline Policy Iteration
Output	Policy recommending parameter updates

Table 5.3: Conceptual instantiation of the RL components in the proposed ELSA framework.

6

Computational Framework and Offline Validation

6.1 Overview of the Computational Pipeline

Following the conceptual framework introduced in Chapter 5, this chapter describes its practical implementation by focusing on the architectural choices that make the code *modular*, *interpretable*, and *reusable*.

The implementation is organized into four Python modules, summarized in Table 6.1. This separation mirrors the structure of the framework: ELSA-specific configuration, reward computation, offline learning, and validation are handled independently. As a result, the reward formulation, tuning parameters, or hyperparameters can be modified without changing the core OPI algorithm introduced in Section 5.6.

Module	Main role
`elsa_configOPI.py`	Defines the ELSA-specific parameter bounds, state and action dimensions, and model hyperparameters.
`reward.py`	Computes the transition reward by combining biomechanical deviations with comfort and perceived assistance feedback.
`OPI_core.py`	Implements the offline OPI procedure, including normalization, cost conversion, policy evaluation, policy improvement, and action prediction.
`OPI_validation.py`	Provides offline validation tools to assess Bellman consistency, action plausibility, policy improvement, robustness, and cross-validation behavior.

Table 6.1: Main implementation modules and their role in the proposed framework.

The following sections describe these modules, focusing on the implementation choices needed to understand and reproduce the proposed framework. In practice, the pipeline

takes as input an offline transition dataset with the computed rewards translated into costs, trains the OPI model, and outputs both the learned matrix S and diagnostic metrics used in Chapter 8.

6.2 Prosthesis-Specific Configuration and Safety Constraints

The module `elsa_configOPI.py` defines the interface between the ELSA prosthesis and the generic OPI algorithm. Its role is not only to store constants, but to make the ELSA-specific assumptions explicit and consistent across the full pipeline. The same configuration is used by reward computation, action normalization, policy improvement, and validation, reducing the risk of inconsistent parameter bounds or scaling conventions across modules.

The configuration entries can be grouped into three categories. First, prosthesis-specific entries define the tuned parameters, their admissible ranges, and the dimensions of the state and action spaces. Second, learning-related entries define the quantities that control the OPI procedure itself, such as the discount factor, the maximum number of iterations, the convergence tolerance, and the regularization strength. Third, reward and validation entries define how biomechanical and human-feedback signals are weighted, how action changes are penalized, and how unstable or uninformative training iterations are detected. Table 6.2 summarizes the main entries and their role in the computational pipeline.

Hyperparameter	Role
`n_features`, `n_actions`	Set the dimensions of the state vector and action vector used by the OPI model.
`gamma`	Discount factor used in the Bellman cost-to-go equation during policy evaluation.
`n_iterations`, `tol`	Set the maximum number of OPI iterations and the convergence tolerance on the Frobenius norm between successive S matrices.
`lambda_h`, `alpha`	Set the reward-shaping weights: λ_h balances prosthesis-centered and human-feedback rewards, while α balances comfort and perceived assistance.
`feature_weights`	Weight the normalized gait-feature deviations in the prosthesis-centered reward.
`R_a`, `R_a_per_dim_scale`	Define the action-cost matrix and optional per-parameter scaling used to penalize large updates of d_{plant}, θ_{trig}, and τ_{lim}.
`SDP_eps`, `lambda_reg`	Set the numerical margin used in the positive semi-definite constraint and the Tikhonov regularization strength used during policy evaluation.
`cost_shift_margin`, `use_bias`	Support the reward-to-cost transformation and the optional constant bias term in the quadratic cost-to-go approximation.
`cycle_*`, `stagnation_*`	Control early-stopping checks for plateaus, limit cycles, or lack of progress during OPI iterations.

Table 6.2: Main configuration entries defined in `elsa_configOPI.py`.

This configuration module therefore acts as the interface between the ELSA-specific tuning problem and the generic OPI implementation.

6.3 Reward Computation and Human Feedback Integration

The module `reward.py` implements the reward formulation defined in Section 5.5. Its role is to compute the transition-level reward used by OPI, without introducing a new objective.

The main implementation choice concerns human feedback. Comfort and perceived assistance scores are normalized with a *per-subject z-score* before being included in the reward. In the present dataset, which contains a single participant (cf. Section 7.1.3), this expresses feedback as a relative preference within that participant. In future multi-subject datasets, the same procedure would also reduce differences in individual scale use: some users may systematically give higher or lower ratings, independently of the actual prosthesis configuration. Standardization therefore expresses feedback as a relative preference within each user, which is a common strategy when subjective ratings may be affected by response-style effects [56].

Keeping the reward in a separate module also makes the framework easier to test and extend. Different reward variants can be evaluated by modifying this module or its configuration, without changing the core OPI algorithm.

6.4 Offline Policy Iteration Core

The module `OPI_core.py` contains the generic implementation of the Offline Policy Iteration algorithm introduced in Section 5.6. Its role is to transform an offline transition dataset into a *greedy tuning policy*, without depending on ELSA-specific definitions beyond the configuration object described in Section 6.2. In practice, the training pipeline follows six steps:

1. load the offline transitions (s_k, a_k, r_k, s_{k+1});
2. normalize states and actions;
3. convert rewards into positive costs, as described in Section 5.6.1;
4. initialize the policy with a zero-action policy;
5. alternate policy evaluation and policy improvement;
6. store the final quadratic matrix S and the corresponding greedy policy.

The following paragraphs only discuss the implementation choices that were necessary to make this procedure numerically reliable on a small real dataset.

State and action normalization. Before policy evaluation, biomechanical features are mapped to $[-1, 1]$ using *min-max normalization*. The normalizer is fitted jointly on current and next states, so both sides of the Bellman relation share the same coordinate system. This implementation choice is consistent with the prosthesis-centered reward introduced

in Eq. (5.7): the normalized feature deviations $\tilde{x}_k^i$ are used to prevent features expressed in different physical units from dominating the reward, while the OPI normalization further ensures that the state coordinates used in the Bellman regression remain numerically comparable. This follows previous OPI work for prosthesis tuning, where normalization to $[-1, 1]$ was used to reduce ill-conditioning during convex optimization [45].

Actions are normalized separately by dividing each incremental parameter update by half of the corresponding physical range. This mirrors the role of the action penalty $a_k^\top R_a a_k$ in Eq. (5.7): parameter updates must be compared on a common scale before their magnitude can be meaningfully penalized. This keeps the no-update action centered at zero while making damping, trigger angle, and torque-limit updates numerically comparable. Without this scaling, differences in units and physical ranges could bias both the reward and the policy-improvement step toward some action dimensions.

Initial zero policy. The initial policy is the *zero-action policy*, which proposes no parameter change. This conservative choice is always feasible, avoids introducing an arbitrary preference before learning, and corresponds to keeping the current prosthesis configuration unchanged. For an offline first implementation intended for a human-prosthesis system, this is safer than starting from an aggressive or random policy.

Bias term in the quadratic approximation. Because rewards are converted into positive costs through a constant shift (cf. Section 5.6.1), the learned cost-to-go may contain a global offset independent of the current state and action. To represent it, the joint vector used in the quadratic approximation can include a *constant bias term*:

$$\mu(s, a) = [1; s; a]. \tag{6.1}$$

This entry acts as a baseline cost level. It allows the quadratic model to represent this global offset directly. Without it, the model would have to absorb the constant shift into the state and action coefficients, potentially distorting the learned matrix S and the resulting greedy actions.

Tikhonov regularization. A *Tikhonov regularization* term is added during policy evaluation to improve numerical robustness [57]. Since the matrix S contains many coefficients while the available dataset is limited, a pure least-squares fit may overfit the observed transitions and produce large, unstable coefficients. This could lead to unrealistic action recommendations.

To reduce this risk, policy evaluation minimizes the Bellman residual together with a penalty on the size of the matrix:

$$\min_S \sum_k e_k(S)^2 + \lambda_{\text{reg}} \|S\|_F^2, \tag{6.2}$$

where $e_k(S)$ is the Bellman residual of transition k, and $\|S\|_F^2$ penalizes large entries in S. This follows the principle: among matrices that explain the data similarly well, the algorithm favors the smoother one with smaller coefficients [57, 58]. Here, the goal is not to change the OPI objective, but to make the learned cost surface less sensitive to noise and less likely to generate extreme actions.

Convex optimization with CVXPY. Policy evaluation and policy improvement are implemented as convex optimization problems using *CVXPY* [59]. This keeps the implementation close to the mathematical formulation, with explicit objectives, variables, and constraints. Policy evaluation solves the constrained least-squares problem used to estimate S, while policy improvement solves the bounded quadratic program used to select the greedy action when the unconstrained analytic solution is not feasible.

The implementation tries several solvers in sequence, starting with CLARABEL and falling back to SCS when needed [60]. This improves robustness to occasional solver failures or numerical conditioning issues while preserving the same optimization problem.

Convergence monitoring and early stopping. Strict convergence is monitored through the *Frobenius norm* between two successive matrices:

$$\|S^{(i+1)} - S^{(i)}\|_F. \tag{6.3}$$

This criterion directly measures whether the learned quadratic approximation is still changing across OPI iterations, and is consistent with previous prosthesis-tuning implementations [45]. Ideally, training stops when this norm falls below the prescribed tolerance.

However, small and noisy offline datasets can produce stagnation, oscillations, or small limit cycles instead of strict convergence. The implementation therefore includes early-stopping checks for plateaus, cycles, and lack of progress. These checks do not modify the policy iteration equations. They simply avoid unnecessary iterations once the learned matrix no longer improves meaningfully. When this occurs, the best matrix observed during training can be retained.

Overall, `OPI_core.py` isolates the generic learning mechanism from prosthesis-specific choices. It remains a direct implementation of the formalism described in Section 5.6, but includes the numerical protections required to make OPI usable on a limited and noisy human-prosthesis dataset.

6.5 Offline Evaluation and Validation Pipeline

The module `OPI_validation.py` implements the offline evaluation pipeline used after training. Its purpose is to answer a simple but essential question: is the learned policy reliable enough to be considered further? Since this thesis does not include online clinical testing or a complete simulator of the human-prosthesis system, this validation cannot prove real-world safety. Instead, it provides a data-driven safety net based only on the offline dataset and the learned matrix S. This is particularly important in a HITL-RL setting, where learned behaviors should remain interpretable, stable, and inspectable before deployment with a human user [15].

Bellman consistency. The first check measures, on each transition of the dataset, the Bellman residual introduced in Section 2.2.3 (Eq. 2.7), written here on costs:

$$\varepsilon_k = \hat{Q}(s_k, a_k) - \gamma \hat{Q}(s_{k+1}, \pi(s_{k+1})) - U_k. \tag{6.4}$$

A small residual, summarised by its RMSE [1] and relative RMSE [2], indicates that the learned matrix S is internally consistent with the OPI cost-to-go logic on the offline transitions. It does not prove policy optimality.

Action distribution. The second check compares the actions proposed by the learned policy with those observed in the dataset. For each parameter, it reports the mean and variability of predicted updates, their distance from the dataset actions, and their proximity to physical bounds. This matters because a low Bellman error can still hide an implausible policy. A policy that systematically pushes parameters to their limits may be exploiting constraints rather than learning a meaningful tuning strategy, while one that almost always predicts zero updates may be too conservative to be useful.

Policy improvement. The third check verifies that the greedy action proposed by the learned policy actually reduces the predicted cost compared with the action observed in the dataset:

$$\hat{Q}(s, \pi(s)) \leq \hat{Q}(s, a_{\text{data}}). \tag{6.5}$$

This condition should hold for all samples, because the policy is defined as the action that minimizes the learned quadratic cost function. If it fails, the issue is mainly numerical or algorithmic rather than biomechanical, indicating an inconsistency between the learned Q-function and the policy improvement step.

Stability and robustness. The fourth check evaluates the sensitivity of the policy to small variations in gait features. Random perturbations are added to the state, and the resulting change in predicted action is measured. This reflects the fact that real gait features are noisy and naturally variable across steps or sessions. A stable policy should not change its recommendation drastically for a small input variation. High sensitivity values or frequent action sign changes would suggest that the learned policy is too fragile for a human-prosthesis setting.

Cross-validation. The final check estimates whether the learned model generalizes beyond the data used for fitting. The dataset is split into training and validation folds, and a fresh OPI model is trained on each training subset. The validation Bellman error is then compared with the training error. A large gap between training and validation errors would suggest overfitting.

These diagnostics play complementary roles: Bellman consistency and policy improvement check algorithmic coherence, whereas action distribution, robustness, and cross-validation assess whether the learned policy remains plausible and stable enough to justify further investigation.

Overall, this module evaluates more than a training loss. Because the framework is offline and applied to a human-prosthesis interaction, the learned policy must also be checked for coherence, action plausibility, and stability. These diagnostics do not

[1] RMSE = Root Mean Squared Error
[2] relative RMSE = RMSE divided by the observed value mean

replace future clinical validation, but they provide a necessary offline screening step before considering deployment on the real ELSA prosthesis.

6.6 Reproducibility and Code Availability

The complete implementation of the proposed HITL-RL framework is available in a private Git repository[3]. By separating configuration, reward computation, offline learning, and validation, the implementation can be reused to test different reward designs, selected features, hyperparameters, or ELSA parameter sets without modifying the full algorithm. This modular structure also facilitates future extensions to additional users, larger datasets, and other locomotion modes.

The next chapter builds on this implementation to describe the experimental protocol, the construction of the offline dataset, and the analysis of the obtained results.

[3] `https://git.immc.ucl.ac.be/marsanodacos1/rl_poc.git`. Access can be granted upon request.

7

Experimental Data and Evaluation Protocols

After defining the conceptual and practical design of the proposed framework, this chapter describes how the RL-compatible dataset was collected and how the learned policies were evaluated offline.

7.1 RL-Compatible Data Collection

7.1.1 Preliminary Analysis of Historical ELSA Data

Since ELSA had already been developed and validated at UCLouvain, the first step was to assess whether previous experimental recordings could be reused to build the offline transition dataset required by the proposed framework. A detailed review of these recordings (cf. Appendix A) showed that they were useful to understand the prosthesis behavior, the manual tuning process, and the influence of key parameters on gait. They also confirmed that ELSA had been tested across several campaigns with prosthesis users, prototypes, tasks, and control configurations [2].

However, these recordings were not structured as an offline reinforcement learning dataset. They did not provide consistent tuples of the form (s_k, a_k, r_k, s_{k+1}), as required by Section 5.2. Parameter changes were mostly performed manually between recordings, hardware and firmware versions changed across campaigns, and subjective feedback was limited to sparse qualitative notes rather than systematic block-level scores. These data could therefore not be used directly to train the OPI policy.

7.1.2 Dedicated Data Collection Protocol

To obtain data compatible with the offline formulation of Chapter 5, a dedicated experiment was conducted in collaboration with Luana Marsano Da Costa Nunes, a PhD student. The goal was to collect structured walking blocks in which ELSA operated under controlled parameter configurations, while recording both prosthesis telemetry and user feedback.

Due to the limited experimental window, this first collection was performed with a healthy participant walking with ELSA through an adapter. Although this setup does

not reproduce all biomechanical characteristics of amputee gait, it provided a controlled first approximation of human-prosthesis interaction under real prosthesis operation. The participant walked in successive blocks of 60 steps. Within each block, the selected ELSA parameters remained fixed, while updates between blocks were randomly generated within the admissible bounds of Section 5.4. This allowed different regions of the parameter space to be sampled while preserving the prosthesis operating constraints.

For each block, prosthesis telemetry was recorded through ELSA's acquisition pipeline and later used to extract the gait events and biomechanical features defined in Section 5.3.2. Subjective feedback was collected during the last 15 steps of each block, , so that the participants could first experience the current parameter configurations before rating it. Two scores were reported on a 0–10 scale: general walking comfort and perceived assistance. These scores form the human-feedback component of the reward introduced in Section 5.5.2.

A real transition was then defined between two consecutive walking blocks. If block k was performed with parameter vector p_k, and block $k+1$ with parameter vector p_{k+1}, the action was defined as the incremental update: $a_k = p_{k+1} - p_k$. The transition therefore links the gait state observed under one configuration to the gait state observed after applying the next one. This block-level definition is consistent with the offline tuning formulation introduced in Section 5.2.

7.1.3 Hardware Constraints and Final Dataset Scope

However, the experimental campaign was strongly affected by hardware constraints on ELSA. Several mechanical and wear-related issues occurred during the collection period, independently of the learning framework developed in this thesis. These incidents reduced the available testing window and limited the amount of exploitable walking data.

The final dataset therefore contains data from one participant only, with 52 real block-to-block transitions. This value corresponds to the number of independent parameter updates actually tested on the prosthesis. Although each walking block contains multiple gait cycles, the true experimental information remains limited by the number of distinct parameter transitions. This number, rather than the number of gait cycles, defines the amount of independent experimental evidence available for learning parameter-update effects.

7.1.4 Dataset Augmentation Through All-to-All Transitions

The limited number of real transitions raised an important methodological issue for policy learning. OPI does not learn a simple rule for each tuned parameter. It estimates a quadratic matrix S describing how state features, actions, and their interactions contribute to the long-term cost, as explained in Section 5.6.2. With 6 features, 3 actions, and 1 bias term, this represents approximately 55 free coefficients, while only 52 independent block-to-block transitions were available. The model therefore had almost as many degrees of freedom as real observations, making the regression weakly identifiable and potentially sensitive to noise. To mitigate this numerical fragility, an *all-to-all transition dataset* was constructed from the natural cycle-to-cycle variability within each walking block: instead of representing each block by a single averaged gait state, every valid gait cycle from block k was paired with every valid gait cycle from block $k+1$. All pairs share the same

block-level action, since the parameter update is identical, but differ in their current and next gait-cycle features:

$$\mathcal{D}_{\text{all}} = \left\{ (s_i^{(k)}, a_k, r_k, s_j^{(k+1)}) \mid s_i^{(k)} \in B_k,\ s_j^{(k+1)} \in B_{k+1} \right\}. \tag{7.1}$$

In the final dataset, this procedure produced 83,200 pseudo-transitions. This augmentation does not create new independent experimental trials: the number of real parameter transitions remains 52. Its purpose is more limited. It exposes the model to the natural variability already present within the recorded gait cycles and makes policy evaluation numerically less fragile than with one averaged transition per block pair. The all-to-all dataset should therefore be interpreted as a practical mitigation strategy for small-data offline learning, not as a substitute for collecting more users and real transitions. Consequently, the augmented dataset improves numerical conditioning, but it should not be interpreted as increasing the amount of independent experimental evidence by the same factor.

7.2 Computational Evaluation Protocol

After constructing the offline transition dataset, the learned policy was evaluated using the validation pipeline described in Section 6.5. The evaluation was organized around a baseline configuration, used as the reference for all comparisons. From this baseline, a one-factor-at-a-time ablation study was performed across six hyperparameter axes covering the biomechanical cost shape, the global OPI behavior, and the human-feedback reward structure. The baseline values and tested variants are summarized in Table 7.1.

Hyperparameter	Baseline	Tested values
Feature weights W	Uniform	Inverse-variance preset
Action-cost matrix R_a	$(1,1,1)$	$(1,1,5)$, $(1,3,5)$, $(1,5,5)$
Discount factor γ	0.90	0.80, 0.85, 0.95
Regularization λ_{reg}	0	10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 10
Human/prosthesis balance λ_h	0.50	0, 0.25, 0.75, 1
Comfort/assistance balance α	0.50	0, 0.25, 0.75, 1

Table 7.1: Baseline configuration and ablation grid used to evaluate the proposed OPI-based HITL-RL framework. For R_a, the triplets denote multiplicative scales applied respectively to $(d_{\text{plant}}, \theta_{\text{trig}}, \tau_{\text{lim}})$.

All configurations were trained on the same all-to-all transition dataset and evaluated with the same validation procedure. For each ablation, only the studied component was modified, while the reward formulation, OPI training pipeline, and validation checks remained unchanged. This makes the comparison interpretable: differences observed in the next chapter can be attributed to the tested hyperparameter changes rather than to changes in the learning or evaluation protocol. This one-factor-at-a-time design makes the analysis interpretable, but it does not explore interactions between hyperparameters.

This chapter has defined the experimental basis of the study: the origin and limitations of the dataset, the all-to-all augmentation strategy used to mitigate data scarcity, and the computational protocol used to screen the fisrt iteration of the model under controlled offline conditions. The next chapter builds on this setup to analyze the learned policies and interpret the effect of each hyperparameter family.

8

Results and Interpretation

This chapter analyzes the most relevant results obtained from the experimental protocol introduced in the previous chapter in order to address the research questions that this master's thesis seeks to answer. The analysis begins by examining the baseline results and identifies the findings that will guide the structure of the remainder of the chapter, leading to an in-depth analysis of the results from the ablation axes with the greatest impact.

8.1 The baseline: a working policy with two limitations

The baseline configuration of Chapter 7 was trained on the all-to-all dataset and evaluated with the validation pipeline of Section 6.5. It serves as the reference against which every subsequent ablation is compared. This section first reports what the baseline learns correctly, then exposes the two patterns that prevent its use as-is on the prosthesis.

8.1.1 What the baseline learns

Three indicators establish that the baseline qualifies as a valid first OPI instance on ELSA.

Convergence. Figure 8.1 reports the Frobenius distance between two successive iterates of S, the convergence indicator of the original OPI formulation [45]. The trajectory drops sharply during the first four iterations, then settles into a small-amplitude oscillation around 5-7. This behaviour is consistent with the expected behaviour of approximate policy iteration: because a quadratic cost-to-go cannot exactly represent the true cost surface of the human-prosthesis system, a residual approximation error may remain at each evaluation step, so the iterates can stabilize near a representable cost-to-go rather than converge to a strict fixed point [8]. The early-stopping criterion of Section 6.4 detects this regime and retains the iterate closest to its predecessor (a more detailed derivation of the neighborhood convergence phenomenom is provided in Appendix B).

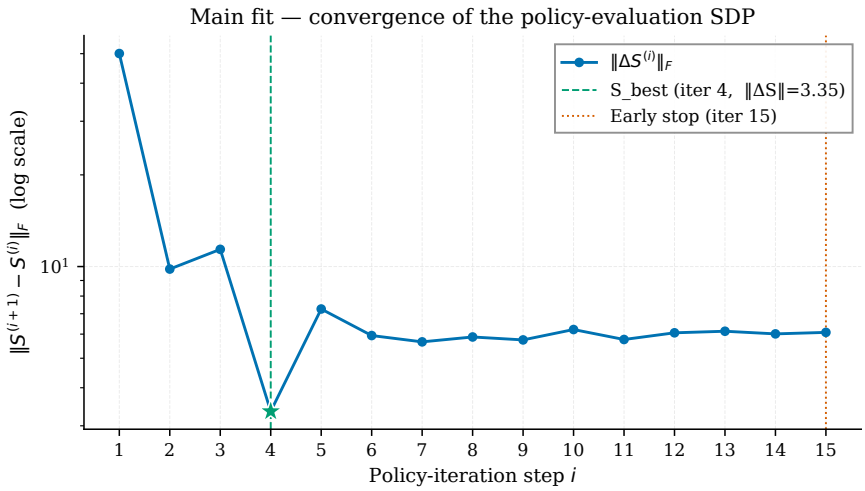


Figure 8.1: Baseline OPI convergence diagnostic. The green marker indicates the retained iterate ($i = 4$, $\|\Delta S\|_F = 3.35$), while the dashed orange line marks the early-stopping iteration ($i = 15$).

Bellman residual. At the retained matrix, the Bellman residual has an RMSE of 1.18 for a mean immediate cost of 6.4, that is a relative error of 18.5%: the predicted long-term cost of an observed transition agrees with the sum of its immediate cost and the predicted cost at the next state up to about one fifth in relative terms.

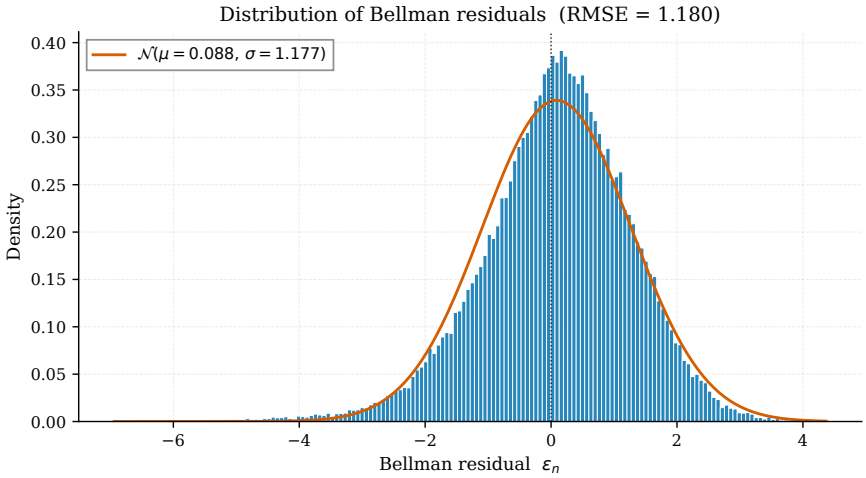


Figure 8.2: Bellman-residual distribution for the retained baseline matrix. The histogram is compared with a fitted Gaussian distribution $\mathcal{N}(\mu = 0.088, \sigma = 1.177)$, with RMSE = 1.180.

Figure 8.2 shows that this residual is unimodal, almost symmetric and closely tracks a Gaussian centered near zero (mean 0.09, two orders of magnitude smaller than the RMSE), with virtually no transition beyond $|\varepsilon| = 5$. The error mass therefore appears to be mainly

variance around zero rather than a systematic bias. This is consistent with approximation error from a low-capacity quadratic surrogate model on a noisy real-world dataset [11, 12], rather than suggesting a localized modeling failure on a few pathological transitions.

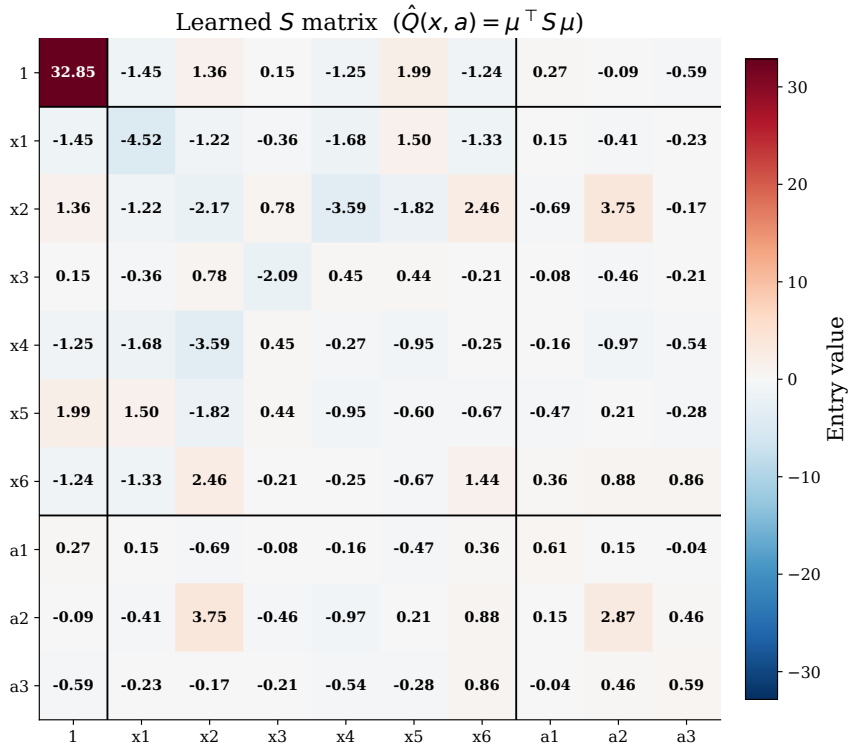


Figure 8.3: Learned S matrix on the baseline. Black lines separate the bias slot (row and column 1), the state block ($x_1, \ldots, x_6$) and the action block ($a_1 = \Delta d_{\text{plant}}$, $a_2 = \Delta\theta_{\text{trig}}$, $a_3 = \Delta\tau_{\text{lim}}$). The action–action block S_{aa} in the bottom-right 3×3 corner is positive semi-definite by construction, as enforced by the SDP constraint. The reason behind the large value (32.25) in the bias slot ($S[1,1]$) is detailed in the Appendix E.

Beyond these aggregated metrics, the learned matrix S itself, shown in Figure 8.3, carries an encouraging interpretability signal. Each entry of S is a coefficient in the quadratic cost-to-go formula: diagonal entries weigh how strongly an individual variable contributes to the predicted cost on its own, off-diagonal entries quantify how a pair of variables jointly amplifies or cancels the cost (a positive coupling adds to the cost when both move in the same direction, a negative one partially compensates). Two entries stand out in the action-related blocks of the learned surrogate: a marked diagonal coefficient on a_2 in S_{aa}, indicating that changes in the push-off trigger are assigned the largest action-related cost within this fitted model, and a coupling of similar order on the x_2-a_2 entry, suggesting that the policy uses the MPA-to-push-off timing feature (cf. Section 2.1) when selecting trigger-angle updates. This is the kind of clinically interpretable structure sought when the framework was designed, and suggests that the regression has not collapsed into a flat or purely noise-dominated object.

8.1.2 Limitation 1: Action saturation

A first concern emerges when the greedy actions are compared, state by state, against the admissible bounds defined in Section 5.4. Figure 8.4 reports the per-parameter saturation rate. The picture is heavily asymmetric: θ_{trig} almost never saturates (0.5%), d_{plant} does so in 16.8% of states (with a mild lean towards the lower bound), but τ_{lim} is pinned to its upper bound in 68.4%. Whenever the policy has room to increase torque, it drives τ_{lim} to its maximum admissible value.

Two complementary readings explain this behavior. Geometrically, since $S_{aa} \succ 0$ by construction, the learned cost-to-go has a single unconstrained minimum per state. For τ_{lim}, that minimum sits above the maximum admissible torque in most states, so the policy returns the closest feasible value, which the validation pipeline records as saturation. A plausible explanation lies in the choice of references: features f_5 and f_6 are scored against young able-bodied gait [55], and recent characterisations of state-of-the-art powered ankle prostheses report peak push-off torques that fall 20–30% short of the able-bodied trajectory even on dedicated research platforms [61, 62]. The prosthesis-centered reward therefore signals a residual under-assistance deficit that ELSA cannot fully close at the actuator level, and the optimizer responds in the only way available to it: by pushing τ_{lim} to its bound.

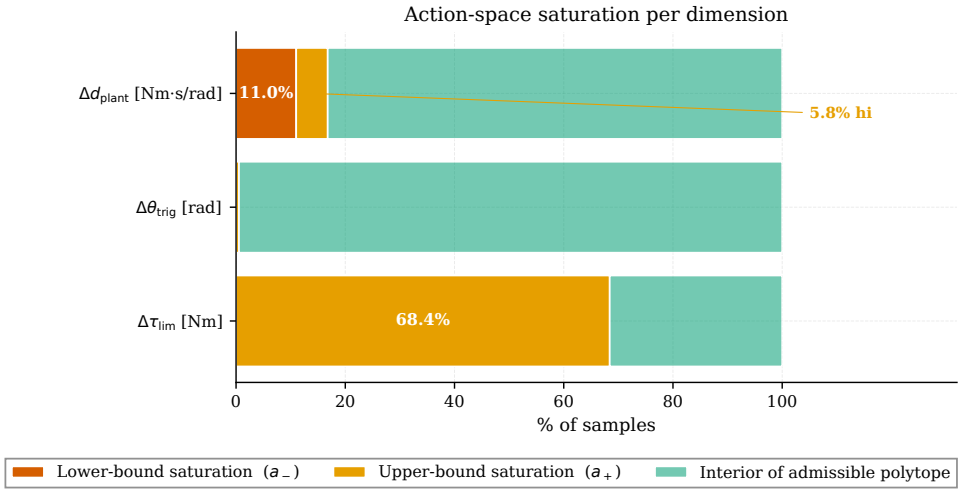


Figure 8.4: Action-space saturation per dimension on the baseline.

The outcome is clinically undesirable on two grounds. Operationally, recommending the actuator limit on almost every walking block drives the prosthesis to the upper margin of its energy and acoustic envelope, a regime flagged as inappropriate for daily use [2, 49]. More fundamentally, when the same actuator is pinned across most states, the policy is no longer exploiting the biomechanical information in the state vector, it is exploiting the geometry of the constraint.

8.1.3 Limitation 2: Sensitivity to gait variability

The second diagnostic concerns how the greedy policy responds when the input gait state is perturbed by an amount comparable to the natural cycle-to-cycle variability of human walking. The sensitivity measure is the ratio $\|\Delta a\|_2/\|\delta\|_2$ between the change in the recommended action and a feature perturbation δ drawn at 2% of the empirical standard deviation of each feature. Its cumulative distribution over the dataset is shown in Figure 8.5.

The median value of 2.85 is unremarkable on its own. The concern lies in the upper tail of the distribution. At the 95th percentile, for 5% of the states, the same 2% perturbation amplifies into a change of nearly 14× its size in the recommended action, with a maximum of 24.5 observed on the dataset. In a non-negligible fraction of states, two consecutive walking blocks that differ only by the natural step-to-step variability of human walking can therefore lead the policy to recommend very different parameter updates, an instability that a prosthetist could not reliably follow in practical use.

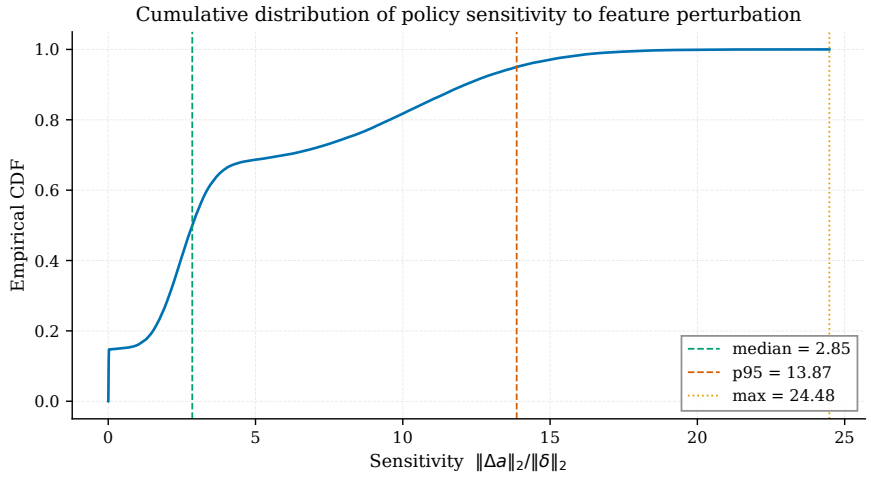


Figure 8.5: Cumulative distribution of the baseline policy sensitivity to a 2% feature perturbation. The median sensitivity is 2.85, but the long upper tail reaches 13.87 at the 95th percentile and 24.48 at the maximum.

8.1.4 Setting the targets for the ablation study

Read together, the three diagnostics above frame the rest of the chapter: a working OPI policy weighed down by two quantified limitations. The 68.4% saturation rate on τ_{lim} and the 95th-percentile sensitivity of 13.87 are the two reference values against which every ablation of the next sections will be read.

8.2 Reshaping the action cost for more Plausible Policies

The two diagnostics that closed Section 8.1.4 translate the baseline issues into quantitative limitations that should be reduced before considering the policy for supervised prosthesis testing. The ablation campaign tested six independent hyperparameter axes, introduced

in Section 7.2, against this objective. Figure 8.6 summarizes the net effect of the most impactful values for each ablation axis on the saturation indicator. Five of the six axes move the policy by at most a few percentage points. But one axis clearly stands out, the rescaling of the action-cost matrix R_a on its τ_{lim} entry by a factor of five, which alone lifts 39 percentage points off the "≥ 1 dim" saturation rate. The present section analyzes that lever in detail.

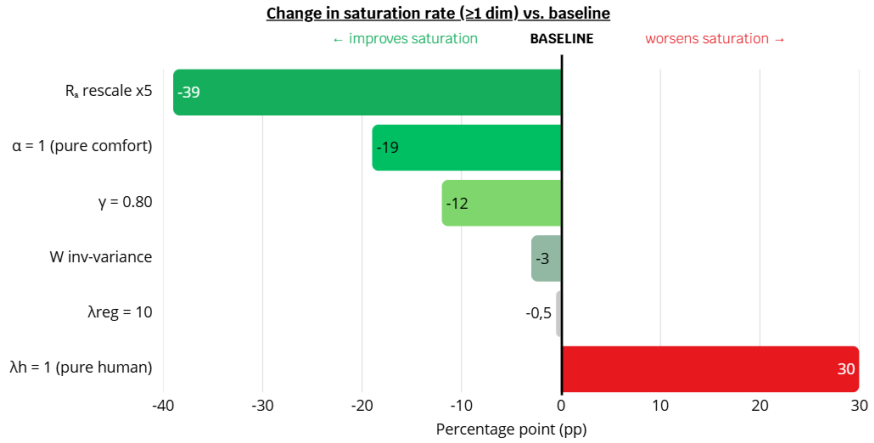


Figure 8.6: Net effect of the most impactful value of each ablation axis on the fraction of gait states with at least one parameter saturated, expressed in percentage points (pp) relative to the baseline. Negative values are improvements; positive values indicate the policy degraded relative to the baseline.

8.2.1 Both limitations relax under a single targeted rescaling

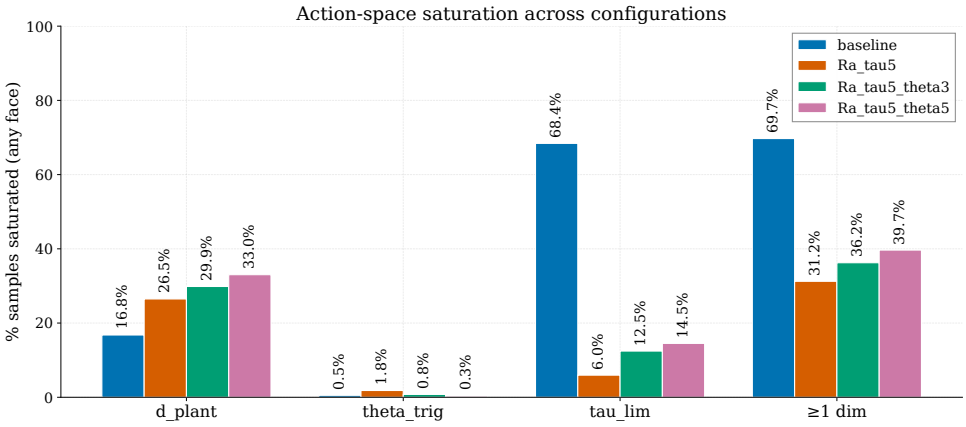


Figure 8.7: Saturation rate per parameter and aggregated across parameters, for the four configurations of the R_a sweep (cf. Section 7.2).

Figure 8.7 reports the saturation rate of each parameter under the baseline and three variants of the rescaling. The dominant change is, as intended, on the torque limit: τ_{lim} saturation collapses from 68.4% to 6.0% in the configuration that rescales only the τ_{lim} entry of R_a by a factor of five. The mechanism is direct. The entry of R_a that the rescaling touches sets how expensive a one-unit change of τ_{lim} is to the policy. In other words, it determines how much cost the policy assigns to changes along that parameter dimension. Multiplying it by five makes the cost bowl in the τ_{lim} direction five times sharper, which pulls the bottom of that bowl back inside the admissible parameter box which means that the policy's recommendation no longer collapses onto the upper bound. This factor-of-eleven reduction settles a question that the baseline analysis had left open. Had the actuator bound been physically too low, no reshaping of the cost would have brought the saturation down: the value would simply have dropped slightly and remained large. The collapse seen here therefore identifies the cause of the limitation as the geometry of the learned cost surface, not as a hardware mismatch.

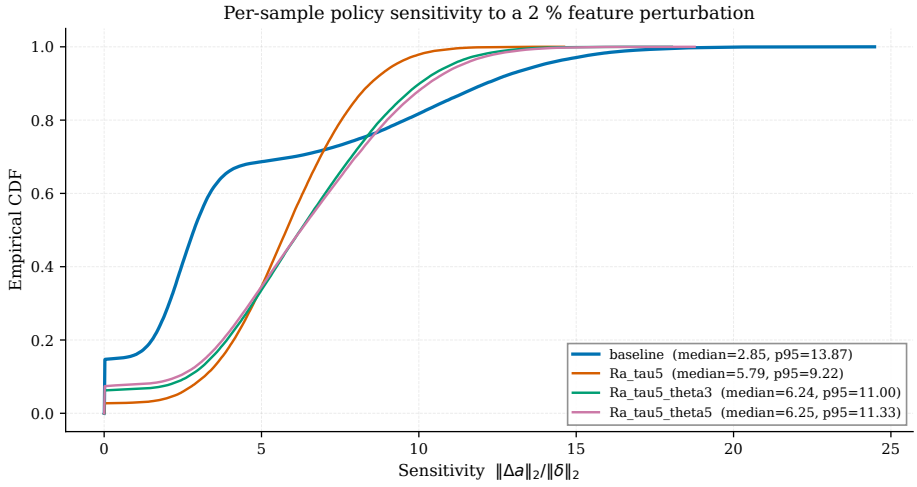


Figure 8.8: Cumulative distribution of the per-sample policy sensitivity to a 2% feature perturbation. The two-regime shape of the baseline (step at zero from saturated samples, then a heavy tail to 24.5) is replaced under `Ra_tau5` by a smoother unimodal distribution with a substantially tighter upper tail.

The sensitivity diagnostic moves in the same direction. Figure 8.8 compares the four configurations on the same 2% perturbation protocol used for the baseline. The baseline curve has the two-regime shape introduced in Section 8.1.3: a vertical step at zero from the saturated samples, followed by a long heavy tail that reaches 24.5. The rescaling removes both features. The `Ra_tau5` curve is smoother and unimodal, and its 95th percentile drops from 13.87 to 9.22, a reduction of 34%. The rise of the median from 2.85 to 5.79 is the structural counterpart of the saturation reduction, not a regression in policy quality. The assumption is that the baseline median was artificially compressed by samples whose policy is pinned to a bound and therefore does not respond to small input perturbations at all, and releasing those samples reveals a small but bounded sensitivity rather than an exactly zero one. From the perspective of future supervised prosthesis testing, the

upper tail matters more than the median: an occasional large unexplained change in the recommended update would be harder to interpret and less acceptable than a slightly larger but consistent variation across steps.

The third reading of this section concerns the structure of S itself. The biomechanically interpretable learned matrix S obtained in the baseline is preserved under this R_a rescaling. Its entries correlate with their baseline counterparts at a Pearson value of 0.91. The algorithm absorbs the new constraint on τ_{lim} by strengthening a small number of state-action couplings and by slightly relaxing the curvature on the untouched dimensions (cf. detailed R_a rescaling S matrix in Appendix C). The policy that emerges from the rescaling is therefore a quantitatively different recommendation, expressed in the same biomechanical language, already interpretable, as the baseline.

8.2.2 The trade-off: a balloon-squeeze on the saturation

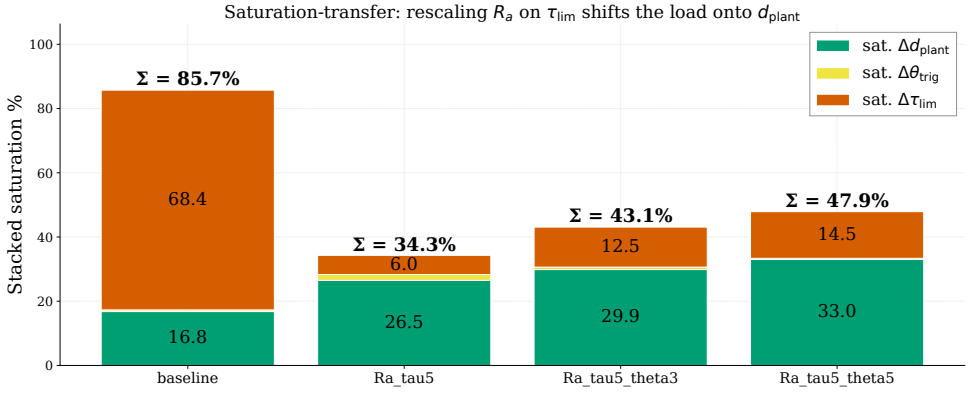


Figure 8.9: Saturation budget per configuration: the total height of each bar is the sum of per-parameter saturation rates, and the segments indicate which parameter contributes how much.

A closer reading of Figure 8.7 reveals that the saturation has not simply disappeared, part of it has moved. The plantarflexion damping, which the baseline pinned to a bound on 16.8% of the states, rises to 26.5% under `Ra_tau5`. The reframing in Figure 8.9 makes the redistribution explicit: the total saturation budget drops from 85.7% to 34.3%, but its largest contributor shifts from the torque limit (orange) to the damping (green). This behavior is analogous to the "balloon-squeeze" effect described in the literature on cognitive load [63]: constraining a finite resource along one axis can redistribute it onto others in a way that is difficult to anticipate. In this case, the resource is not cognitive workload but the SDP regression capacity available on a 52-transition dataset. Because the fit is global, increasing the penalty on one decision variable can free capacity on other dimensions, which the optimizer then uses to explain the data through different action directions. This is visible in the learned action curvature: the diagonal entry of S_{aa} for d_{plant} decreases from 0.61 to 0.51, even though the corresponding coefficient in R_a was unchanged. The cost bowl along this axis therefore becomes slightly flatter, and its minimum falls outside the admissible range for more states. The two control variants in Figure 8.9, which also stiffen the non-saturating

θ_{trig} dimension, confirm the same mechanism. Penalizing a dimension that did not saturate is not neutral: it removes a corrective lever implicitly used through state-action couplings, causing part of the τ_{lim} saturation to return.

8.2.3 The dataset, not the algorithm, sets the ceiling

The balloon-squeeze effect is not caused by the chosen rescaling. It reflects the data limitation already discussed in Sections 7.1.3 and 7.1.4: only 52 independent block-to-block transitions are available for a matrix S with 55 free entries after symmetry, and the all-to-all augmentation does not change this ratio. In this regime, the regression can recover the dominant patterns of the data, as shown by the baseline analysis in Section 8.1.1, but it lacks enough independent information to absorb a new constraint on one parameter without shifting part of the saturation to another. The transfer toward d_{plant} in Figure 8.9 and the fact that the "≥ 1 dim" saturation remains near 30% in Figure 8.7 therefore reflect the same limitation.

This also frames the answer to the integration research question of this thesis. With one targeted change in the action cost, the framework reduces the two main weaknesses of the baseline: the aggregated saturation drops to 31%, and the 95th-percentile sensitivity falls to 9.22. At the same time, the global structure of the learned policy is preserved. In this sense, the policy becomes more plausible as a candidate for future supervised testing, although it remains an offline result.

8.3 Assessing the Contribution of Human Feedback

This section addresses the second research question: does the human feedback term in the reward actually contribute to what the policy learns, or could the biomechanical part alone do the work? The weight λ_h defined in Section 5.5 is precisely the lever needed to answer this, since it interpolates between a purely biomechanical objective ($\lambda_h = 0$) and a purely human-driven one ($\lambda_h = 1$). The remainder of this section therefore examines this ablation axis to characterize the role played by the subjective signal.

8.3.1 The pure-human regime: a structural collapse

Coming back to Figure 8.6, the fully human variant degrades the policy more severely than any other tested configuration, while removing the human signal entirely makes the model perform worse than the baseline on every scale-free diagnostic. Setting $\lambda_h = 1$ removes the biomechanical reference from the reward and trains the policy exclusively on the standardized comfort and assistance scores reported by the participant. The validation pipeline of Chapter 7 reads an unambiguous pattern: 100% of the dataset samples have at least one parameter pinned to a hardware bound, the 95th-percentile sensitivity reaches 67.8 against 13.9 for the baseline (×4.9) with a maximum of 700.7 on individual samples (×28), and the cross-validation overfitting ratio reaches the worst value of the entire ablation campaign at 2.57. The model has not become aggressive, it has become degenerate.

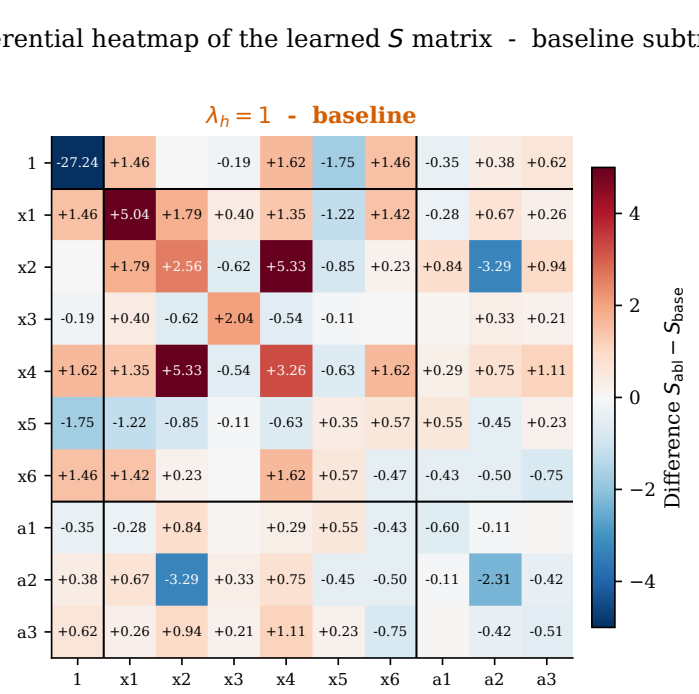


Figure 8.10: Differential heatmap $S_{\lambda_h=1} - S_{\text{baseline}}$. Red indicates an entry that has grown, blue one that has shrunk. The diagonal of the bottom-right 3×3 action-action block is uniformly negative: this is the signature of the curvature collapse described in the text.

The structural cause of this collapse is visible in the learned matrix itself. The block that matters in Figure 8.10 is the 3×3 action-action block in the bottom-right corner, where every diagonal entry has weakened. The curvature controlling how costly it is to move d_{plant} has shrunk by roughly a factor of seventy (from 0.609 to 0.009), while θ_{trig} and τ_{lim} have lost a factor of five (from 2.865 to 0.554) and eight (from 0.589 to 0.076) respectively. The cost-to-go has lost its bowl shape entirely: there is no clear minimum inside the admissible parameter range any more, and the policy falls back to a corner of that box. The 97.9% saturation observed on d_{plant} (cf. Table 8.1) is the visible consequence, with the same corner chosen for almost every gait state. Even the strongest off-diagonal signal of the baseline matrix, the coupling that translated MPA-to-push-off timing into a trigger-angle update, has weakened by -3.29 and now contributes almost nothing.

This outcome is also consistent with the data limitation already discussed in Section 8.2.3. The standardized human reward, by construction, has a much narrower range than the biomechanical one and provides per-sample variance only through the 52 block-level scores actually rated by the participant. With 55 free coefficients to learn and almost no curvature signal in the reward itself, the regression problem becomes severely under-determined. The collapse seen at $\lambda_h = 1$ is therefore not an inherent property of preference-based learning, which has been used successfully when calibrated on large datasets of human comparisons [17], but an indication that, in this experimental regime, the human signal alone does not carry enough independent information to drive the OPI fit.

8.3.2 Hybrid configurations: the human signal contributes to learning

The behaviour observed at $\lambda_h = 1$ provides the upper boundary of the sweep, while the lower boundary at $\lambda_h = 0$ is in fact the more informative endpoint. At $\lambda_h = 0$, the reward is purely biomechanical: the human ratings are discarded entirely and only the deviation from the normative gait drives the learning. Table 8.1 compares the five configurations on the validation diagnostics that admit a meaningful cross-configuration interpretation, that is, those that are not directly proportional to the reward magnitude.

Diagnostic (scale-free)	$\lambda_h = 0$	$\lambda_h = 0.25$	$\lambda_h = 0.5$ (BL)	$\lambda_h = 0.75$	$\lambda_h = 1$
Sat ≥ 1 dim [%]	76.9	54.3	69.7	92.0	100.0
Sat τ_{lim} [%]	73.6	51.2	68.4	92.0	61.2
Sat d_{plant} [%]	12.1	18.0	16.8	15.8	97.9
Sens. p95	27.5	12.8	13.9	7.3	67.8
Sens. max	51.2	20.9	24.5	16.7	700.7
CV overfit ratio	2.05	1.89	**1.83**	2.19	2.57

Table 8.1: Scale-free validation metrics across the λ_h sweep. Cells shaded green improve on the baseline, cells shaded red degrade. Bellman RMSE, mean policy gain, and CV train/val RMSE are omitted because they scale with the absolute reward magnitude and therefore decrease automatically as $\lambda_h \to 1$; they cannot be used to rank configurations on this axis.

Reading the table from the left, the purely biomechanical configuration $\lambda_h = 0$ does not match the baseline on any of the dimensionless diagnostics: its aggregate saturation is 7pp higher, its sensitivity tail almost doubles, and its overfit ratio creeps from 1.83 to 2.05. The validation pipeline now identifies a controller that is less stable and less generalizable than the hybrid baseline. Adding even a quarter of human signal ($\lambda_h = 0.25$) substantially recovers the saturation budget and the sensitivity tail, and the baseline ($\lambda_h = 0.5$) sits very close to the optimum on every line of the table. The overfit ratio is particularly telling: it is dimensionless, computed on data the OPI fit never saw, and it reaches its minimum exactly at the baseline. This is the strongest indication in this experiment that the human ratings provide useful information beyond the biomechanical reward. If those comfort and assistance scores were uninformative noise, asking the model to also explain them could only waste capacity and hurt generalization. The fact that the opposite happens shows that the human ratings capture something about gait quality that the biomechanical features alone do not.

The interpretation is consistent with the broader human-in-the-loop RL literature discussed in Section 2.3.2, where the now-standard practice of combining human approval with a well-defined environmental reward has been repeatedly shown to outperform either component in isolation, especially in low-data regimes [15, 26]. In the present setting, the comfort and assistance scores resolve a degeneracy that the biomechanical features cannot resolve on their own: two gait configurations may produce nearly identical normative-deviation features while feeling substantially different to the user, and the ratings allow the policy to distinguish between them. Conversely, the biomechanical reference provides the action curvature without which the OPI regression cannot retain a usable shape, as

Section 8.3.1 just established. In this dataset, neither term appears sufficient on its own and the most usable policies arise from their combination.

8.3.3 Role of the Biomechanical Anchor in Human Feedback

Read together, the two extremes of the λ_h sweep give a coherent answer to the second research question. The biomechanical term cannot be removed without collapsing the policy onto the corners of the admissible action box, because it is the only source of curvature in the action cost that the OPI regression has access to on this dataset. The human term cannot be removed either: without it, the same regression produces a policy that saturates more, reacts more violently to gait variability, and generalises worse on unseen transitions than the hybrid baseline. The hybrid reward proposed in Section 5.5 is therefore not a convenience. In the experimental regime of this thesis, the framework benefits from the human signal precisely because that signal is read against a normative reference that constrains the geometry of the learned cost surface. In short, the human feedback acts as a complement that informs the policy on subjective dimensions the biomechanical features cannot capture, but it does so only as long as the biomechanical anchor is preserved.

8.4 Retained Configuration After Ablation

The analysis above deliberately focused on the two ablation axes that speak most directly to the research questions of this thesis: the action-cost rescaling (Section 8.2) and the human-feedback weighting (Section 8.3). The four remaining axes proved to be either second-order or empirically inactive on the present dataset, and their detailed numerical analysis is therefore relegated to the appendices to preserve the readability of this chapter. Briefly, changing γ did not materially alter the main diagnostics, varying α had less effect than changing the global human-feedback weight λ_h, inverse-variance feature weighting did not improve the learned policy, and regularization λ_{reg} proved to be inactive on this dataset.

Read across the whole analysis, the configuration retained as the final one of this thesis simply augments the baseline with the single R_a rescaling on τ_{lim} by a factor of five. Every other hyperparameter is left at its baseline value, as summarized in Table 8.2.

Hyperparameter	Decision	Justification
R_a on τ_{lim}	**Adopt** (×5)	Section 8.2
λ_h	Keep at baseline (0.5)	Section 8.3
α	Keep at baseline (0.5)	Appendix D
γ	Keep at baseline (0.9)	Appendix E
W	Keep uniform	Appendix F
λ_{reg}	Keep at baseline	Appendix G

Table 8.2: Final configuration retained at the end of the ablation study. Only one hyperparameter departs from the baseline; the others were either confirmed at their baseline value or empirically inactive on the present dataset.

Overall, these results form the empirical basis on which the next chapter steps back to discuss the contributions of this work, its current limitations, and the perspectives it opens for future iterations of the framework.

9

Discussion, Limitations and Perspectives

This thesis asked two questions: whether a Human-in-the-Loop Reinforcement Learning layer could sit on top of ELSA's existing controller without breaking its safety guarantees, and whether structured subjective feedback brings measurable learning value in the small-data offline regime of a real prosthesis experiment. Both can now be answered, but only within the limits of what the experimental campaign actually delivered.

Contributions. Three contributions anchor this work. First, a focused review of reinforcement learning for impedance tuning of robotic prostheses (Chapter 3) identified the absence of structured subjective feedback as a recurring gap even in the most recent OPI-based formulations [45–47]. Second, it presents, to the best of our knowledge, the first application of OPI to a powered ankle-foot prosthesis: prior work has consistently focused on knee prostheses, and ELSA's constraints required dedicated state, action and reward designs (Chapter 5). Third, and the most central, the first explicit integration of comfort and perceived-assistance scores into the reward of an OPI controller, allowing the same algorithmic backbone to combine biomechanical references with the user's own perception of the device.

What this work has established, and his limitations. On the first research question, the proposed framework coexists with ELSA's mid-level controller without bypassing its safety envelope, and the learned policy becomes more plausible and better behaved after a single, principled reshaping of the action cost (Section 8.2). On the second, the comfort and assistance ratings carry measurable learning value: the hybrid baseline reaches the lowest cross-validation overfit ratio of the entire campaign, while both reward extremes degrade every dimensionless diagnostic (Section 8.3). These results rest on two structural limitations. The first is the size of the experimental campaign relative to the capacity of the model: 52 real transitions for a quadratic surrogate with ~ 55 free coefficients leave the regression under-determined, and the all-to-all augmentation of Section 7.1.4 does not lift that gap. The second is methodological: anchoring the state on healthy normative references [55] embeds

into the reward itself a target no current powered prosthesis can fully reach mechanically, a choice that silently shaped the saturation limitation of Section 8.1.2. Together, these limitations also force a more cautious reading of the hybrid-reward result: the human signal may help as much as it does in part because the under-determined biomechanical regression leaves residual variance that comfort and assistance scores can absorb. Whether this complementarity persists once more independent transitions push the biomechanical fit closer to the data is an empirical question that the present campaign cannot resolve on its own.

Perspectives. The results of this thesis open four main directions for future work:

1. **Expanding the experimental dataset.** The present conclusions remain limited by the small dataset, the use of a single healthy participant, and the absence of repeated sessions. A larger cohort, including transtibial amputees (the actual target population) and multiple visits per subject, would make it possible to distinguish true parameter effects from adaptation, fatigue, and day-to-day variability.

2. **Matching model capacity to realistic prosthesis data.** The quadratic approximation used here was expressive enough to learn meaningful state-action couplings, but its number of coefficients was close to the number of independent transitions. Future versions could therefore explore simpler state representations, stronger biomechanical priors, or more constrained cost-to-go approximations. The goal would not be to make the model more complex, but to make it better matched to the amount of data that prosthesis experiments can realistically provide.

3. **Extending the learning algorithm.** Since the policy was designed as an offline-first iterate before any supervised online use, the offline phase could support heavier methods, such as experience-replay variants of policy iteration [46], constraint-embedded formulations [47], or modern offline RL methods controlling extrapolation outside the dataset. The modular pipeline of Chapter 6 was designed to support such extensions without redefining the full framework.

4. **Rethinking the feedback protocol.** The 0–10 comfort and assistance ratings used here were intentionally simple, but future protocols could collect richer signals, such as pairwise preferences between assistance profiles [17, 21] or structured verbal feedback. In this setting, LLMs could play a useful supporting role: not to control the prosthesis directly, but to help interpret free-text comments such as "the push-off feels too late" or "the heel is too hard" and translate them into structured labels, preference constraints, or uncertainty-aware annotations that enrich the reward while keeping the final learning decision transparent and supervised. This would move the framework toward a richer form of HITL-RL, where the user is not only asked to score a configuration, but can explain how and why it feels right or wrong.

Closing. With these caveats stated, the framework remains a usable first step: it runs end-to-end, learns from real data, integrates the user where prior work only inferred them, and degrades gracefully when its assumptions are stressed. This is a step toward prostheses that are no longer simply tuned by humans, but that adapt to them.

A

Appendix - Analysis of Existing Experimental Data for ELSA

A.1 Context and Motivation

During its development, ELSA underwent four experimental campaigns (DC1-DC4) between 2022 and 2024, primarily in collaboration with Össur laboratories in Iceland, providing a high amount of experimental data across several amputee users. These campaigns aimed at progressively refining ELSA's control strategies and validating its biomechanical performance through experiments on ELSA's different modes of locomotion and different set of parameters.
Thus, before designing a HITL-RL framework for ELSA, a crucial question arises: could the experimental data collected during the previous development campaigns be reused to initialize a meaningful policy in an offline setting?

Given the results and conclusions observed in the literature (cf. Chapter 3), a central challenge is consistently highlight for RL in physical human-robot interaction: **online exploration is costly and potentially unsafe.** In assistive robotics, and especially in prosthetic control, unstructured exploration cannot be tolerated, as each policy update directly affects a human user. For this specific reason, several recent works mentioned in Chapter 3 [43, 45, 47] emphasize the importance of initializing learning from prior data, demonstrations, or structured offline datasets before deploying online adaptation. A well-chosen offline initialization can significantly:

- reduce unsafe exploration,
- accelerate convergence,
- and stabilize early policy updates.

A

In other words, starting from a reasonable initial policy isn't merely a convenience, it's often a prerequisite for safe and efficient learning in embodied systems.

Knowing the cost and complexity of conducting human experiments, and in light of the demonstrated benefits of offline pre-training in RL, it was both natural and methodologically sound to investigate whether these datasets could serve as a foundation for **constraining the policy search space**, **calibrating initial parameter ranges**, or **training a preliminary policy in an offline RL framework**.
This chapter doesn't attempt to reuse the data opportunistically. Rather, it systematically evaluates whether the existing experimental campaigns satisfy the structural requirements necessary to support offline policy initialization for HITL-RL.

A.2 Overview of the Experimental Campaigns

Across DC1 to DC4, several users (transtibial and transfemoral amputees) tested different versions of ELSA under varying conditions.

A.2.1 DC1 - Early Validation (ELSA 3.0)

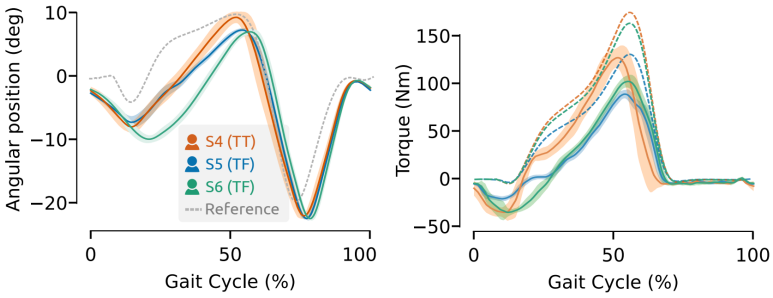


Figure A.1: DC1: Results for level-ground walking (LGW) on a treadmill

The first campaign revealed important mechanical and control limitations:

- Limited motor velocity (115°/s)
- Inconsistent LPS behavior,
- Suboptimal timing despite correct torque-angle shapes,
- Slow dorsiflexion during swing.

Although the general gait profiles exhibited expected biomechanical patterns (cf. Figure A.1), temporal alignment issues and hardware limitations significantly influenced the results. Knowing this, the DC1 data is barely valuable as a historical baseline, and even less so as a reliable training dataset.

A.2.2 DC2 - Parameter Exploration (ELSA 3.0+)

DC2 focused on systematic variations of assistance levels, damping values, and trigger thresholds.

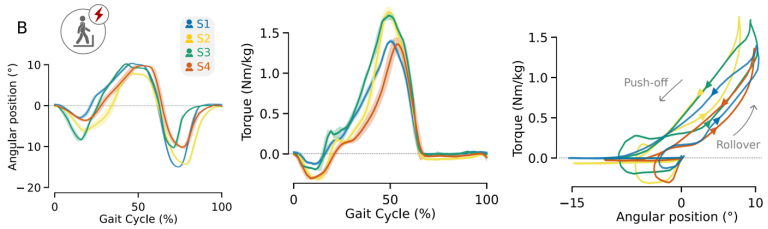


Figure A.2: DC2: Results for LGW on a treadmill

This campaign (cf. Figure A.2) highlights an important insight: optimal parameter values are highly user-dependent. However, from an RL perspective, the data do not form a coherent trajectory dataset, as parameter changes occur discretely between sessions rather than continuously within a Markovian transition structure. Above all, changes made to each parameter are not explicit, requiring assumptions to be made about certain parameter values.

A.2.3 DC3 - Strategy Comparison (ELSA 3.1)

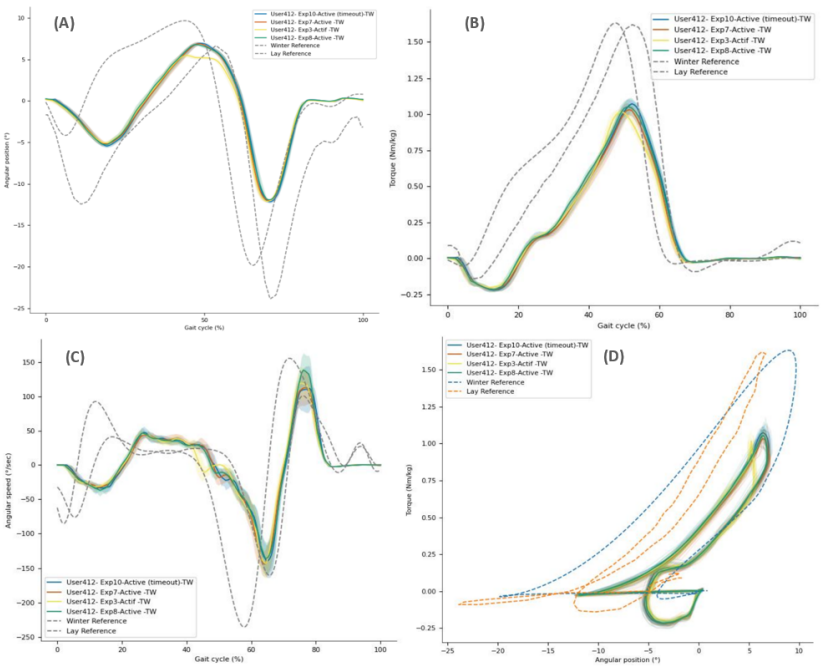


Figure A.3: DC3: Aggregated Results for LGW on a treadmill from the single tested user

A

During this campaign, the goal was to compared 3 assistance strategies regarding how to deliver assistance most effective and comfortable way. The strategy kept was the Time-based one where assistance is provided when the trigger angle is reached and the maximal value is reached after a predefined time period δt.

The problem with this campaign is that it was carried out on a single user (cf. Figure A.3), so there is a clear lack of data. In addition, patient feedback annotations remain sparse and unclear, limiting their direct integration as reward signals.

A.2.4 DC4 - Assistance Level and LPS Comparison

DC4 further evaluated multiple assistance levels (20 Nm, 40 Nm, 60 Nm) with and without the LPS (cf. Figure A.4).

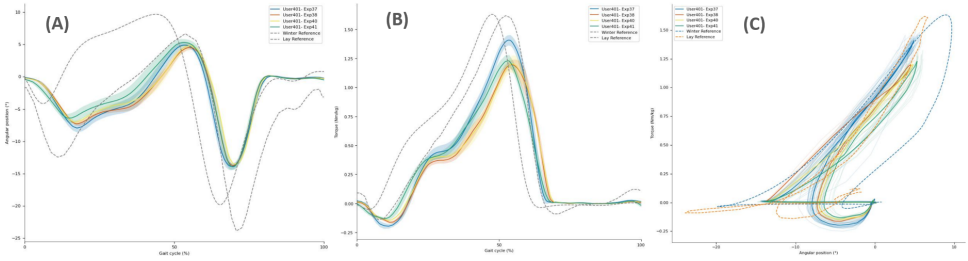


Figure A.4: DC4: Aggregated Results for LGW on a treadmill from the single tested user

Importantly, DC4 with DC3 together provide what can be interpreted as physiological envelopes of acceptable walking patterns rather than optimal trajectories. These envelopes can help define:

- Safe angular ranges,
- Temporal windows for push-off,
- Penalty regions for implausible behavior.

But once again, the main problem lies in the fact that this campaign was also carried out on a single user, resulting in a significant lack of data on other users.

A.3 Unsuitability of the Data for Offline RL

Although the datasets are rich in biomechanical information, they lack several structural properties required for standard offline RL.

A.3.1 Absence of State-Action-Reward Transitions

The datasets do not contain tuples of the form:

$$(s_t, a_t, r_t, s_{t+1})$$

Parameters were manually updated between experiments, not at each timestep. Consequently, no action sequence is recorded in the RL sense, no explicit reward is defined, and

no Markovian transition structure can be reconstructed.
Without consistent state-action transitions, it is impossible to infer a valid dynamics model or apply conventional offline RL algorithms.

A.3.2 Hardware and Firmware Heterogeneity

Across DC1 to DC4:

- Different hardware versions were used (ELSA 3.0 vs 3.1),
- The LPS was inconsistently functional,
- Motor limits changed,
- Control strategies evolved.

This heterogeneity prevents treating the dataset as a stationary environment. An RL agent trained across such conditions would implicitly face a non-stationary MDP with hidden confounders.

A.3.3 Lack of Systematic User Feedback

Most importantly, systematic subjective feedback is largely absent. Only occasional qualitative notes are available (e.g., "*good*", "*more consistent*", "*too much for this speed*"), and these are sparse and not standardized across sessions. Since the core motivation of HITL-RL is precisely to integrate human perception (comfort, stability, propulsion quality) into the learning process, this lack of structured feedback severely limits the dataset's suitability for reward modeling.

B

Appendix - Neighborhood Convergence in Approximate Offline Policy Iteration

Section 8.1.1 reports that $\|\Delta S^{(i)}\|_F$ drops sharply over the first four iterations and then stabilizes in a band of 5-7, never crossing the strict threshold $\varepsilon = 10^{-4}$. This appendix explains why this is the expected outcome and why the returned S_{best} is a principled choice despite the absence of strict convergence.

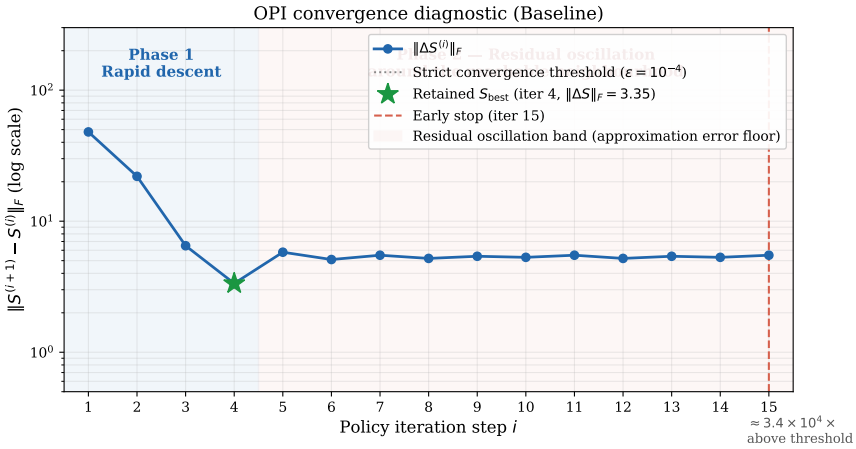


Figure B.1: OPI convergence diagnostic for the baseline. The Frobenius norm contracts rapidly over the first four iterations (Phase 1) then stabilizes within a residual band (Phase 2). The strict convergence threshold is never reached, consistent with the API neighborhood-convergence theorem. S_{best} is retained at iteration 4 ($\|\Delta S\|_F = 3.35$).(Generated with the help of ClaudeAI)

B.1 What the Theory Predicts

The classical Approximate Policy Iteration (API) framework [8] proves that, under a bounded approximation error ε per evaluation step, the sequence $\{Q^{(i)}\}$ produced by API does not converge to the optimal Q^* in the strict sense. It remains in a neighborhood of Q^* whose diameter is bounded by

$$\limsup_{i\to\infty} \|Q^{(i)} - Q^*\|_\infty \;\le\; \frac{2\gamma}{(1-\gamma)^2}\,\varepsilon. \tag{B.1}$$

With $\gamma = 0.9$, this factor equals 100: any residual approximation error is amplified into the bound by two orders of magnitude. The quadratic approximator $\hat{Q}(s,a) = \mu^\top S \mu$ is applied here to a system whose true cost surface is almost certainly not quadratic, so $\varepsilon > 0$ is unavoidable. The theory therefore predicts that $\|\Delta S^{(i)}\|_F$ should never reach zero, which is precisely what Figure B.1 shows.

B.2 Three Regimes and the S_{best} Selection

Three regimes can be distinguished in the Frobenius norm history.

Strict convergence. $\|\Delta S\|_F < \varepsilon_{\text{tol}}$ in finite iterations. This is the textbook fixed-point convergence. On ELSA, it is never observed, in line with the bound above.

Neighborhood convergence. $\|\Delta S^{(i)}\|_F$ remains in a small band around a characteristic value, neither crossing zero nor diverging. This is the regime guaranteed by Equation (B.1) and observed empirically from iteration 5 onward.

Empirical detection and early stopping. Three detectors in `OPI_core.py` identify this regime and terminate training: a *limit-cycle* detector that identifies periodic alternation between near-identical levels (the regime observed in the baseline), a *plateau* detector that fires when the oscillation amplitude collapses near zero, and a *no-progress* detector that fires when the running minimum no longer improves over a sliding window. In the baseline, the limit-cycle detector fires at iteration 15.

Within the detected neighborhood, the iterates $\{S^{(i)}\}$ are not equally good. The one with the smallest $\|\Delta S^{(i)}\|_F$ is the iterate whose policy-improvement step changed the value function the least, making it the closest approximation to a fixed point of the API operator. Returning S_{best} is therefore the minimum-change choice: no other iterate within the neighborhood has a stronger claim to be the converged value function. In the baseline, this selects iteration 4 ($\|\Delta S\|_F = 3.35$), which sits at the boundary between the contraction phase and the residual band, confirming that it is the last iterate that still achieved a genuine reduction.

B.3 Dataset Size and the Radius of the Residual Band

The single term ε in Equation (B.1) implicitly captures all sources of evaluation error, including the limitation of the chosen function class and the quality of the dataset used

B

to fit the Bellman regression. Munos and Szepesvári [11] make the dataset dependence explicit by showing that the effective approximation error in fitted value iteration grows with the mismatch between the data distribution and the distribution induced by successive policies: a poorly covered state-action space inflates ε and therefore widens the residual neighborhood.

In the present setting, this coverage issue is compounded by the under-determination of the Bellman regression itself. As described in Section 7.1.4, the symmetric matrix S has 55 free parameters, while the number of truly independent block-to-block transitions is 15 in the first campaign and 52 in the second. In both cases the system remains under-determined, meaning that multiple matrices S can produce near-identical Bellman residuals on the observed data while differing elsewhere, which sustains the oscillation. As the number of independent transitions grows, the regression becomes better conditioned, ε decreases, and the residual band narrows.

Figure B.2 illustrates this directly by overlaying the Frobenius norm histories of two baseline runs that differ only in dataset size. With 15 real block-to-block transitions (24000 pseudo-transitions), the residual band spans [3.8, 13.5] and the oscillations are erratic, with a spike reaching 13.5 as early as iteration 5. With 52 real transitions (83200 pseudo-transitions), the band narrows to [3.4, 7.3] and the oscillations are more regular. The S_{best} values follow the same trend: 3.82 at iteration 13 for the smaller dataset against 3.35 at iteration 4 for the larger one. Collecting additional real transitions would be expected to continue this trend, until the band approaches the irreducible floor set by the quadratic function class, which cannot be crossed without replacing the approximator with a richer one.

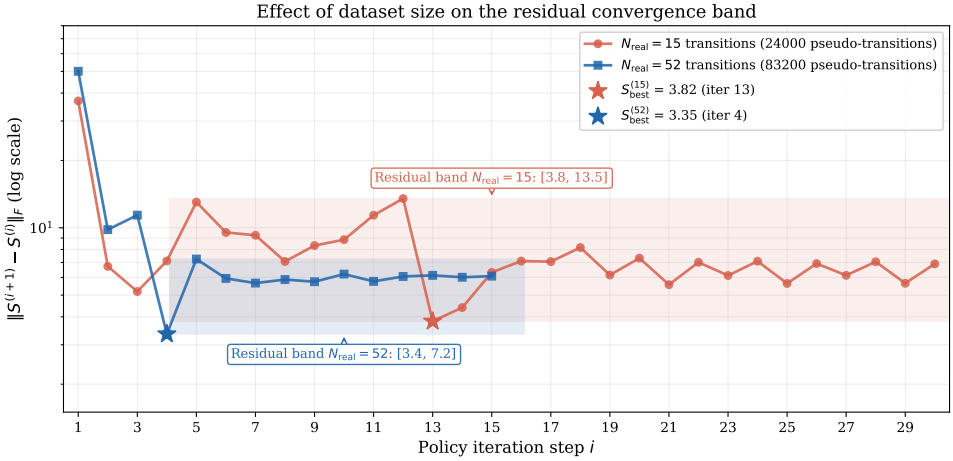


Figure B.2: Frobenius norm histories for two baseline runs differing only in dataset size. The residual oscillation band narrows as the number of independent real transitions increases from 15 to 52, consistent with the dataset-dependent view of ε discussed in [11]. Both curves approach but do not cross the irreducible floor set by the quadratic function class.(Generated with the help of ClaudeAI)

C

Appendix - Learned S-matrix under the R_a Rescaling

Figure C.1 shows the entry-wise difference $S_{\text{abl}} - S_{\text{base}}$ for each configuration of the R_a ablation axis. Red entries have grown relative to the baseline, blue entries have shrunk.

Mechanistic effect on S_{aa} The dominant change is the (a_3, a_3) diagonal entry of S_{aa}, which rises from 0.59 to 2.93 under `Ra_tau5`, a factor of 4.97, in line with the factor-of-five applied to the corresponding entry of R_a. The off-diagonal couplings in S_{xa} adjust accordingly, while the global sign pattern is preserved (Pearson correlation 0.91 on the entries excluding the bias slot).

Global structure preservation Despite these targeted adjustments, the overall sign pattern and block structure of S are preserved. The Pearson correlation between $S_{\text{Ra\_tau5}}$ and S_{base} (computed on the 90 entries excluding the bias slot) equals 0.91. The baseline entry that encoded the strongest interpretable signal, the coupling $S[x_2, a_2]$ linking MPA-to-push-off timing to trigger-angle updates (+3.75 in the baseline), remains large and positive (+3.59 under `Ra_tau5` and +4.10, +4.19 under the two combined rescalings), confirming that the policy continues to use the same biomechanical signal for θ_{trig} selection.

C

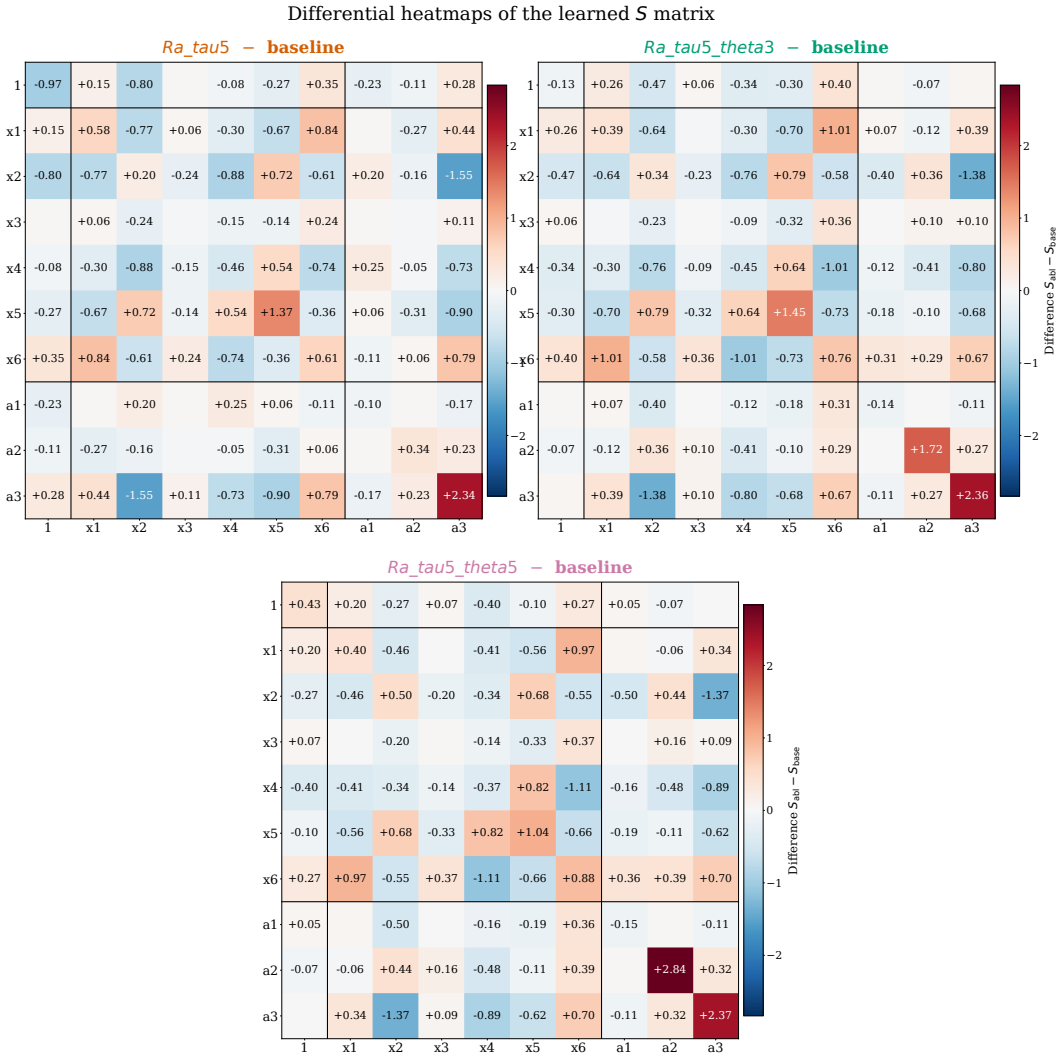


Figure C.1: Differential heatmaps $S_{\text{abl}} - S_{\text{base}}$ for the three R_a configurations. Top left: `Ra_tau5` (R_a scaled ×5 on τ_{lim}). Top right: `Ra_tau5_theta3` (additional ×3 on θ_{trig}). Bottom: `Ra_tau5_theta5` (additional ×5 on θ_{trig}).

D

Appendix - The Comfort-vs-Assistance Balance (α axis)

Within the hybrid reward described in Section 5.5, the coefficient $\alpha \in [0,1]$ sets how much of the human contribution comes from comfort and how much from perceived assistance in Eq 5.8. Varying α at fixed $\lambda_h = 0.5$ shifts the qualitative nature of the subjective signal without changing its overall weight in the reward. The four tested values $\{0, 0.25, 0.75, 1\}$ interpolate between a purely assistance-driven reward and a purely comfort-driven one.

Table D.1 gathers the scale-free diagnostics across the sweep.

Diagnostic	$\alpha = 0$	$\alpha = 0.25$	$\alpha = 0.5$ (BL)	$\alpha = 0.75$	$\alpha = 1$
Bellman RMSE	1.19	1.13	**1.18**	1.57	1.61
Rel. RMSE [%]	18.4	17.5	**18.5**	24.7	25.5
Sat ≥ 1 dim [%]	86.1	76.6	**69.7**	73.6	50.5
Sat τ_{lim} [%]	85.9	75.1	**68.4**	60.8	34.4
Sat d_{plant} [%]	14.5	14.3	**16.8**	34.8	33.2
Sens. p95	11.76	14.82	**13.87**	18.67	15.36
CV overfit ratio	2.14	1.85	**1.83**	1.58	1.51
$S_{aa}[d_{\text{plant}}]$	0.734	0.727	**0.609**	0.527	0.454
$S_{aa}[\theta_{\text{trig}}]$	2.672	2.690	**2.865**	2.515	2.760
$S_{aa}[\tau_{\text{lim}}]$	0.488	0.465	**0.589**	0.713	0.938

Table D.1: Validation metrics across the α sweep at fixed $\lambda_h = 0.5$. Cells shaded green improve on the baseline, cells shaded red degrade. Unlike the λ_h axis, the Bellman RMSE is a valid comparison metric here because the overall reward magnitude changes little (λ_h is held constant), so cost scales are comparable across configurations.

D.1 A second-order axis without collapse

Three observations can be drawn from the table.

1. **No collapsing**: none of the configurations produces the curvature collapse documented at $\lambda_h = 1$: all S_{aa} diagonal entries remain in the same order of magnitude as the baseline across the entire sweep, and the policy never becomes degenerate.

2. **No significant variations**: the amplitude of the variations is substantially smaller than along the λ_h axis: the Bellman RMSE spans about 35% from its minimum to its maximum, the saturation budget shifts by some thirty percentage points, and the sensitivity tail stays within a factor of two of the baseline.

3. **No clear domination**: no configuration uniformly dominates the baseline: $\alpha = 0$ worsens saturation while $\alpha = 1$ worsens the Bellman fit, and neither endpoint consistently outperforms the baseline across all metrics.

D.2 Generalization improves monotonically toward pure comfort

The cross-validation overfitting ratio decreases monotonically from 2.14 ($\alpha = 0$) to 1.51 ($\alpha = 1$), which is the lowest value recorded across the entire ablation campaign. This finding is counterintuitive at first sight: introducing more comfort signal and less assistance signal should, if anything, add noise to the learning target. The likely explanation is that, in this dataset, the comfort scores are more consistent across the gait cycles within a block than the assistance scores. A smoother target function is easier for the quadratic surrogate to generalize, so the regression trained on comfort-weighted rewards extrapolates less aggressively to unseen folds. Whether this advantage persists with larger or more diverse datasets is an open question.

D.3 Data Limitation

These results should be read against the data limitations of the experiment. With a single participant and 52 independent transitions, the comfort and assistance scores each provide 52 scalar values to the regression. Rebalancing between them through α does not inject new information into the fit... It only re-weights two signals that are themselves too sparse to reveal a reliable preference structure. The observed metric variations therefore reflect noise and the same balloon-squeeze mechanism documented in Section 8.2.2, rather than a genuine sensitivity of the learned policy to the comfort-versus-assistance trade-off. The α axis is consequently a second-order modulation of the policy, and the baseline value of 0.5 is retained as a balanced, defensible operating point pending a larger dataset.

E

Appendix - The Discount Factor Axis (γ)

The discount factor γ controls how far into the future the Bellman operator integrates costs when fitting the quadratic cost-to-go $\hat{Q}(s,a) = \mu^\top S \mu$, with $\mu = (1, s^\top, a^\top)^\top$. Four values were tested: $\gamma \in \{0.80, 0.85, 0.90, 0.95\}$, with $\gamma = 0.90$ as the baseline (cf. Section 7.2).

This axis is the natural place to study the *bias-absorption phenomenon*: the observation that the dominant change in S across all global-magnitude perturbations concentrates almost entirely in the single entry $S[1,1]$. The reason is structural, not coincidental, and understanding it is important both for interpreting the learned model and for reassuring that this concentration has no effect on the policy recommendations produced by the model.

Metric	**baseline** ($\gamma = 0.90$)	$\gamma = 0.80$	$\gamma = 0.85$	$\gamma = 0.95$
Bellman RMSE	**1.18**	1.67	1.32	1.52
Relative RMSE [%]	**18.5**	26.1	20.7	23.8
Sat. ≥ 1 dim [%]	69.7	57.7	62.8	76.1
Sat. $\Delta\tau_{\text{lim}}$ [%]	68.4	53.2	60.5	62.9
Sat. Δd_{plant} [%]	16.8	18.8	17.5	36.6
Sens. p95	13.87	11.04	13.21	21.66
CV overfit ratio	**1.83**	2.08	2.04	1.84
$\|S\|_F$	35.31	23.65	26.36	81.39
$S[1,1]$ (bias)	+32.85	+19.55	+22.93	+80.65

Table E.1: Validation metrics across the γ sweep. The last row shows the dominant structural change: the bias entry $S[1,1]$ absorbs almost all of the dilation of $\|S\|_F$ with γ.

Summary metrics. Table E.1 gathers the main diagnostics across the four configurations. The Bellman RMSE is non-monotone, with its minimum at the baseline (1.18), and the CV overfit ratio reaches its best value at the same point (1.83). No tested configuration strictly dominates the baseline on any combined fit-and-generalization criterion. The γ axis is

therefore primarily a confirmation axis: it validates the originally chosen value rather than revealing a better alternative.

E.1 The bias-absorption phenomenon

What happens and why. Varying γ from 0.80 to 0.95 causes $\|S\|_F$ to change by a factor of nearly 3.5 (from 23.65 to 81.39). A natural question is where in S that change goes. Figure E.1 answers it directly: in all three ablation panels, the dominant perturbation is the single entry $S[1,1]$, the top-left coefficient of the full 10×10 matrix. At $\gamma = 0.95$, this one entry accounts for 99.1% of $\|\Delta S\|_F$.

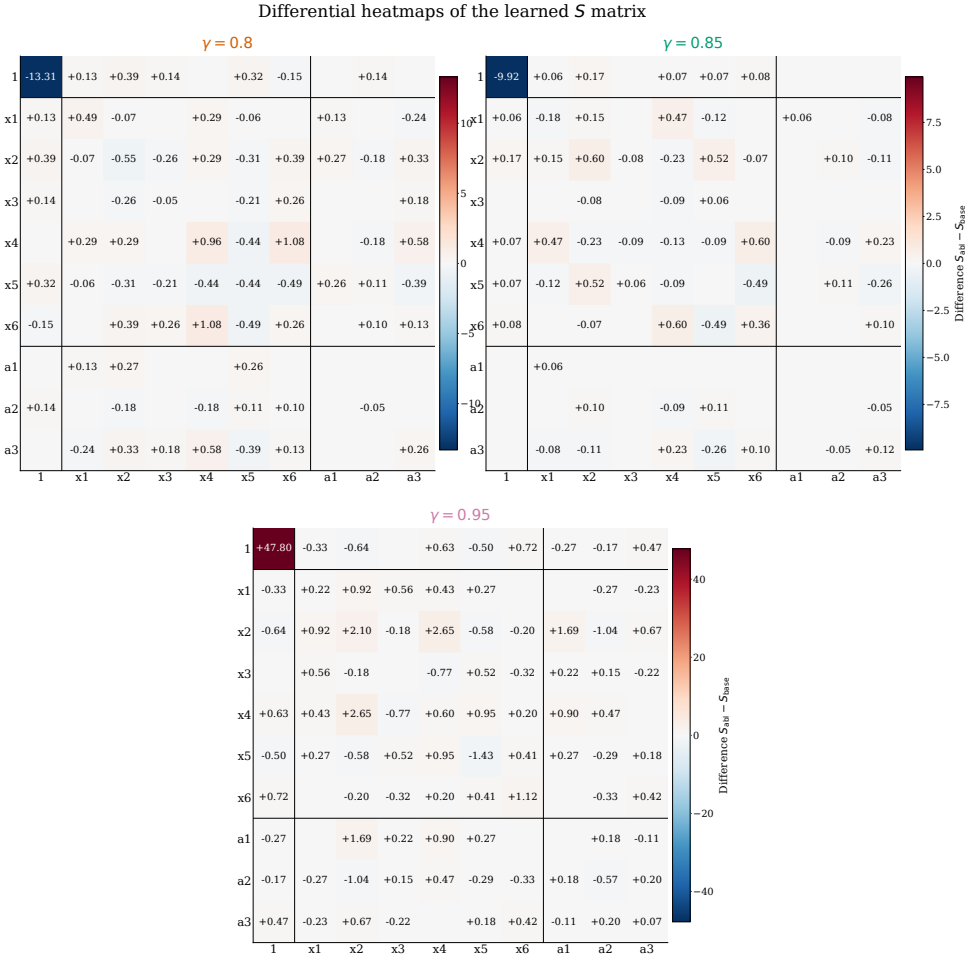


Figure E.1: Entry-wise difference $S_{\text{abl}} - S_{\text{baseline}}$ for the three ablated discount factors. The color scale is dominated by $S[1,1]$ in every panel.

The mechanism is rooted in a classical identity from discounted reinforcement learn-

ing [3, 8]. In infinite-horizon discounted problems, the expected value of the cost-to-go $\hat{Q}$ under any fixed policy π satisfies the fixed-point relation:

$$\mathbb{E}\big[\hat{Q}(s,\pi(s))\big] \;=\; \frac{\mathbb{E}[U]}{1-\gamma}, \tag{E.1}$$

where U is the immediate (one-step) cost. This identity follows directly from taking the expectation of the Bellman equation $\hat{Q}(s,a) = U(s,a) + \gamma\,\mathbb{E}[\hat{Q}(s',\pi(s'))]$ on both sides and solving for the mean. Concretely, on the baseline configuration the mean immediate cost is $\mathbb{E}[U] \approx 3.5$, and with $\gamma = 0.90$ the right-hand side gives $\mathbb{E}[\hat{Q}] \approx 35$. The fitted model must therefore predict an average $\hat{Q}$ of roughly 35 on the dataset.

Now expand $\hat{Q}$ using the quadratic parametrization from Eq. 5.12:

$$\hat{Q}(s,a) \;=\; S[1,1] + 2\,b_s^\top s + 2\,b_a^\top a + s^\top S_{ss}\,s + 2\,s^\top S_{sa}\,a + a^\top S_{aa}\,a. \tag{E.2}$$

Taking the empirical mean over the dataset: the state s is standardized ($\mathbb{E}[s_i] = 0$) and the action a is expressed as an increment ($\mathbb{E}[a_d] \approx 0$), so the linear terms vanish in expectation. The quadratic terms contribute a positive but bounded amount, on the order of $\sum_i S_{ss}[i,i]\,\mathbb{E}[s_i^2] \approx 3$. The remainder must be carried by $S[1,1]$ alone:

$$S[1,1] \;\approx\; \frac{\mathbb{E}[U]}{1-\gamma} \;-\; \mathbb{E}[\text{quadratic terms}] \;\approx\; 35 - 3 \;=\; 32, \tag{E.3}$$

which matches the observed baseline value of 32.85 in Figure 8.3. When γ changes, the factor $1/(1-\gamma)$ changes in direct proportion, and so does $S[1,1]$: it grows to 80.65 at $\gamma = 0.95$ (factor 2.45, close to the predicted factor 2 from doubling the horizon) and shrinks to 19.55 at $\gamma = 0.80$. Every other entry of S is affected far less because the quadratic curvature terms in (E.2) depend mainly on the reward gradients, not on the global horizon length.

E.2 Why this does not affect the policy

The finding above might raise concern: if $S[1,1]$ changes so dramatically with γ, could the recommended prosthesis parameter updates change just as dramatically? The answer is no, and the reason is clear.

The greedy policy is defined as $\pi(s) = \arg\min_{a\in\mathcal{A}(p)} \hat{Q}(s,a)$. Setting the unconstrained gradient to zero:

$$\nabla_a \hat{Q}(s,a) \;=\; 2\,S_{aa}\,a + 2\,S_{sa}^\top s + 2\,b_a \;=\; 0 \quad\Longrightarrow\quad a^\star(s) = -S_{aa}^{-1}\big(S_{sa}^\top s + b_a\big). \tag{E.4}$$

The entry $S[1,1]$ does not appear anywhere in (E.4). Geometrically, $S[1,1]$ is a *vertical shift* of the entire $\hat{Q}$ surface in the (s,a) space: it raises or lowers every value uniformly without changing where the minimum is located. The arg min of a function is invariant to additive constants. Consequently, any hyperparameter change that acts exclusively through $S[1,1]$ is a null intervention from the policy's perspective, regardless of how large the numerical change in $S[1,1]$ appears to be.

This is the structural reason why $\gamma = 0.85$, despite moving $|S[1,1]|$ by -30% relative to the baseline, produces only minor changes in the saturation and sensitivity profiles (Table E.1). The change funnels through the bias entry and bypasses the policy-shaping blocks S_{aa} and S_{sa} almost entirely.

F

Appendix - The Feature-Weight Axis (W)

The biomechanical term of the prosthesis-centered reward penalizes the weighted sum of normalized feature deviations from Eq. 5.7.

Under the uniform baseline ($w_i = 1, \forall i$), each weight is equal, but the actual contribution of each feature to the cost depends on its empirical variance: a feature that deviates more from its reference will naturally carry more weight in the optimization objective.

Figure F.1 reports the measured per-feature contribution $\langle w_i \tilde{x}_i^2 \rangle / \langle \sum_j w_j \tilde{x}_j^2 \rangle$ under the baseline. The distribution is far from uniform: f_6 (push-off mechanical energy) alone accounts for 26.5 % of the feature term, while f_1 (plantarflexion drop) contributes only 9.0 %, a ratio of roughly 3 : 1 between the most and least influential features.

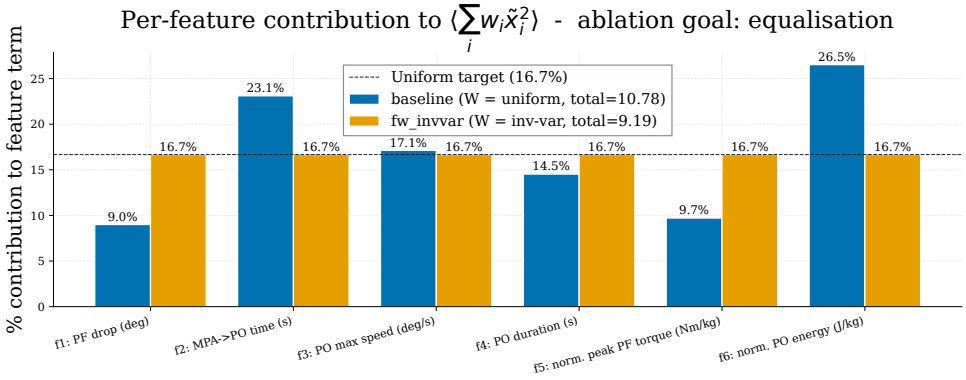


Figure F.1: Per-feature contribution to the biomechanical cost term, expressed as a percentage of the total feature sum. The dashed line at 16.7 % is the uniform target. Under the baseline ($W = I$), contributions reflect the empirical variance of each feature; under `fw_invvar`, they are equalized by construction.

This observation raised a natural question: would forcing every feature to contribute equally to the reward, by effectively removing the influence of empirical variance from

the cost landscape, improve the learned policy? The ablation `fw_invvar` answers it by setting $w_i = n_x / \left(\sum_j 1/\langle \tilde{x}_j^2 \rangle \cdot \langle \tilde{x}_i^2 \rangle \right)$, normalized so that $\sum_i w_i = 6$. By construction, every feature then contributes exactly 16.7 % to the cost term, as the right panel of Figure F.1 confirms.

F.1 A trade-off without a clear winner

Table F.1 compares the two configurations on the scale-free diagnostics used throughout the ablation campaign.

Metric	**baseline** ($W = I$)	**fw_invvar**
Bellman RMSE	1.180	1.155
Relative RMSE [%]	18.5	20.5
Sat. $\Delta\tau_{\text{lim}}$ [%]	68.4	64.8
Sat. Δd_{plant} [%]	16.8	12.8
Sat. ≥ 1 dim [%]	69.7	66.9
Sens. median	2.85	3.52
Sens. p95	13.87	15.39
CV overfit ratio	1.826	1.890

Table F.1: Validation metrics for the feature-weight ablation.

The picture is mixed. On the Bellman fit and on the saturation footprint, `fw_invvar` shows marginal gains: the absolute RMSE improves by 2 %, and the aggregate saturation rate drops by 2.8 percentage points. These improvements are small in absolute terms and arise from a single mechanism: equalising the feature costs reduces the total feature-term magnitude by roughly 15 %, which softens the action-penalty balance and slightly relaxes the saturation pressure on all dimensions uniformly. There is no redistribution between actuators, unlike the R_a rescaling of Section 8.2.1, where the saturation transferred from τ_{lim} toward d_{plant}.

The sensitivity diagnostic moves in the opposite direction. Figure F.2 shows that the policy sensitivity CDF shifts uniformly to the right under `fw_invvar`: the median increases by 24 % (from 2.85 to 3.52) and the 95th percentile by 11 % (from 13.87 to 15.39). Every part of the distribution worsens.

The combination of a marginal saturation gain and a clear sensitivity loss, together with a slightly degraded cross-validation overfit ratio, leads to the conclusion stated in Table 8.2 of Section 8.4: the uniform feature weighting is retained. The disparity in per-feature contributions observed in the baseline is informative rather than problematic. Indeed, it reflects how this participant's gait departs from normative walking, and the policy is right to weight those departures accordingly.

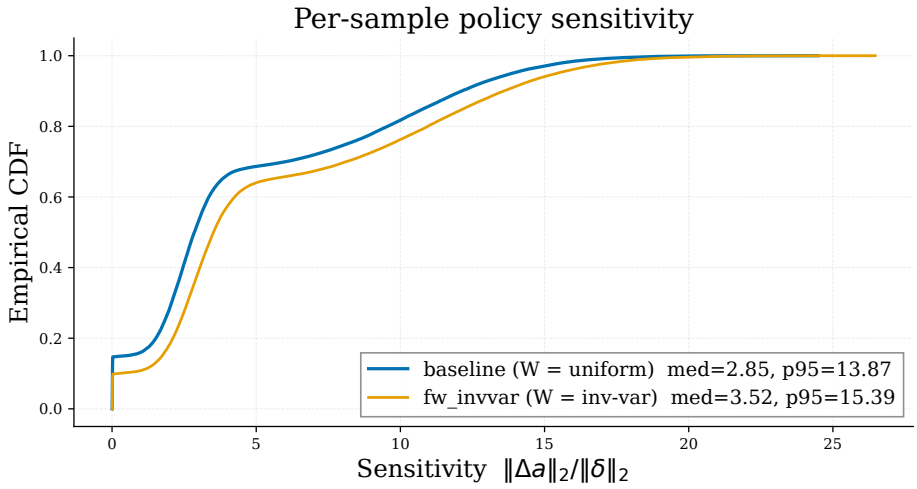


Figure F.2: Cumulative distribution of policy sensitivity to a 2% feature perturbation. The `fw_invvar` curve (orange) sits uniformly to the right of the baseline (blue), indicating a consistent increase in sensitivity across all percentiles.

G

Appendix - The Tikhonov Regularisation Axis (λ_{reg})

The policy-evaluation step minimises a penalised least-squares objective over the symmetric matrix S (cf. Section 6.4, Eq. 6.2):

$$\min_{S_{aa} \succeq \varepsilon I} \underbrace{\sum_{n=1}^{N} e_n(S)^2}_{\text{data fit}} + \underbrace{\lambda_{\text{reg}} \|S\|_F^2}_{\text{regulariser}}, \tag{G.1}$$

where $e_n(S) = \mu_n^\top S \mu_n - \gamma \mu_n'^\top S \mu_n' - U_n$ is the Bellman residual of pseudo-transition n, U_n is the observed immediate cost, and the sum runs over all N pseudo-transitions in the augmented dataset. The regulariser $\lambda_{\text{reg}}\|S\|_F^2$ penalises the overall magnitude of S by summing the squares of all its entries. Eight log-spaced values were tested: $\lambda_{\text{reg}} \in \{0, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 10\}$ (cf. Section 7.2). The result is that the regulariser is mathematically inactive for $\lambda_{\text{reg}} \leq 10^{-2}$, and this section explains why.

G.1 A scale calibration issue

The regulariser can only influence the SDP solution if its contribution to the objective is non-negligible relative to the data-fit term. The two terms do not scale in the same way with dataset size, and this asymmetry is the root cause of the inactivity observed here.

The data-fit term sums squared Bellman residuals over every pseudo-transition in the dataset. As described in Section 7.1.4, the all-to-all augmentation produces $N = 83\,200$ such pseudo-transitions from 52 real block-to-block transitions. With a baseline Bellman RMSE of 1.18 (cf. Section 8.1.1), the total data-fit magnitude is:

$$N \cdot \text{RMSE}^2 \approx 83\,200 \times 1.18^2 \approx 1.15 \times 10^5. \tag{G.2}$$

The regulariser, by contrast, depends only on the learned matrix S and not on N:

$$\lambda_{\text{reg}} \cdot \|S\|_F^2 \approx 1\,247 \cdot \lambda_{\text{reg}}, \tag{G.3}$$

where $\|S\|_F^2 \approx 35.31^2 \approx 1\,247$ is the squared Frobenius norm of the baseline solution (cf. Figure 8.3). Because N appears in the data-fit term but not in the regulariser, the activity threshold of λ_{reg} scales linearly with N: the larger the augmented dataset, the stronger λ_{reg} must be to have any effect. Parity between the two terms is reached at $\lambda_{\text{reg}} \approx 92$; a detectable contribution (1 % of the data fit) requires $\lambda_{\text{reg}} \gtrsim 0.9$. At the standard literature value $\lambda_{\text{reg}} = 10^{-3}$, the regulariser accounts for roughly 10^{-5} of the total objective.

Figure G.1 illustrates this concretely at five representative values of λ_{reg}.

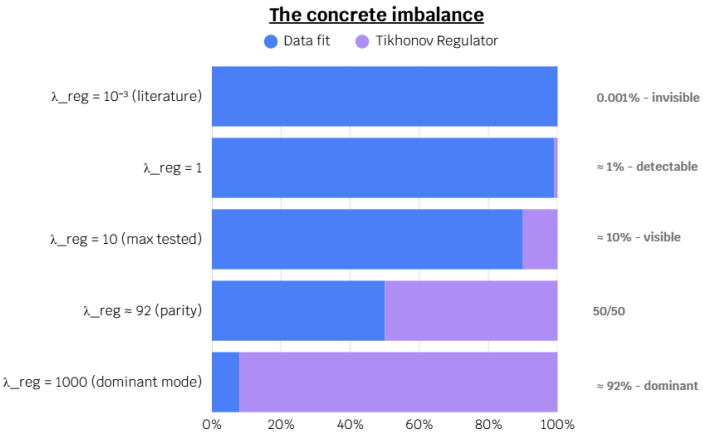


Figure G.1: Relative contribution of the data-fit term (blue) and the Tikhonov regulariser (purple) to the total SDP objective, at five representative values of λ_{reg}.

G

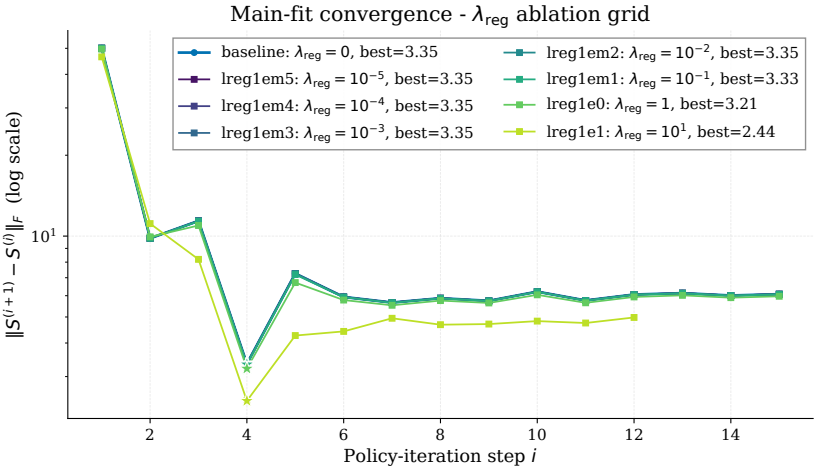


Figure G.2: Main-fit convergence trajectories $\|S^{(i+1)} - S^{(i)}\|_F$ on a log scale, overlaid for the eight λ_{reg} configurations. Stars mark the retained S-best iterate per configuration. The five inactive configurations produce indistinguishable trajectories; $\lambda_{\text{reg}} = 10$ converges to a tighter neighborhood ($\|\Delta S\|_F = 2.44$, a reduction of 27 %) consistent with the stronger convexity conferred by Tikhonov regularisation.

G.2 Empirical consequences across the sweep

The convergence trajectories in Figure G.2 make the inactivity concrete. The first 5 configurations, from $\lambda_{\text{reg}} = 0$ to $\lambda_{\text{reg}} = 10^{-2}$, produce trajectories that are identical to four decimal places: the same S-best iterate (iteration 4), the same Frobenius norm at S-best ($\|\Delta S\|_F = 3.347$), and the same early-stopping iteration (15, limit-cycle detector). Only $\lambda_{\text{reg}} = 1$ and $\lambda_{\text{reg}} = 10$ produce visibly distinct behavior.

Table G.1 gathers the key diagnostics for all eight configurations. A few observations are worth retaining.

Metric	$\lambda = 0$ **(BL)**	10^{-5}	10^{-4}	10^{-3}	10^{-2}	10^{-1}	10^{0}	10^{1}
Bellman fit								
RMSE	**1.180**	1.180	1.180	1.180	1.180	1.179	1.174	1.163
Saturation								
Sat. ≥ 1 dim [%]	**69.71**	69.71	69.71	69.71	69.71	69.71	69.42	69.23
Sensitivity (2 % perturbation)								
Median	**2.85**	2.85	2.85	2.85	2.85	2.85	2.84	2.89
p_{95}	**13.87**	13.87	13.87	13.87	13.87	13.87	13.86	13.70
Cross-validation								
CV train RMSE	**1.610**	1.610	1.610	1.610	1.606	1.598	1.579	1.305
CV val RMSE	**2.669**	2.669	2.669	2.669	2.667	2.672	2.890	2.556
CV overfit ratio	**1.83**	1.83	1.83	1.83	1.83	1.84	1.95	2.01
S structure (informational)								
$\|S\|_F$	**35.31**	35.31	35.31	35.31	35.30	35.27	35.00	32.91
$S[1,1]$ (bias)	**32.85**	32.85	32.85	32.85	32.85	32.82	32.54	30.40

Table G.1: Validation metrics across the λ_{reg} sweep. Grey cells mark configurations numerically identical to the baseline.

When the regulariser does become active, at $\lambda_{\text{reg}} \in \{1, 10\}$, its effect is modest and structurally uninformative for the two pathologies identified in Sections 8.1.2 and 8.1.3. The aggregate saturation rate moves from 69.71 % at baseline to 69.23 % at $\lambda_{\text{reg}} = 10$, a reduction of less than half a percentage point. The sensitivity tail (p_{95}) shifts from 13.87 to 13.70. Both changes are negligible relative to the −39 percentage point gain achieved by the R_a rescaling of Section 8.2.1.

The reason Tikhonov fails to address saturation is structural. The unconstrained policy magnitude scales roughly as $\|S_{xa}\|/\|S_{aa}\|$. Tikhonov shrinks S_{xa} and S_{aa} coherently and in proportion to their absolute values, so the ratio, which is what governs where the policy minimum falls relative to the admissible box, is nearly preserved. By contrast, the R_a rescaling acts surgically on the τ_{lim} diagonal of S_{aa} alone, directly moving the unconstrained minimum back inside the box for that dimension. Global shrinkage of S cannot replicate a targeted change in action-space curvature.

The shrinkage that does occur is also poorly distributed from a policy standpoint. At every active λ_{reg}, the bias entry $S[1,1]$ absorbs roughly 85 % of the total Frobenius norm reduction (86 % at $\lambda_{\text{reg}} = 1$, 84 % at $\lambda_{\text{reg}} = 10$). This is the expected Tikhonov pattern: the largest entry by absolute value bears the largest shrinkage pressure, and $S[1,1] = 32.85$ is an order of magnitude larger than most other entries. As shown in Appendix E, $S[1,1]$ is a vertical shift of the cost surface that does not enter the greedy policy formula. The overwhelming majority of the regularization budget is therefore spent on an entry that has no effect on the recommended parameter updates.

The S_{xa} coupling block, which encodes how gait features translate into parameter recommendations, is preserved across the entire sweep. At $\lambda_{\text{reg}} = 10$ the Pearson correlation between S_{xa} and its baseline counterpart is 0.999, with no sign change on any of the 18 entries. In this sense, the λ_{reg} axis is the most conservative of the four ablation axes studied in this work: it leaves the qualitative policy structure intact under any tested value.

G.3 Persistence of the cross-validation gap

Tikhonov regularization is classically expected to reduce overfitting. Table G.1 shows the opposite, with the CV overfit ratio growing monotonically from 1.83 at baseline to 2.01 at $\lambda_{\text{reg}} = 10$.

The reason is direct: as established in Section G.1, the all-to-all augmentation inflates the data-fit term by a factor of $\sim 1\,600$ relative to the 52 truly independent transitions, pushing the regulariser's activity threshold far beyond any tested value. The tool designed to combat overfitting is deactivated by the very augmentation introduced to mitigate data scarcity. The validation gap that remains reflects block-level distributional shift between training and held-out folds, which global shrinkage of S cannot resolve regardless of λ_{reg}.

Bibliography

References

[1] J. Perry and J. M. Burnfield, *Gait Analysis: Normal and Pathological Function*, 2nd ed. SLACK Incorporated, 2010.

[2] J. Evrard, "Control, characterization and validation of the efficient lockable spring ankle (elsa) prosthesis," Ph.D. dissertation, UCLouvain, Louvain-la-Neuve, Belgium, 2025.

[3] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. Cambridge, MA: MIT Press, 2018.

[4] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: A survey," *Journal of Artificial Intelligence Research*, vol. 4, pp. 237–285, 1996. [Online]. Available: https://arxiv.org/abs/cs/9605103

[5] E. L. Thorndike, *Animal Intelligence: Experimental Studies*. New York, NY: The Macmillan Company, 1911.

[6] R. Bellman, *Dynamic Programming*. Princeton, NJ: Princeton University Press, 1957.

[7] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. New York: Wiley-Interscience, 1994.

[8] D. P. Bertsekas and J. N. Tsitsiklis, *Neuro-Dynamic Programming*. Belmont, MA: Athena Scientific, 1996.

[9] R. A. Howard, *Dynamic Programming and Markov Processes*. Cambridge, MA: MIT Press, 1960.

[10] L. C. Baird, "Residual algorithms: Reinforcement learning with function approximation," in *Proceedings of the Twelfth International Conference on Machine Learning (ICML)*, 1995, pp. 30–37. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B978155860377650013X

[11] R. Munos and C. Szepesvári, "Finite-time bounds for fitted value iteration," *The Journal of Machine Learning Research*, vol. 9, pp. 815–857, 06 2008. [Online]. Available: https://api.semanticscholar.org/CorpusID:12973375

[12] M. G. Lagoudakis and R. Parr, "Least-squares policy iteration," *Journal of Machine Learning Research*, vol. 4, p. 1107–1149, Dec. 2003.

[13] S. Levine, A. Kumar, G. Tucker, and J. Fu, "Offline reinforcement learning: Tutorial, review, and perspectives on open problems," 2020. [Online]. Available: https://arxiv.org/abs/2005.01643

[14] S. Fujimoto, D. Meger, and D. Precup, "Off-policy deep reinforcement learning without exploration," 2019. [Online]. Available: https://arxiv.org/abs/1812.02900

[15] C. O. Retzlaff, S. Das, C. Wayllace, P. Mousavi, M. Afshari, T. Yang, A. Saranti, A. Angerschmid, M. E. Taylor, and A. Holzinger, "Human-in-the-loop reinforcement learning: A survey and position on requirements, challenges, and opportunities," *Journal of Artificial Intelligence Research*, vol. 79, pp. 359–415, 2024. [Online]. Available: https://jair.org/index.php/jair/article/view/15348

[16] G. Li, R. Gomez, K. Nakamura, and B. He, "Human-centered reinforcement learning: A survey," *IEEE Transactions on Human-Machine Systems*, vol. 49, no. 4, pp. 337–349, 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8708686

[17] P. Christiano, J. Leike, T. Brown, M. Martic, S. Legg, and D. Amodei, "Deep reinforcement learning from human preferences," *ArXiv*, 06 2017. [Online]. Available: https://api.semanticscholar.org/CorpusID:4787508

[18] S. Griffith, K. Subramanian, J. Scholz, C. L. Isbell, and A. L. Thomaz, "Policy shaping: Integrating human feedback with reinforcement learning," in *Advances in Neural Information Processing Systems*, C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger, Eds., vol. 26. Curran Associates, Inc., 2013. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2013/file/e034fb6b66aacc1d48f445ddfb08da98-Paper.pdf

[19] J. Lin, Z. Ma, R. Gomez, K. Nakamura, B. He, and G. Li, "A review on interactive reinforcement learning from human social feedback," *IEEE Access*, vol. 8, pp. 120 757–120 765, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:220569668

[20] W. Bradley Knox and P. Stone, "Tamer: Training an agent manually via evaluative reinforcement," in *2008 7th IEEE International Conference on Development and Learning*, 2008, pp. 292–297. [Online]. Available: https://ieeexplore.ieee.org/document/4640845

[21] C. Wirth, R. Akrour, G. Neumann, and J. Fürnkranz, "A survey of preference-based reinforcement learning methods," *J. Mach. Learn. Res.*, vol. 18, pp. 136:1–136:46, 2017. [Online]. Available: https://api.semanticscholar.org/CorpusID:703818

[22] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robotics and Autonomous Systems*, vol. 57, no. 5, pp. 469–483, 2009. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0921889008001772

[23] P. Abbeel and A. Ng, "Apprenticeship learning via inverse reinforcement learning," *Proceedings, Twenty-First International Conference on Machine Learning, ICML 2004*, 09 2004. [Online]. Available: https://api.semanticscholar.org/CorpusID:207155342

[24] E. Biyik, D. P. Losey, M. Palan, N. C. Landolfi, G. Shevchuk, and D. Sadigh, "Learning reward functions from diverse sources of human feedback: Optimally integrating demonstrations and preferences," *The International Journal of Robotics Research*, vol. 41, pp. 45 – 67, 2022. [Online]. Available: https://doi.org/10.1177/02783649211041652

[25] S. A. Mehta and D. P. Losey, "Unified learning from demonstrations, corrections, and preferences during physical human–robot interaction," *J. Hum.-Robot Interact.*, vol. 13, no. 3, Aug. 2024. [Online]. Available: https://doi.org/10.1145/3623384

[26] W. B. Knox and P. Stone, "Combining manual feedback with subsequent mdp reward signals for reinforcement learning," in *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: Volume 1 - Volume 1*, ser. AAMAS '10. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, 2010, p. 5–12. [Online]. Available: https://dl.acm.org/doi/10.5555/1838206.1838208

[27] W. B. Knox, P. Stone, and C. Breazeal, "Training a robot via human feedback: A case study," in *International Conference on Software Reuse*, 10 2013. [Online]. Available: https://api.semanticscholar.org/CorpusID:266033110

[28] N. Stiennon, L. Ouyang, J. Wu, D. M. Ziegler, R. Lowe, C. Voss, A. Radford, D. Amodei, and P. Christiano, "Learning to summarize from human feedback," in *Proceedings of the 34th International Conference on Neural Information Processing Systems*, ser. NIPS '20. Red Hook, NY, USA: Curran Associates Inc., 2020. [Online]. Available: https://arxiv.org/abs/2009.01325

[29] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P. Christiano, J. Leike, and R. Lowe, "Training language models to follow instructions with human feedback," in *Proceedings of the 36th International Conference on Neural Information Processing Systems*, ser. NIPS '22. Red Hook, NY, USA: Curran Associates Inc., 2022. [Online]. Available: https://arxiv.org/abs/2203.02155

[30] R. Rafailov, A. Sharma, E. Mitchell, S. Ermon, C. D. Manning, and C. Finn, "Direct preference optimization: your language model is secretly a reward model," in *Proceedings of the 37th International Conference on Neural Information Processing Systems*, ser. NIPS '23. Red Hook, NY, USA: Curran Associates Inc., 2023. [Online]. Available: https://arxiv.org/abs/2305.18290

[31] S. Casper, X. Davies, C. Shi, T. K. Gilbert, J. Scheurer, J. Rando, R. Freedman, T. Korbak, D. Lindner, P. Freire, T. Wang, S. Marks, C.-R. Segerie, M. Carroll, A. Peng, P. Christoffersen, M. Damani, S. Slocum, U. Anwar, A. Siththaranjan, M. Nadeau, E. J. Michaud, J. Pfau, D. Krasheninnikov, X. Chen, L. Langosco, P. Hase, E. Bıyık, A. Dragan, D. Krueger, D. Sadigh, and D. Hadfield-Menell, "Open problems and fundamental limitations of reinforcement learning from human feedback," 2023. [Online]. Available: https://arxiv.org/abs/2307.15217

[32] S. Ross, G. J. Gordon, and J. A. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," 2011. [Online]. Available: https://arxiv.org/abs/1011.0686

[33] T. Mandel, Y.-E. Liu, E. Brunskill, and Z. Popović, "Where to add actions in human-in-the-loop reinforcement learning," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31, 02 2017. [Online]. Available: https://api.semanticscholar.org/CorpusID:29170255

[34] A. D. Santis, B. Siciliano, A. D. Luca, and A. Bicchi, "An atlas of physical human-robot interaction," *Mechanism and Machine Theory*, vol. 43, pp. 253–270, 2008. [Online]. Available: https://api.semanticscholar.org/CorpusID:102340737

[35] N. Hogan, "Impedance control: An approach to manipulation: Part i—theory," *Journal of Dynamic Systems Measurement and Control-transactions of The Asme*, vol. 107, pp. 1–7, 03 1985. [Online]. Available: https://doi.org/10.1115/1.3140702

[36] F. Sup, A. Bohara, and M. Goldfarb, "Design and control of a powered transfemoral prosthesis," *The International journal of robotics research*, vol. 27, no. 2, pp. 263–273, 2008. [Online]. Available: https://doi.org/10.1177/0278364907084588

[37] S. K. Au, J. Weber, and H. Herr, "Powered ankle-foot prosthesis improves walking metabolic economy," *IEEE Transactions on Robotics*, vol. 25, no. 1, pp. 51–66, 2009. [Online]. Available: https://ieeexplore.ieee.org/document/4738392

[38] K. Yuan, J. Zhu, Q. Wang, and L. Wang, "Finite-state control of powered below-knee prosthesis with ankle and toe," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 2865–2870, 2011, 18th IFAC World Congress. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1474667016440486

[39] L. J. Hargrove, A. M. Simon, A. J. Young, R. D. Lipschutz, S. B. Finucane, and T. A. Kuiken, "Robotic leg control with emg decoding in an amputee with nerve transfers," *New England Journal of Medicine*, vol. 369, no. 13, pp. 1237–1242, 2013. [Online]. Available: https://www.nejm.org/doi/10.1056/NEJMoa1300126?url_ver=Z39.88-2003&rfr_id=ori:rid:crossref.org&rfr_dat=cr_pub%20%200www.ncbi.nlm.nih.gov

[40] J. Zhang, P. Fiers, K. A. Witte, R. W. Jackson, K. L. Poggensee, C. G. Atkeson, and S. H. Collins, "Human-in-the-loop optimization of exoskeleton assistance during walking," *Science*, vol. 356, no. 6344, pp. 1280–1284, 2017. [Online]. Available: https://www.science.org/doi/abs/10.1126/science.aal5054

[41] Y. Ding, M. Kim, S. Kuindersma, and C. J. Walsh, "Human-in-the-loop optimization of hip assistance with a soft exosuit during walking," *Science Robotics*, vol. 3, no. 15, p. eaar5438, 2018. [Online]. Available: https://www.science.org/doi/abs/10.1126/scirobotics.aar5438

[42] Y. Wen, J. Si, X. Gao, S. Huang, and H. H. Huang, "A new powered lower limb prosthesis control framework based on adaptive dynamic programming," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 9, pp. 2215–2220, 2017. [Online]. Available: https://ieeexplore.ieee.org/document/7508991

[43] Y. Wen, J. Si, A. Brandt, X. Gao, and H. 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. [Online]. Available: https://ieeexplore.ieee.org/document/8613842

[44] J. Si and Y.-T. Wang, "Online learning control by association and reinforcement," *IEEE Transactions on Neural Networks*, vol. 12, no. 2, pp. 264–276, 2001. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/914523

[45] 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)*. IEEE Press, 2019, p. 2831–2837. [Online]. Available: https://doi.org/10.1109/ICRA.2019.8794212

[46] 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, 2022. [Online]. Available: https://ieeexplore.ieee.org/document/9424995

[47] 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. [Online]. Available: https://ieeexplore.ieee.org/document/9442858

[48] R. Wu, M. Li, Z. Yao, W. Liu, J. Si, and H. Huang, "Reinforcement learning impedance control of a robotic prosthesis to coordinate with human intact knee motion," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 7014–7020, 2022. [Online]. Available: https://ieeexplore.ieee.org/document/9786637

[49] F. Heremans, J. Evrard, D. Langlois, and R. Ronsse, "Elsa: A foot-size powered prosthesis reproducing ankle dynamics during various locomotion tasks," *IEEE Transactions on Robotics*, vol. 41, pp. 415–429, 2025. [Online]. Available: https://ieeexplore.ieee.org/document/10770576

[50] H. Herr and A. Grabowski, "Bionic ankle-foot prosthesis normalizes walking gait for persons with leg amputation," *Proceedings. Biological sciences / The Royal Society*, vol. 279, pp. 457–64, 07 2011. [Online]. Available: https://pmc.ncbi.nlm.nih.gov/articles/PMC3234569/

[51] F. Heremans, J. Evrard, D. Langlois, and R. Ronsse, "A lightweight and compact lockable parallel spring enhances the performance of a powered ankle-foot prosthesis," in *2024 10th IEEE RAS/EMBS International Conference for Biomedical Robotics and Biomechatronics (BioRob)*, 2024, pp. 407–412. [Online]. Available: https://ieeexplore.ieee.org/document/10719808

[52] M. Tucker, J. Olivier, A. Pagel, H. Bleuler, M. Bouri, O. Lambercy, J. d. R. Millan, R. Riener, and R. Gassert, "Control strategies for active

lower extremity prosthetics and orthotics: A review," *Journal of NeuroEngineering and Rehabilitation*, vol. 12, p. 1, 01 2015. [Online]. Available: https://www.researchgate.net/publication/270508342_Control_Strategies_for_Active_Lower_Extremity_Prosthetics_and_Orthotics_A_Review

[53] J. Evrard, F. Heremans, and R. Ronsse, "Validation of a heuristic intention detection algorithm for a powered ankle prosthesis across various ambulation tasks," in *2024 10th IEEE RAS/EMBS International Conference for Biomedical Robotics and Biomechatronics (BioRob)*, 2024, pp. 75–81. [Online]. Available: https://ieeexplore.ieee.org/document/10719811

[54] F. Sup, A. Bohara, and M. Goldfarb, "Design and control of a powered knee and ankle prosthesis," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*, 2007, pp. 4134–4139. [Online]. Available: https://ieeexplore.ieee.org/document/4209732

[55] L. Moreira, J. Figueiredo, P. Fonseca, J. P. Vilas-Boas, and C. Santos, "Lower limb kinematic, kinetic, and emg data from young healthy humans during walking at controlled speeds," *Scientific Data*, vol. 8, p. 103, 04 2021. [Online]. Available: https://www.nature.com/articles/s41597-021-00881-3?utm_source=researchgate.net&utm_medium=article

[56] J.-P. Guilloux, M. Seney, N. Edgar, and E. Sibille, "Integrated behavioral z-scoring increases the sensitivity and reliability of behavioral phenotyping in mice: Relevance to emotionality and sex," *Journal of Neuroscience Methods*, vol. 197, no. 1, pp. 21–31, 2011. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S016502701100046X

[57] G. H. Golub, P. C. Hansen, and D. P. O'Leary, "Tikhonov regularization and total least squares," *SIAM Journal on Matrix Analysis and Applications*, vol. 21, no. 1, pp. 185–194, 1999. [Online]. Available: https://doi.org/10.1137/S0895479897326432

[58] A. E. Hoerl and R. W. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems," *Technometrics*, vol. 42, no. 1, pp. 80–86, 2000. [Online]. Available: https://www.tandfonline.com/doi/abs/10.1080/00401706.2000.10485983

[59] S. Diamond and S. Boyd, "Cvxpy: a python-embedded modeling language for convex optimization," *J. Mach. Learn. Res.*, vol. 17, no. 1, p. 2909–2913, Jan. 2016. [Online]. Available: https://arxiv.org/abs/1603.00943

[60] CVXPY Developers, "CVXPY documentation: Solver features," 2026. [Online]. Available: https://www.cvxpy.org/tutorial/solvers/index.html

[61] L. Gabert, S. Hood, M. Tran, M. Cempini, and T. Lenzi, "A compact, lightweight robotic ankle-foot prosthesis: Featuring a powered polycentric design," *IEEE Robotics & Automation Magazine*, vol. 27, no. 1, pp. 87–102, 2020. [Online]. Available: https://ieeexplore.ieee.org/document/8963856

[62] M. Grimmer and A. Seyfarth, *Mimicking Human-Like Leg Function in Prosthetic Limbs*. Springer Netherlands, 07 2014, pp. 105–155. [Online]. Available: https://www.researchgate.net/publication/263814767_Mimicking_Human-Like_Leg_Function_in_Prosthetic_Limbs

[63] J. Ralph, W. Gray, and M. Schoelles, "Squeezing the balloon: Analyzing the unpredictable effects of cognitive workload," *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 54, pp. 299–303, 09 2010. [Online]. Available: https://www.researchgate.net/publication/273597143_Squeezing_the_Balloon_Analyzing_the_Unpredictable_Effects_of_Cognitive_Workload

UNIVERSITÉ CATHOLIQUE DE LOUVAIN
École polytechnique de Louvain
Rue Archimède, 1 bte L6.11.01, 1348 Louvain-la-Neuve, Belgique | **www.uclouvain.be/epl**