

Feedback-Error Learning Control for Powered Assistive Devices*

Pedro Nuno Fernandes, Joana Figueiredo, Juan C. Moreno, Cristina P. Santos

Abstract—Active orthoses (AOs) are becoming relevant for user-oriented training in gait rehabilitation. This implies efficient responses of AO’s low-level controllers with short time modeling for medical applications. This thesis investigates, in an innovative way, the performance of Feedback-Error Learning (FEL) control to time-effectively adapt the AOs’ responses to user-oriented trajectories and changes in the dynamics due to the interaction with the user. FEL control comprises a feedback PID controller and a neural network feedforward controller to promptly learn the inverse dynamics of two AOs. It was carried out experiments with able-bodied subjects walking on a treadmill and considering external disturbances to user-AO interaction. Results showed that the FEL control effectively tracked the user-oriented trajectory with position errors between 5% to 7%, and with a mean delay lower than 25 ms. Compared to a single PID control, the FEL control decreased by 16.5% and 90.7% the position error and delay, respectively. Moreover, the feedforward controller was able to learn the inverse dynamics of the two AOs and adapt to variations in the user-oriented trajectories, such as speed and angular range, while the feedback controller compensated for random disturbances. FEL demonstrated to be an efficient low-level controller for controlling AOs during gait rehabilitation.

Keywords- *gait rehabilitation; robotic assistive devices; human-centered robotics; feedback error learning control; neural network*

I. INTRODUCTION

Gait rehabilitation has been proposed as a more appropriate intervention to restore motor abilities, which are boosted by user-oriented and repetitive gait training [1], [2]. Wearable robotic assistive devices, such as active orthoses (AOs), are becoming a prominent intervention for fostering user-oriented gait training by tuning the AOs’ mechanical work in accordance with the real-time evaluation of the user’s needs, approaching assist-as-needed (AAN) strategies [2], [3]. Additionally, AOs have embodied assistive strategies, such as the trajectory

*This work has been supported in part by the Fundação para a Ciência e Tecnologia (FCT) with the Reference Scholarship under Grant SFRH/BD/108309/2015, and part by the FEDER Funds through the Programa Operacional Regional do Norte and national funds from FCT with the project SmartOs -Controlo Inteligente de um Sistema Ortótico Ativo e Autónomo—under Grant NORTE-01-0145-FEDER-030386, and by the FEDER Funds through the COMPETE 2020—Programa Operacional Competitividade e Internacionalização (POCI)—with the Reference Project under Grant POCI-01-0145-FEDER-006941.

Pedro Nuno Fernandes, Joana Figueiredo, and Cristina P. Santos are with Center for MicroElectroMechanical Systems (CMEMS), University of Minho, Guimarães, Portugal (e-mail: {a68385,id6003}@alunos.uminho.pt, cristina@dei.uminho.pt). Juan C. Moreno is with Neural Rehabilitation Group, Cajal Institute, Spanish National Research Council, Madrid, Spain (e-mail: jc.moreno@csic.es).

tracking control based on user-oriented trajectories to improve the movement coordination [4].

Nowadays, the principles of the human motor control system start to influence the design of these active orthotic systems, and it has been proposed bioinspired architectures with emphasis on the human-AOs interaction to deliver an effective response in accordance with the user’s needs [2].

Feedback and feedforward controllers play an important role in the low-level layer of these bioinspired architectures. They should guarantee that the AOs’ actuation system has accurate and time-effective responses to track the position- or torque-based reference trajectory and the stability of the human-AOs interaction. Feedback controllers, such as the Proportional-Integral-Derivative (PID) and its variants, are the low-level controllers mostly applied in AOs due to their feasibility and mathematical straightforwardness, without an arduous implementation process [5], [6], [7]. Nevertheless, adopting user-oriented assistance strategies merely based on PID control might present limitations, such as steady-state errors, measurement noise in the feedback loop, and low robustness to disturbances and human-AO dynamic changes. This limited adaptability may lead the human-AO interface to instability [8]. On the other hand, optimal and robust controls have been used to minimize the influence of disturbances over the control response and to confer adaptability in different rehabilitative scenarios [9]. However, the applicability of these controls for clinical purposes is limited by the considerable complex mathematical understanding and modeling [10].

The development of time-effective low-level controllers for AOs capable of providing low steady-state errors, emphasizing adaptiveness to different walking scenarios without requiring system modeling, becomes imperative. To address these features, Feedback-error learning (FEL) control, proposed by Kawato [11], has been applied in robotic systems [10]–[12]. FEL, bioinspired on the learning process of the human motor cortex, is a hybrid control combining a feedback controller with a feedforward controller capable of learning the inverse dynamics of the robotic system using the feedback control command as the error signal for the learning (feedback error). The feedforward controller usually includes regression techniques to simplify and shorten the learning of the inverse dynamics for real-time and medical applications not requiring an explicit model of the robotic system neither separating the learning and control phase [10], [12]. FEL allows the feedback controller to control the system while the inverse model is being learned in real-time. The learned inverse model is capable to track the reference trajectory. Consequently, this reduces the

feedback contribution to the final control command. The overall performance of FEL beneficiaries from the adaptive and anticipatory features of the feedforward control to adapt to changes in AOs' dynamics and enable time-effective fast movements while the feedback controller is able to compensate for disturbances [12].

So far, the application of FEL in real robotic assistive devices has been limited to upper limb assistance by functional electrical stimulation [12] and neuroprosthetics [10] and has not yet been tested in lower limb assistive devices such as AOs. This thesis innovatively investigates the FEL effects in AOs for gait rehabilitation. The main goal of this thesis is to present the real-time implementation of FEL control to time-effectively adapt the AOs' responses to user-oriented trajectories and changes in the dynamics due to the interaction with the user. It was applied an artificial neural network (ANN) as the feedforward controller [11] to learn the inverse dynamics of the AO taking the output of a PID command (feedback controller) as an error signal. The PID control was also used to control the AOs during the real-time learning phase of ANN (performed at 1 kHz), and to handle with disturbances as the learning is completed. The FEL control was tuned and validated in two AOs, a powered knee orthosis (PKO) and a powered ankle-foot orthosis (PAFO), using position-based trajectory tracking assistive strategies for gait rehabilitation of the knee and ankle joints, respectively. These AOs are embodied into a smart, active orthotic system – SmartOs – controlled by a bioinspired architecture. The ANN was designed using normalized inputs to allow variations in the user-oriented trajectory, such as gait speed (0.8, 1.0, and 1.2 km/h) and trajectory angular range.

II. METHODS

This thesis presents the FEL as a new low-level controller for the smart, active lower limb orthotic system - SmartOs. It includes a right PKO and a right PAFO synergistically linked to a wearable motion lab (with biomechanical and muscular sensors) for the post-stroke robotic-based gait rehabilitation with a user-oriented, repetitive gait training. SmartOs was designed with a bio-inspired control architecture structured into three levels [2]. Currently, this control architecture includes a set of AAN assistive strategies, namely: adaptive trajectory tracking control, adaptive impedance control, and myoelectric control, using the PID as the low-level control.

Fig. 1 illustrates the general overview of the SmartOs modules involved in this thesis. The position-based trajectory tracking control was selected as the control assistive strategy.

A. PKO and PAFO Assistive Devices

The used PKO and PAFO (see Fig. 1.a) are modules of the lower limb H2-exoskeleton (Technaid S.L., Spain) [4]. Each AO has embedded sensors (potentiometer, strain gauges, and hall effect sensors) and an actuation technology comprising an electrical actuator (brushless DC motor EC60-100W (Maxon)) coupled to a gearbox CSD20-160-2A (Harmonic Drive). The Control Area Network (CAN) was used to establish a deterministic communication among the AOs and the low-level processing unit, the STM32F407VGT microcontroller running freeRTOS OS that manages the low-level controllers at 1 kHz.

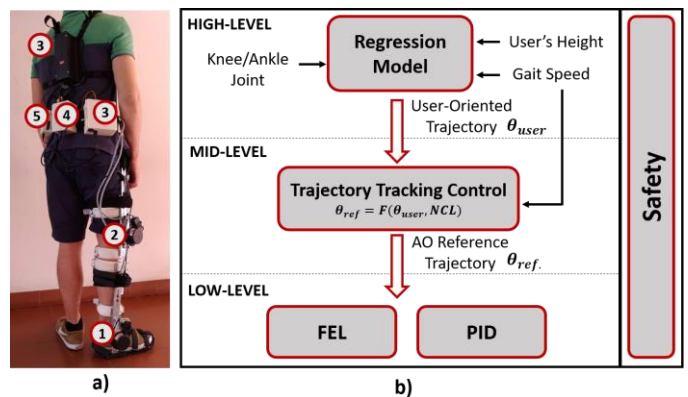


Figure 1. SmartOs overview: a) User wearing SmartOs modules: 1- PAFO, 2- PKO, 3- Power supply system, 4- central processing unit, 5- low-level processing unit; b) hierarchical control architecture.

The reference trajectories are set in the high- and mid-level controllers running at 100 Hz in a central processing unit, the Raspberry Pi 3 [13] with the Ubuntu Mate OS [14]. A serial interface was used, through Universal Asynchronous Receiver-Transmitter (UART), to manage the communication between the central processing unit and the low-level processing unit. The system was powered by a LiFePO4 battery of 24 V and 3 Ah, which enables at least 10 hours of autonomy.

B. High- and Mid-level Controllers

For the position-based trajectory tracking control, the high-level includes a regression model dependent on the user's height and gait speed [15] to generate user-oriented position trajectories (θ_{user}) for the knee and ankle joints. It was set the gait speeds as 0.8 km/h, 1.0 km/h, and 1.2 km/h as they are the target speeds to be used with end-users.

As shown in Fig. 1, the mid-level controller sets the AO's reference trajectory (θ_{ref}) as the user-oriented angular trajectories (θ_{user}) parameterized in accordance with the gait speed. As such, (1), empirically found, sets the Number of Control Loops (NCL), each lasting 1 ms, that must occur to update the value of the reference trajectory. This speed-parameterization aims to ensure user-AO coordination.

$$NCL = -34.62 * Gait\ Speed + 107.31 \quad (1)$$

C. Low-level Controller

In the scope of this thesis, the low-level controller covers a position-based control loop to ensure that the AO's angular position matches the user-oriented trajectory. Previously to the FEL control, it has been achieved through PID control. Both controls were implemented in SmartOs as follows.

1) *PID Control*: It was tuned using the Ziegler-Nichols method with a similar procedure as was reported in [16]. This tuning considered the practical application of PKO and PAFO in a rehabilitation scenario to avoid abrupt movements that can cause discomfort to the user and to avoid oscillations and overshoot in the actuator's response. It was used a $K_p = 90$ and $K_i = K_d = 1.5$ as PID gains. The PID controller implements the control law presented in (2) to compute the feedback command (u_{fb}), where e_k and e_{k-1} correspond to the current

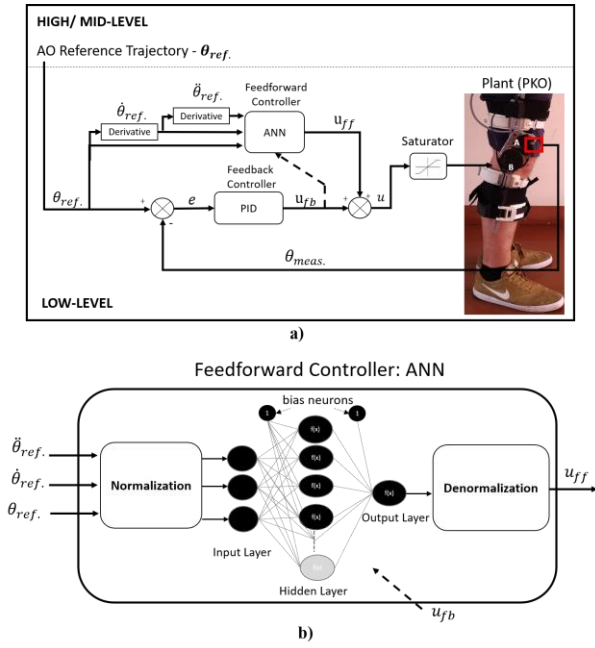


Figure 2. FEL control: a) FEL control loop for PKO. θ_{ref} is the reference angular position; $\dot{\theta}_{ref}$ is the reference angular velocity; $\ddot{\theta}_{ref}$ is the reference angular acceleration θ_{meas} is the measured angular position; e is the position error; u_{fb} is the feedback command; u_{ff} is the feedforward command; u is the final control command; A is the potentiometer and B is the actuator, a) ANN-based feedback controller.

and previous error between the reference (θ_{ref}) and measured (θ_{meas}) angular position by a potentiometer embedded on the AO, as illustrated in Fig. 2.

$$u_{fb} = K_p e_k + K_i \sum_{n=1}^k e_n \Delta t + K_d \frac{e_k - e_{k-1}}{\Delta t} \quad (2)$$

2) *FEL Control*: It implies a feedback and a feedforward controller to command the AO in the desired way, as depicted in Fig. 2. The PID feedback controller provides control commands and guarantees stability during the real-time learning of the AOs' inverse dynamics model and to compensate disturbances as the learning phase is completed. The feedforward controller learns the inverse dynamics model using an ANN as the regression technique, as proposed in [10], [11], [17]. It was involved a supervised ANN to infer the inverse dynamics due to its proper estimation performance, good generalization, and its capability to map non-linearities [18]. Note that two inverse models were learned, one for each AO.

The ANN was trained in real-time following a supervised learning training method, running at 1 kHz. The Multi-Layer Perceptron (MLP) was the adopted structure for the ANN, as in [10], [17]. As illustrated in Fig. 2.b), each ANN has 3 layers as follows: the input layer with 3 neurons for the 3 input trajectories, the reference position (θ_{ref}), the speed ($\dot{\theta}_{ref}$) and acceleration ($\ddot{\theta}_{ref}$); the hidden layer with 4 and 5 neurons for PKO and PAFO, respectively; and, the output layer with 1 neuron, i.e., the feedforward command (u_{ff}). The hyperbolic tangent and the identity were the used activation functions for the hidden and output layers, respectively. It was included bias

nodes in the hidden and output layers (see Fig. 2.b)) to increase the flexibility of the ANN to fit the input data.

As the learning method, it was used the backpropagation algorithm that includes a forward and a backward phase. The forward phase predicts feedforward commands (u_{ff}) for the given inputs and based on the current weights. The backward phase updates the weights based on the gradient descent (i.e., minimization) of the current feedback command (u_{fb} , PID command serves as the cost function) with respect to the network's weights change. A stochastic gradient descent (SGD) approach was implemented such that the weights are updated based on inputs presented in random order on each training iteration [19]. To implement SGD in real-time and to meet the temporal requirement of 1 ms (1 kHz), it was applied the Adaptive Moment Estimation (ADAM) optimizer [20] that is an SGD approach for computationally efficient stochastic optimization that only requires first-order gradients with little memory requirements [20]. ADAM used adaptive learning rates per weight connection, providing adaptability to the ANN training and decreasing the training time.

Table I presents the empirically found set-up for training the ANN for PKO and PAFO, namely the number of neurons in the hidden layer, the maximum learning rate, and the initial weights in the hidden (w_{hid}) and output (w_{out}) layers. This parameters' setting resulted from a trade-off between the ANN performance and the real-time temporal requirements (1 kHz).

TABLE I. ANN'S SET-UP FOR TRAINING PHASE

| AO | Hidden Neurons | Initial Weights | Learning Rate |
|------|----------------|--|-----------------------------|
| PKO | 4 | $-0.058 \leq w_{hid} \leq 0.058$ $-0.05 \leq w_{out} \leq 0.05$ | Adaptive (≤ 0.0001) |
| PAFO | 5 | $-0.058 \leq w_{hid} \leq 0.058$ $-0.045 \leq w_{out} \leq 0.045$ | Adaptive (≤ 0.00001) |

For both training and recall phases, the input signals of ANN (θ_{ref} , $\dot{\theta}_{ref}$, $\ddot{\theta}_{ref}$) were normalized between [-1;1] in an attempt to provide adaptability and versatility to the ANN. This enables to handle with variations in the reference trajectories, such as gait speed and trajectory angular range, and the application of the tuned ANN in different user-oriented trajectories. Moreover, it reduces the estimation error and accelerates the training phase [21]. As presented in Fig. 2.b), the predicted output of the ANN (u_{ff}) was denormalized to the maximum operating magnitude of the control commands, experimentally set to [-2500; 2500], i.e., to the maximum values of the AOs' pulse-width modulation.

Moreover, in both training and recall phases, the computed feedback (u_{fb}) and feedforward (u_{ff}) commands are summed to get the final control command (u) to be applied to the AOs. The final command was limited to [-2500; 2500] with a saturator (Fig. 2.b)) to protect the operability of AOs. When three gait cycles are performed, the contribution of the feedback control command (u_{fb} (%)) to the final control command (u) is estimated, as given by (3), considering $\overline{u_{ff}^2}$ as the mean squared feedforward contribution.

$$u_{fb}(\%) = 100\% \times \frac{u_{ff}^2}{u} \quad (3)$$

The training phase ends when the contribution of the feedback controller is equal or lower to 5% ($u_{fb} \leq 5\%$) since it was empirically verified that the feedback contribution did not reach much lower than 5%. Then, the recall phase starts and the learned inverse dynamics models of PKO and PAFO are able to predict time-effective control commands to timely track the reference trajectory (θ_{ref}) while the feedforward controller is released from this task to compensate for disturbances.

All software interfaces for FEL real-time implementation were done in C language in the STM32F407VGT microcontroller, using its hardware abstraction layer libraries and following the time-effectiveness architecture of SmartOs.

D. Safety Measures

Different safety measures were included throughout the control architecture (as illustrated in Fig. 1.b) to safeguard the human-AO interaction. The mechanical limits, i.e., the range of motion (ROM) of PKO and PAFO were limited to $[3^\circ; 98^\circ]$ and $[-18^\circ; 18^\circ]$, respectively. These intervals are contained within the ROMs of the user's joints so that the AOs do not compromise the users' joints integrity by applying overextension or over flexion motions. Additionally, a proper and comfortable attachment and alignment of the AOs with the user's lower limbs and joints, respectively, were considered. Lastly, the therapy can only be interrupted or stopped in a smooth and safe manner when the user is touching on the ground (tracked by force resistive sensors embedded on AOs).

III. FEL CONTROL PERFORMANCE

A. Experimental Tuning

The tuning of FEL control focused on the feedforward controller. No tuning was made in the feedback controller since it was used the already tuned PID control. The tuning of the feedforward controller consisted of the real-time training of the ANN by considering the normalized inputs randomly presented to the ANN (SGD approach) and the actual PID commands as the feedback error to be minimized. This procedure was separately performed for the PKO and PAFO with the respective reference trajectories adjusted with NCL for 1 km/h and 0.8 km/h (speeds randomly selected), respectively.

During the real-time training of the feedforward controller, the ANN is trying to learn the AOs' inverse dynamics models. Consequently, the feedforward commands that contribute to the final command (u) could lead the AOs to exceed their mechanical limits and compromising their integrity. Therefore, for the first training phase of the ANN, it was decided to modify the original reference position trajectory. It was added an offset of 15° to the original knee reference trajectory and smoothed the original ankle reference trajectory with an attenuation gain of 40% and 4° of offset. With this procedure, it is possible to get the inverse dynamics models of PKO and PAFO while operating far from their mechanical limits. As illustrated in Fig. 3, the trajectory modification was possible since the normalized signals of the original and modified trajectory are equal. This

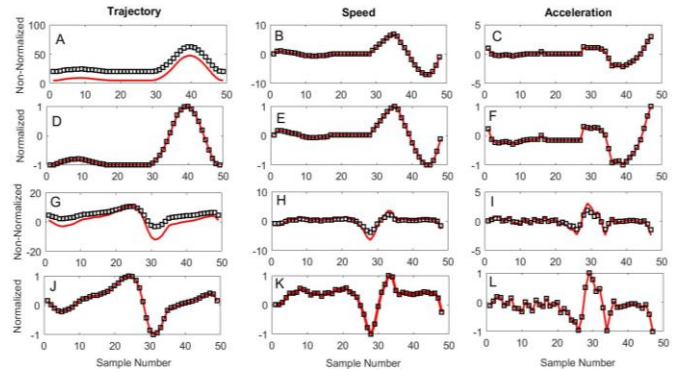


Figure 1. Original (red) and modified (black) input signals for PKO (A-C non-normalized signals, D-F normalized signals) and PAFO (G-I non-normalized signals, J-L normalized signals).

finding shows that the ANN would receive equal inputs in both situations. Note that the initial AOs' angular position was set close to the first value of the non-normalized input signals.

The pre-trained ANNs were subsequently retrained using the original reference trajectories to get the final configuration for the ANNs. The pre-trained state is an important advantage to clinical application once it may decrease the training time for new user-oriented trajectories, as reported in [12].

B. Experimental Validation

FEL control was validated in PKO and PAFO following three experimental procedures. In the first one, it was validated the robustness, repeatability, and time-effectiveness of the FEL control, with focus on the ANN-based feedforward controller. Moreover, it was studied the stability of FEL control commands along the time such that to investigate the possible inclusion of FEL as a low-level controller in SmartOs. For these purposes, a set of experiments were separately performed for the PKO and PAFO without involving the user, considering the original reference trajectories tuned for 0.8 km/h, 1.0 km/h, and 1.2 km/h. Each experiment lasted 5 minutes.

The second procedure covers the validation of FEL control considering the user-AO interaction to investigate the FEL adaptability to changes in the dynamics due to the interaction with the user. For this purpose, two able-bodied subjects (a male and a female) with 25.5 ± 0.71 years old, the height of 1.69 ± 0.1 m, and the weight of 64.50 ± 14.84 kg were involved. They gave their informed consent to take part in the experiment. They were asked to walk at 0.8, 1.0, and 1.2 km/h in level-ground on a treadmill with the PKO and PAFO. Each participant performed 3 trials with a duration of 5 minutes.

With the third procedure, it was studied the FEL response to external perturbations to the user-AO interaction. Under the same conditions as the ones taken for the second procedure, the participants were informed and asked to counteract the PKO and PAFO in the terminal stance and in the initial stance phase, respectively, preventing both AOs from increasing their position. The participants performed the disturbances in self-selected gait cycles. These disturbances were selected given their high prevalence during a gait therapy.

C. Performance Measures and Metrics

To evaluate the FEL performance while comparing with the single PID feedback controller, it was collected the position trajectory measured by the potentiometer (θ_{meas}), the feedback (u_{fb}) and feedforward commands (u_{ff}), for PKO and PAFO. All signals were sampled at 100 Hz. As performance metrics, it was computed the phase delay (ms), the angular position error ($^\circ$) and its normalized root mean square error (NRMSE (%)), and the feedback command contribution (u_{fb} (%)).

IV. RESULTS AND DISCUSSION

A. FEL Tuning

It was analyzed the AOs' state during FEL tuning, by considering the evaluation of the measured position trajectory, the control commands, and the angular position error. For instance, Fig. 4 presents the results achieved during the ANN training using the modified trajectory for the PKO at 1 km/h under three different periods of FEL control: Initial Phase, Middle Phase, and Final Phase.

During the Initial Phase (first 11 s), the feedforward controller is starting to tune its ANN. Consequently, feedforward contribution to the final control command is not significant, being the PID control responsible to track the reference trajectory (Fig. 4.B). It is possible to see in Fig. 4.A that the measured trajectory is delayed 210 ms comparatively to the reference one. Hence, the angular position error varies from -20° to 20° , as demonstrated in Fig. 4.C.

In the Middle Phase (between the first 11 s until 40 s), the FEL controller is learning the inverse dynamics model of the PKO. Therefore, the measured trajectory starts to decrease its phase difference to the reference signal to a mean value of 6 ms (considering three gait cycles presented in Fig. 4.D). To correct this delay, the feedforward controller produces commands that when summed with PID commands lead the PKO to perform a trajectory with 10° more than the reference trajectory, as illustrated in Fig. 4.D. This happens because the ANN has not learned the inverse dynamics model with the best performance yet. Fig. 4.E highlights an increase and decrease in the feedforward command and the PID command, respectively, relatively to the Initial Phase. As depicted in Fig. 4.F, the position error decreased, varying from -20° to 10° .

For the Final Phase (after 40s), when the ANN already learned the inverse dynamics of the PKO, it was verified that the FEL control has successfully aligned the PKO trajectory with the reference one (Fig. 4.G) and corrected the amplitude divergence previous observed. The position error decreased by 75% compared to the Initial Phase. In this phase, the feedback controller contribution was 4.4% (Fig. 4.K).

In overall, the findings of FEL tuning indicate that the designed ANN was capable of correctly learning the inverse dynamics for both AOs. When the learning phase finished, the ANN can timely track the reference trajectory, discharging the feedback controller for this task. It was observed that the learning phase lasted 90 s (approximately 25 gait cycles) and 315 s (approximately 70 gait cycles) for PKO and PAFO, respectively. The temporal difference is mainly explained by

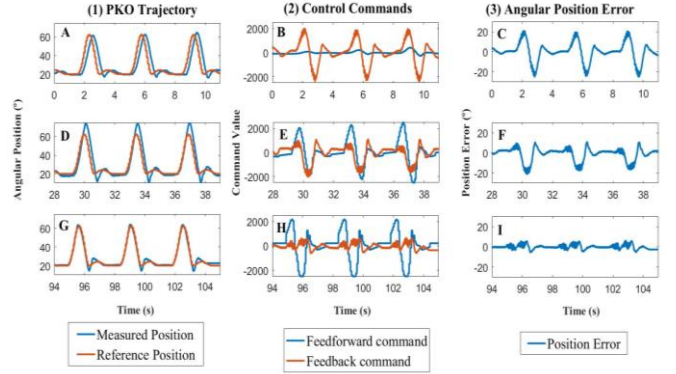


Figure 2. FEL tuning over three periods: Initial Phase (A-C), Middle Phase (D-F), and Final Phase (G-I). Random example for PKO using the modified trajectory tuned for 1 km/h.

the different learning rates used in both cases (Table I). Additionally, it was verified that the run time of FEL control loop (0.25 ms) is lower than the one required by the bioinspired control architecture of SmartOs (1 ms). These temporal findings suggest that the techniques applied for approaching a real-time implementation were effective to avoid long-time periods in the training phase. Furthermore, few iterations were required to tune the pre-trained ANN.

It is important mentioning that the modification made in the reference trajectory for the first learning moment protected the AOs. Fig. 4.D shows a practical situation, during the Middle Phase, where a 10° -deviation would put the device out of its lower operating limit (3°). The 10° -deviation did not damage the PKO since the added 15° -offset was enough to deviate the PKO from its lower mechanical limit (3°). A similar effect was observed in PAFO for the lower and upper limits.

B. FEL Validation

This sub-section presents the validation of FEL control for PKO and PAFO considering their ANNs tuned for the original reference trajectory. It was also compared the FEL performance with the single PID low-level controller of the SmartOs.

1) *PKO validation*: Table II presents the results for FEL and PID controls with and without the user's interaction (first and second procedures, respectively), namely the NRMSE, phase delay, and the contribution of the feedback control (u_{fb}).

For all tested speeds in the first procedure, the FEL control presented a mean NRMSE of 5.13% and a mean phase delay of 13 ms, with the feedback controller only contributing 5.9% for the final control command. Additionally, it was observed that the ANN tuned for 1 km/h well-predicted the feedforward commands for 0.8 and 1.2 km/h speeds by only changing the normalization limits. This finding is supported by the low NRMSE ($5.17 \pm 0.04\%$ and $4.68 \pm 0.03\%$) and delays (20 ± 0.001 ms) found for 0.8 and 1.2 km/h speeds, respectively. These outcomes state that the FEL control provides robust and time-effective control commands in a repetitive manner along the gait cycle independently of the speed variation.

Regarding the validation with the user-PKO interaction (second procedure), a mean NRMSE of 5.87%, a mean phase delay of 12.5 ms, and a mean contribution of 6.52% of the

feedback controller were yielded. Fig. 5 also shows the consistency of FEL performance, presenting low position error and delay, and repetitive time-effective feedforward commands even when considering the user-PKO interaction under different gait speeds. Comparing both procedures, i.e., in the presence or not of interaction with the user, it was noted that i) the phase delay decreased by 0.5 ms; ii) the mean NRMSE increased by 0.74%; and, iii) the mean feedback contribution increased 1% to the final control command. The two last findings could result from the load added (i.e., the user) in the PKO, that may difficult the controller to follow the reference trajectory.

Furthermore, it was also evaluated the FEL performance in the presence of external disturbances to the user-PKO interaction. This analysis was limited to disturbances in the terminal stance phase, like the one marked in Fig. 6 at 58 s. Fig. 6.B shows that at the instant of the disturbance the feedback command increases once the position error grows (see Fig. 6.C) due to the disturbance. The increased contribution of the feedback control aimed to prevent the FEL control from fall into an instability state and from damaging the user-PKO interaction.

On the other hand, the feedforward controller command stayed periodic (Fig. 6.B), as the reference signals and the learned inverse dynamics model did not change.

TABLE II. RESULTS (MEAN \pm STD) OF FEL AND PID CONTROLS FOR FIRST (I) AND SECOND (II) PROCEDURES WITH PKO AT DIFFERENT SPEEDS

| | Speed (km/h) | Control | NRMSE (%) | Delay (ms) | u_{fb} (%) |
|----|--------------|---------|------------------|------------------|-----------------|
| I | 0.8 | PID | 21.58 \pm 0.17 | 260 \pm 0.046 | - |
| | | FEL | 5.17 \pm 0.04 | 20 \pm 0.001 | 6.9 \pm 3.6 |
| | 1.0 | PID | 22.33 \pm 0.16 | 210 \pm 0.024 | - |
| | | FEL | 5.54 \pm 0.08 | 1 \pm 0.001 | 4.5 \pm 0.15 |
| | 1.2 | PID | 22.0 \pm 0.16 | 230 \pm 0.05 | - |
| | | FEL | 4.68 \pm 0.03 | 20 \pm 0.001 | 6.3 \pm 0.53 |
| II | 0.8 | PID | 21.58 \pm 0.17 | 260 \pm 0.046 | - |
| | | FEL | 5.55 \pm 0.04 | 15 \pm 0.007 | 6.94 \pm 0.67 |
| | 1.0 | PID | 22.33 \pm 0.16 | 210 \pm 0.024 | - |
| | | FEL | 5.69 \pm 0.04 | 1 \pm 0.004 | 6.1 \pm 0.95 |
| | 1.2 | PID | 22.0 \pm 0.16 | 230 \pm 0.05 | - |
| | | FEL | 6.37 \pm 0.013 | 22.5 \pm 0.004 | 6.52 \pm 0.23 |

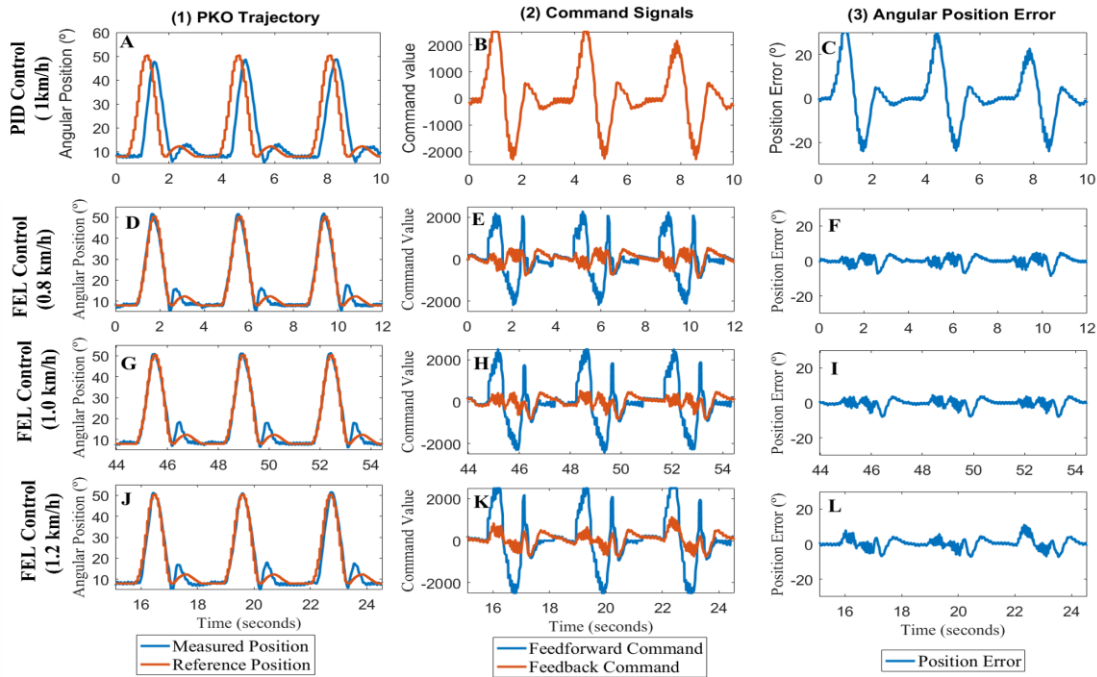


Figure 5. Results of PID (A-C) and FEL (D-L) controls in the recall phase considering the user-PKO interaction for 0.8, 1, and 1.2 km/h speeds.

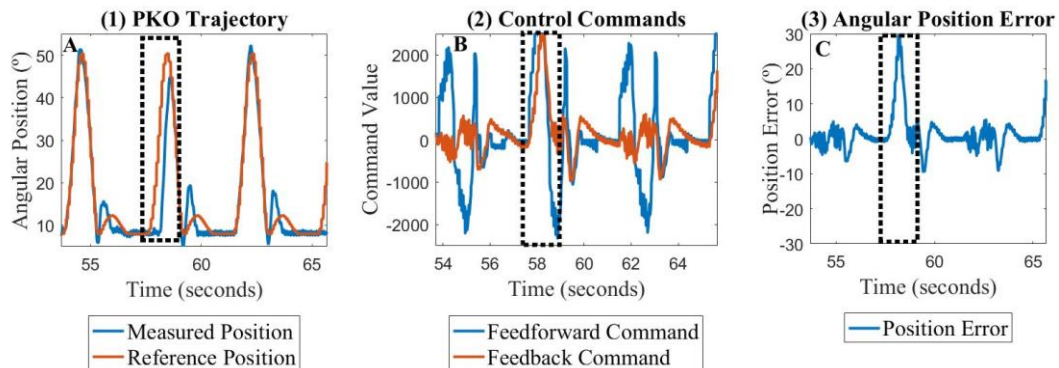


Figure 3. Results of FEL control to external disturbances to the user-PKO interaction for 1 km/h.

TABLE III. RESULTS (MEAN \pm STD) OF FEL AND PID CONTROLS FOR FIRST (I) AND SECOND (II) PROCEDURES WITH PAFO AT DIFFERENT SPEEDS

| | Speed (km/h) | Control | NRMSE (%) | Delay (ms) | u_{fb} (%) |
|----|--------------|---------|------------------|----------------|----------------|
| I | 0.8 | PID | 22.28 \pm 0.14 | 250 \pm 0.28 | - |
| | | FEL | 5.63 \pm 0.04 | 20 \pm 0.03 | 8.5 \pm 0.47 |
| | 1.0 | PID | 24.53 \pm 0.18 | 260 \pm 0.27 | - |
| | | FEL | 6.54 \pm 0.04 | 20 \pm 0.03 | 7.9 \pm 0.07 |
| | 1.2 | PID | 26.99 \pm 0.17 | 250 \pm 0.28 | - |
| | | FEL | 7.68 \pm 0.05 | 20 \pm 0.03 | 8.6 \pm 0.94 |
| II | 0.8 | PID | 22.28 \pm 0.14 | 250 \pm 0.28 | - |
| | | FEL | 6.95 \pm 0.008 | 30 \pm 0.01 | 6.2 \pm 0.01 |
| | 1.0 | PID | 24.28 \pm 0.18 | 260 \pm 0.27 | - |
| | | FEL | 5.99 \pm 0.002 | 20 \pm 0.001 | 6.9 \pm 0.3 |
| | 1.2 | PID | 26.99 \pm 0.17 | 250 \pm 0.28 | - |
| | | FEL | 6.58 \pm 0.004 | 25 \pm 0.007 | 4.84 \pm 1.9 |

2) *PAFO validation*: The performance of FEL and PID controls with and without the user's interaction (first and second procedures, respectively) was evaluated considering the NRMSE, phase delay, and the contribution of the feedback control, as presented in Table III.

Results for the first procedure including all tested speeds demonstrate that FEL achieves a mean phase delay of 23.3 ms, a mean NRMSE of 6.6% with low position errors (0.8 km/h, from -5° to 5° ; 1.0 km/h, from -5° to 5° ; 1.2 km/h, from -6° to 4°). These outcomes were observed under a predominant contribution of the feedforward controller over the feedback controller (mean contribution of 8.3% for all speeds). Thus, the assistance is almost given entirely by the feedforward controller. FEL control was able to learn the inverse dynamics of PAFO, providing accurate and time-effective control commands in a repetitively and independently of the speed variation. It was verified that the same set-up of FEL control can successfully be applied for different gait speeds due to the equal temporal performance with slight deviations in NRMSE under speed variation.

Moreover, it was evaluated the FEL performance considering the user-PKO interaction (second procedure). Fig. 7 shows the obtained results, randomly selected for one subject, walking on a treadmill for 0.8, 1.0 and 1.2 km/h. In this procedure, it was observed a mean NRMSE of 6.51%, a mean phase delay of 25 ms and a mean contribution of 5.8% of the

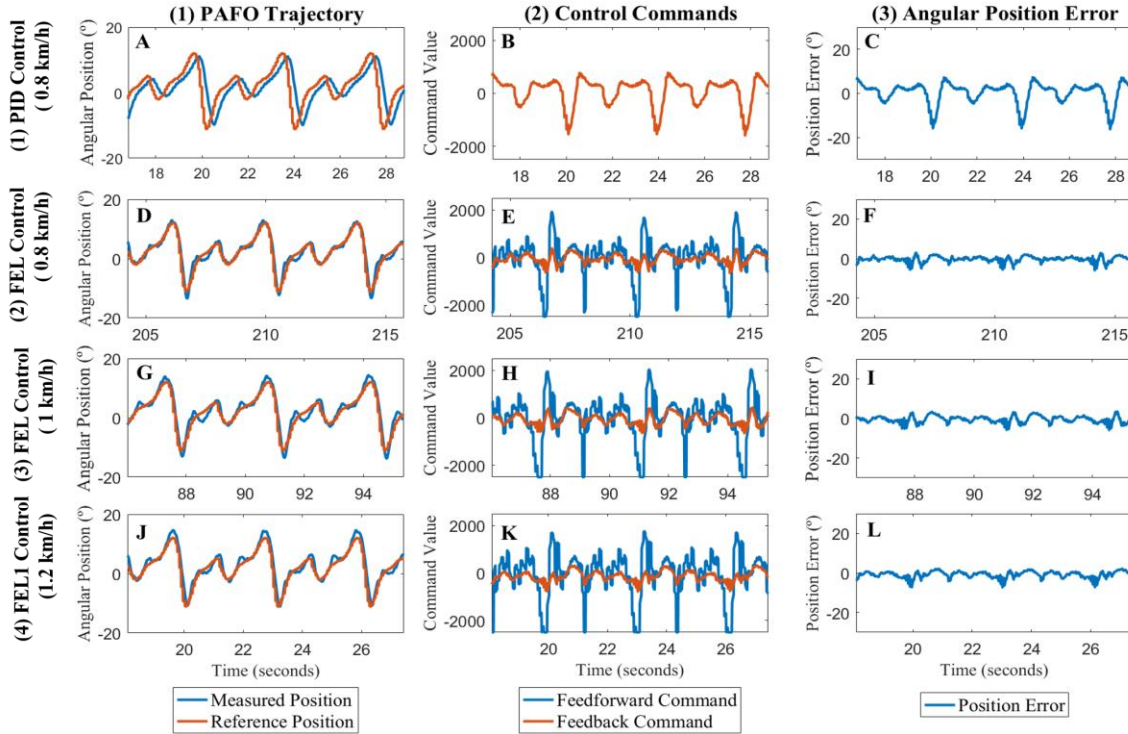


Figure 5. Results of PID (A-C) and FEL (D-L) controls in the recall phase considering the user-PAFO interaction for 0.8, 1, and 1.2 km/h speeds.

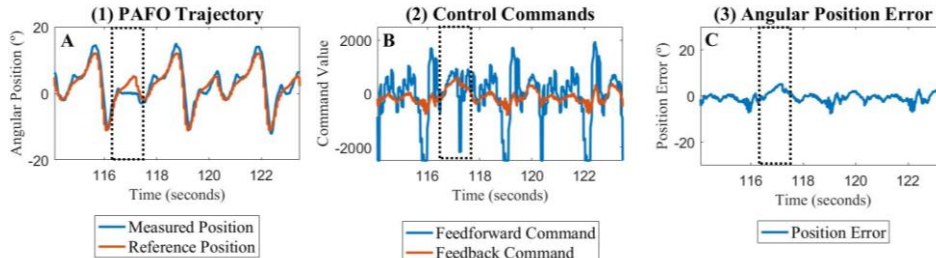


Figure 5. Results of FEL control to external disturbances to the user-PAFO interaction for 1 km/h.

feedback controller to the final control command. It is possible to verify that FEL performs similarly in the presence or not of the interaction with the user given that: i) mean NRMSE decreased 0.09%; ii) the phase delay increased 1.7 ms; and, iii) the feedback contribution decreased 2.5%. The decrease in the NRMSE and consequent reduction of feedback contribution might be due to the load added to the PAFO that stopped it to exceed the reference signal since it was observed in the first procedure a general tendency to the PAFO position exceeds -5° and 5° the reference trajectory. In overall, the findings show that FEL can be used as an adaptive control to deal with dynamics changes due to interaction with the user. It also yielded time-effective commands since the AO's position gets the reference position at least 40 ms before the reference position is updated (every 65 ms for 1.2 km/h, the more demanding condition).

Furthermore, it was validated the FEL performance in the presence of external disturbances to the ones introduced by the user-PAFO interaction, as depicted in Fig. 8. Such disturbances increased the position error due to the displacement between the reference and measured position. In responding to this disturbance, the FEL control augments the feedback controller contribution (increased PID command) while the feedforward command stayed periodic since the PAFO dynamics did not change. These outcomes demonstrate that the feedback control is charged to compensate for small disturbances while the feedforward control has the higher actuation command to drive the desired movement.

C. Comparative analysis with PID control

For the assistance provided by the PKO under all tested speeds, the PID control achieved an NRMSE around 22% and a phase delay of 230 ms. Consequently, it was observed that the FEL control decreased, approximately, the NRMSE 16.5% and 93% for the phase delay. In the ankle-foot assistance by PAFO, the PID achieved a mean NRMSE of 24%, a position error varying from -16° to 7° , and a phase delay of 250 ms for all speeds. For PAFO, the FEL control decreased by 17.5% and 90.7% the NRMSE and delay, respectively. In overall, the FEL control can provide more accurate and time-effective assistance, with lower position error and phase delay than single PID feedback control. Therefore, the hybrid arrangement offered by FEL control can control limbs with a small delay and this holds great importance for gait rehabilitation.

V. CONCLUSION & FUTURE WORK

This thesis innovatively implemented a FEL control in assistive devices for gait rehabilitation. The designed ANN was able to learn the inverse dynamics of PKO and PAFO with an acceptable learning time for clinical applications. Furthermore, the learning time can be decreased by using the pre-trained ANN of AOs. Due to its learning capabilities, the feedforward control was able to handle with variations in the user-oriented trajectories. FEL control time-effectively adapts the AOs' responses to the changes in the dynamics due to the interaction with the user. Moreover, the feedback control compensated for random disturbances. FEL control substantially decreased the tracking error for acceptable ranges and the response delay compared to PID control. This thesis shows that FEL control is an adaptive and time-effective low-level controller for AOs,

such as the ones embedded in SmartOs, which may yield to an efficient gait rehabilitation.

Future work comprises the validation with more subjects, disturbances, and AAN strategies. Evaluation of the FEL control with patients is also necessary from a practical perspective. It is also relevant to compare the FEL control with a hybrid control.

REFERENCES

- [1] W. Meng, Q. Liu, Z. Zhou, Q. Ai, B. Sheng, and S. (Shane) Xie, "Recent development of mechanisms and control strategies for robot-assisted lower limb rehabilitation," *Mechatronics*, vol. 31, pp. 132–145, 2015.
- [2] M. R. Tucker *et al.*, "Control strategies for active lower extremity prosthetics and orthotics: a review.," *J. Neuroeng. Rehabil.*, vol. 12, p. 1, 2015.
- [3] J. Bergmann, C. Krewer, F. Muller, A. Koenig, and R. Riener, "Virtual Reality to control active participation in a subacute stroke patient during robot-assisted gait training," *IEEE Int Conf Rehabil Robot*, vol. 2011, p. 5975407, 2011.
- [4] M. Bortole *et al.*, "The H2 robotic exoskeleton for gait rehabilitation after stroke: early findings from a clinical study," *J. Neuroeng. Rehabil.*, vol. 12, no. 1, p. 54, 2015.
- [5] S. Viteckova, P. Kutilek, and M. Jirina, "Wearable lower limb robotics: A review," *Biocybern. Biomed. Eng.*, vol. 33, no. 2, pp. 96–105, 2013.
- [6] T. Yan, M. Cempini, C. M. Oddo, and N. Vitiello, "Review of assistive strategies in powered lower-limb orthoses and exoskeletons," *Rob. Auton. Syst.*, vol. 64, pp. 120–136, 2015.
- [7] K. Anam and A. A. Al-jumaily, "Active Exoskeleton Control Systems : State of the Art," *Int. Symp. Robot. Intell. Sensors*, pp. 988–994, 2012.
- [8] W. Lu, J. Yang, and X. Liu, "The PID controller based on the artificial neural network and the differential evolution algorithm," *J. Comput.*, vol. 7, no. 10 SPL.ISS., pp. 2368–2375, 2012.
- [9] S. Viteckova, P. Kutilek, and M. Jirina, "Wearable lower limb robotics: A review," *Biocybern. Biomed. Eng.*, vol. 33, no. 2, pp. 96–105, Jan. 2013.
- [10] F. Resquín, J. Gonzalez-vargas, J. Ibáñez, F. Brunetti, and J. L. Pons, "Feedback error learning controller for functional electrical stimulation assistance in a hybrid robotic system for reaching rehabilitation," *Eur. J. Transl. Myol.*, vol. 26, no. 3, pp. 255–261, 2016.
- [11] M. Kawato, "Feedback-error-learning neural network for supervised motor learning," *Adv. neural Comput.*, vol. 6, pp. 365–372, 1990.
- [12] K. Kurosawa, R. Futami, T. Watanabe, and N. Hoshimiya, "Joint Angle Control by FES Using a Feedback," *IEEE Trans. NEURAL Syst. Rehabil. Eng.*, vol. 13, no. 3, pp. 359–371, 2005.
- [13] "RASPBERY PI 3 MODEL B."
- [14] "Ubuntu MATE for the Raspberry Pi 2 and Raspberry Pi 3."
- [15] B. Koopman, E. H. F. van Asseldonk, and H. Van der Kooij, "Speed-dependent reference joint trajectory generation for robotic gait support," *J. Biomech.*, vol. 47, no. 6, pp. 1447–1458, 2014.
- [16] M. Bortole, "Robotic Exoskeleton With an Assist-as-Needed Control for Gait Rehabilitation After Stroke," 2014.
- [17] H. Miyamoto, M. Kawato, T. Setoyama, and R. Suzuki, "Feedback-Error-Learning Neural Network for Trajectory Control of a Robotic Manipulator," *Neural Networks*, no. 1, pp. 251–265.
- [18] A. P. E. Meireles Magali RG, "A comprehensive review for industrial applicability of ANNs.," *IEEE Trans Ind Electron*, vol. 50, no. 3, pp. 585–601, 2003.
- [19] S. Ruder, "An overview of gradient descent optimization," pp. 1–14, 2016.
- [20] D. Kingma and J. L. Ba, "ADAM: A METHOD FOR STOCHASTIC OPTIMIZATION," in *International Conference on Learning Representations (ICLR)*, 2015, pp. 1–15.
- [21] J. Sola and J. Sevilla, "Importance of input data normalization for the application of neural networks to complex industrial problems," *IEEE Trans. Nucl. Sci.*, vol. 44, no. 3 PART 3, pp. 1464–1468, 1997.

