

Article

# Intrinsic Sensing and Evolving Internal Model Control of Compact Elastic Module for a Lower Extremity Exoskeleton

Likun Wang <sup>1,2</sup>, Zhijiang Du <sup>1</sup>, Wei Dong <sup>1,\*</sup>, Yi Shen <sup>2</sup> and Guangyu Zhao <sup>3</sup>

<sup>1</sup> State Key laboratory of Robotics and System, Harbin Institute of Technology, Harbin 150080, China; likunwang@hit.edu.cn (L.W.); duzj01@hit.edu.cn (Z.D.)

<sup>2</sup> School of Astronautics, Harbin Institute of Technology, Harbin 150080, China; shen@hit.edu.cn

<sup>3</sup> Weapon Equipment Research Institute, China Ordnance Industries Group, Beijing 102202, China; guangyuzhao1980@gmail.com

\* Correspondence: dongwei@hit.edu.cn; Tel.: +86-0451-8641-8441-18

Received: 15 January 2018; Accepted: 6 March 2018; Published: 19 March 2018

**Abstract:** To achieve strength augmentation, endurance enhancement, and human assistance in a functional autonomous exoskeleton, control precision, back drivability, low output impedance, and mechanical compactness are desired. In our previous work, two elastic modules were designed for human–robot interaction sensing and compliant control, respectively. According to the intrinsic sensing properties of the elastic module, in this paper, only one compact elastic module is applied to realize both purposes. Thus, the corresponding control strategy is required and evolving internal model control is proposed to address this issue. Moreover, the input signal to the controller is derived from the deflection of the compact elastic module. The human–robot interaction is considered as the disturbance which is approximated by the output error between the exoskeleton control plant and evolving forward learning model. Finally, to verify our proposed control scheme, several experiments are conducted with our robotic exoskeleton system. The experiment shows a satisfying result and promising application feasibility.

**Keywords:** series elastic module; human–robot interaction; evolving internal model control; distributed Gaussian process; force control; exoskeleton

---

## 1. Introduction

In the past two decades, several types of prostheses, orthoses, and exoskeletons have been developed to meet the demand for human–limb assistance [1–3], interface of virtual reality [4,5], and strength augmentation [6] owing to dramatic progress in computing, sensing, as well as control. Thus, various studies have been devoted to developing novel compact mechanical structures and corresponding control methodologies.

To follow the human intention efficiently (i.e., discrete or rhythmic movements), the sensing system should provide full information for the exoskeleton controller. A feasible design of the Hybrid Assistive Limb exoskeleton [7] focuses on the electromyographical (EMG) signals. Besides, the human intention is collected by applying a pattern recognition technique. However, the main limitation is that the EMG signals suffer heavily from unwanted noise and overlap of the spectrum with other signals.

An alternative is based on the physical interaction sensor system, which aims directly onto the mapping between the human–robot interaction (HRI) and the system state variables. This interaction mainly depends on the impedance delivered from the pilot to the exoskeleton and vice versa [8,9]. Therefore, an impedance interaction model and corresponding force control methodology are desired [10,11]. Moreover, an open-loop impedance controller is designed in [12] for a specific task

without feedback derived from the force sensors. An improvement was made in [13], which takes the feedback term into consideration. Nevertheless, the control system could hardly tell apart the HRI and the system disturbance. Thus, there are still substantial uncertainties with respect to model error.

For rehabilitation therapy dealing with the interaction force and a delicate haptic interface, a more accurate impedance dynamic model is required [14]. Thus, several methods of the self-tuning impedance variables are presented in [15] to fulfill the requirement. In addition, a series elastic module has been designed to provide high-fidelity HRI. However, owing to the stochastic human–exoskeleton coupling model, the uncertainties of interaction cannot be avoided.

Apart from the identification process of the interaction model, disturbance observer is a typical technique trick for coping with the dynamic uncertainties, as detailed in [16]. Accordingly, the interaction uncertainty is compensated by the observation from the feedback loop as well as disturbance observer [17]. Nevertheless, a critical limitation of such a controller is that the control performance highly relies on the prior knowledge of the control plant. Besides, in order to obtain an approximation of the interaction torque between the exoskeleton and the user, an independent-joint-based disturbance observer in combination with gravity compensation and friction cancellation is proposed in [18]. Nevertheless, the independent joint-based implementation includes coupled multi-degrees of freedom (DoF) dynamics effects.

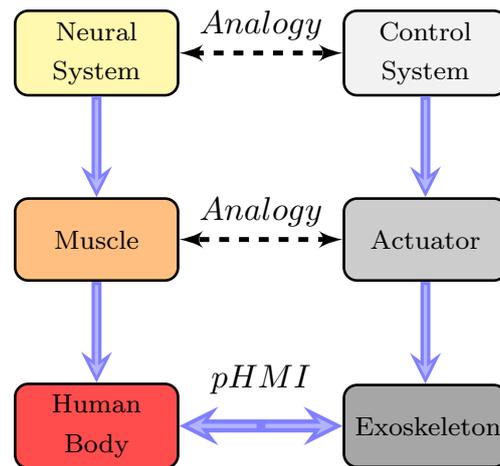
To enhance control performance and system stability in the presence of model uncertainties, Kim and Bae propose a novel time-delay robust control algorithm in [19]. Thus, derived from the time-delay design conception, the model uncertainty and the HRI disturbance are compensated according to previous observations. Thus, the desired performance can be maintained since the disturbance and the time-delay problem are solved. However, the parameters of the time-delay term should also be identified, and the corresponding identification process was not well addressed.

An innovation made in [20] is that the approach takes the human model into consideration. In existing passivity-based algorithms, the human model is simplified as a passive system. However, the control performance cannot be guaranteed, owing to the simple human coupling model. In [20], the human model is defined as a second-order dynamic system whose parameters are updated with an adaptive approach. Although the control performance is well-defined, it only works when the system can be given as a second-order environment. Another example of the force control of the series elastic actuator (SEA) takes the effect of load motion compensation into account and includes the excellent work in [21]. Moreover, a force control with acceleration feedback of series elastic actuators is introduced in [22]. Although this method exhibits higher performance accuracy and robustness and is easy to implement, the feasible application is only based on the assumption that the acceleration can be computed analytically or numerically.

In order to provide relatively low output impedance across the entire frequency spectrum, a novel cable-driven actuation for the lower extremity exoskeleton is designed in [23]. Although the performance is sufficient, the output-impedance variations can still be achieved by several closed-loop interaction control methodologies [24]. Moreover, also driven by the flexible Bowden cable transmission, two kinds of series elastic actuators (i.e., linear compression spring and helical torsion spring) are presented in [25,26] for small-scale exoskeleton application; indeed, the linear springs make the mechanical system more bulky over the finger.

Therefore, considering the above limitations, a novel compact elastic module for lower extremity exoskeleton and its corresponding control methodology—so-called evolving internal model control (EIMC)—are designed and proposed in this paper. Based on our previous work [27], the elastic module consists of two parts (i.e., the distal module for compliant control and the proximal module for HRI detection). The crucial drawback of our previous work is that the pilot feels uncomfortable with the proximal elastic module and the mechanical structure is bulky. In terms of the excellent work in [28], a novel high-power series elastic actuator has been proposed. However, the work is mainly designed for rehabilitation purposes. Moreover, the experiments are only tested on a hardware platform. The assistance performance with a human subject is not presented.

In this paper, we only apply the distal module to achieve both tasks, and the control concept is given in Figure 1. For enhancing the robustness of the control system, a distributed online learning forward model is introduced. Consequently, the HRI can be seen as the disturbance to the control system, and should be compensated. A crucial feature of our control strategy is that the model uncertainties are addressed by the online learning model and the HRI is approximated by the position error between the output of the forward model and the exoskeleton. However, in most control scenarios, the measured noise, the model uncertainties, and the HRI are all considered as the system disturbance, and cannot be distinguished.



**Figure 1.** Abstract analogy between the human and the exoskeleton. The control system can be seen as a high-level neural system of the exoskeleton. According to the high-level control signal, the actuator can drive the exoskeleton mechanical structure. Thus, to obtain an input to the controller, the human intention is interpreted from the physical human–machine interaction (pHMI) between the pilot and the exoskeleton.

To improve readability, the main contribution is presented as follows:

- (1) Based on the intrinsic sensing properties, a novel compact elastic module is designed to provide the input signal for the controller. Moreover, the compact elastic module is also utilized for the compliant actuation.
- (2) To improvement the control performance, a novel control scheme—so-called evolving internal model control—is proposed in this paper. The control scheme aims to compensate the system disturbance and model uncertainties by the difference between the exoskeleton control plant and the forward learning model.
- (3) In order to enhance the system robustness, distributed online model learning is introduced in this paper. Additionally, the main issue of the computation expense is addressed with the distributed learning framework, and the hyper-parameters for each Gaussian process are updated with the Markov Chain Monte Carlo (MCMC) algorithm.
- (4) Finally, to demonstrate our control scheme, the model learning procedure as well as the system properties are tested on our exoskeleton robotic system with several experiments.

The remainder of the paper is organized as follows:

After the introduction, Section 2 is devoted to discussing the intrinsic sensing and the mechanical design of the compact elastic module in light of the biomechanical inspiration. In Section 3, the main routine of the EIMC scheme is outlined, along with the discussion of the distributed evolving model learning. The implementation of the experiments and the comparison of the control methods are presented in Section 4. Finally, Section 5 concludes with a summary.

## 2. Compact Elastic Module

In few exoskeleton design scenarios, the mechanical structure is designed as a rigid body robotic system and the biomechanical aspect may not been well considered. Consequently, the wearer may feel uncomfortable since the natural movement is constrained. In this section, we first discuss the motion range and actuation from the point view of the biomechanical inspiration. Then, the design of a compact elastic module is introduced, and the corresponding sensing properties are carefully explained.

### 2.1. From the Biomechanical Inspiration

Human biomechanics should be taken into consideration when designing an exoskeleton robot, “since human legs have inherent damping as well as stiffness properties” [29]. In almost all daily human life, the leg joints can be simplified as damped spring and mass—especially for sitting, standing up and walking. The viscous damping is of great importance in the functioning of movement primitives. Such a property plays a significant role in shock absorbing as well as maintaining stability.

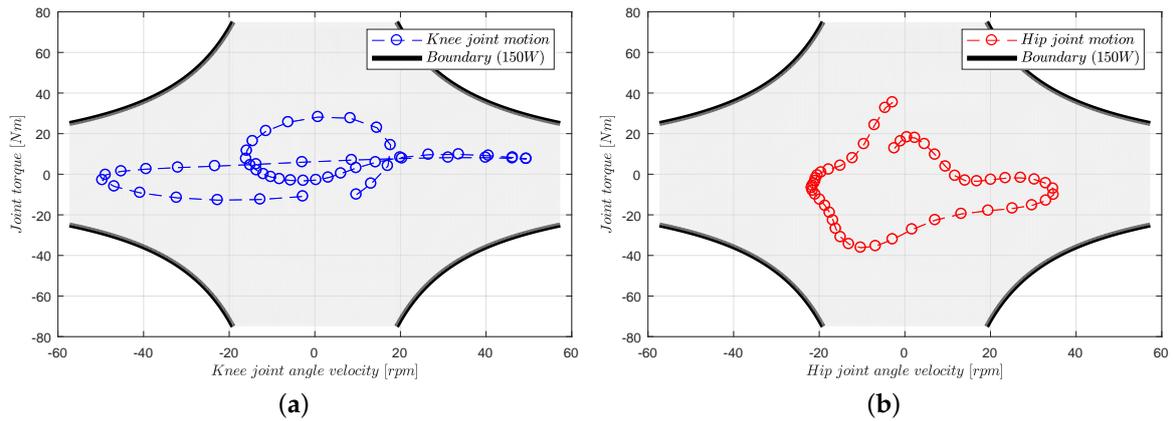
Towards the aim of forming a mechanical scheme of an exoskeleton, all the design layouts almost fall into three classical types; i.e., anthropomorphic, non-anthropomorphic, and pseudo-anthropomorphic. An anthropomorphic scheme seeks to match all the human body physical properties precisely (i.e., limb length, joint range, and misalignment). Nevertheless, in practice, scarcely a perfect anthropomorphic system can be designed regarding critical requirements ranging from exact end-effector matching as well as joint misalignment. Besides, although non-anthropomorphic design concepts are widely used in many scenarios, it is quite difficult to fulfill all the possible maneuvers (deep squats and turning corners) for ankle joints.

Therefore, we consider the above limits and design our exoskeleton based on pseudo-anthropomorphic architecture. This means that the developed mechanical scheme is kinematically similar to the human biomechanics; i.e., three DoFs at both ankle and hip joints, one DoF for the knee joint. Moreover, at the very least, the range of each joint should cover the range of human locomotion and less than the maximum range of human motion for safety concerns. Thus, the whole range of every joint is given in Table 1.

**Table 1.** Joints ranges of motion.

Joints	DoF	Scope
Hip	Flexion/extension	$-45^{\circ}$ – $90^{\circ}$
	Adduction/abduction	$-20^{\circ}$ – $35^{\circ}$
	Medial/lateral rotation	$-35^{\circ}$ – $35^{\circ}$
Knee	Flexion/extension	$0^{\circ}$ – $85^{\circ}$
Ankle	Plantarflexion/dorsiflexion	$-30^{\circ}$ – $25^{\circ}$
	Pronation/external rotation	$-25^{\circ}$ – $15^{\circ}$
	Inversion/eversion	$-15^{\circ}$ – $10^{\circ}$

To fulfill the requirement of normal human walking [30,31] assistance, the exoskeleton robotic system should provide a large range of torques, and the selection of the proper electric motors should be well considered since the maximum speed and torque are limited (presented in Figure 2). In our situation, since the maximum continuous speed and torque of the selected motor (RE40, Maxon Motor Company, Sachseln, Switzerland) [32] are 8200 rpm and 0.181 Nm, the corresponding parameters should be designed in proper scope. Thus, a spur gearhead (26:1) and a pair of bevel gears (4:1) (given in Figure 3) are applied to adjust the speed and torque to 78.8 rpm and 18.8 Nm, separately. The maximum angular speed is near 60 rpm for the strength augmentation, and the exoskeleton should be capable of tracking the human primitives in real-time. Therefore, the maximum driven torque (268 Nm) is large enough for the exoskeleton actuation plus the stall torque (249 Nm).



**Figure 2.** Provement of electric motor selection. For electric actuation, the power of the selected electric motors should cover the range of the joint torque during normal human walking. For safety concerns and efficiency, we chose the RE40 dc motor (150 W) from the Maxon Motor Company. (a) Motion range of the knee joint; (b) motion range of the hip joint.



(a)

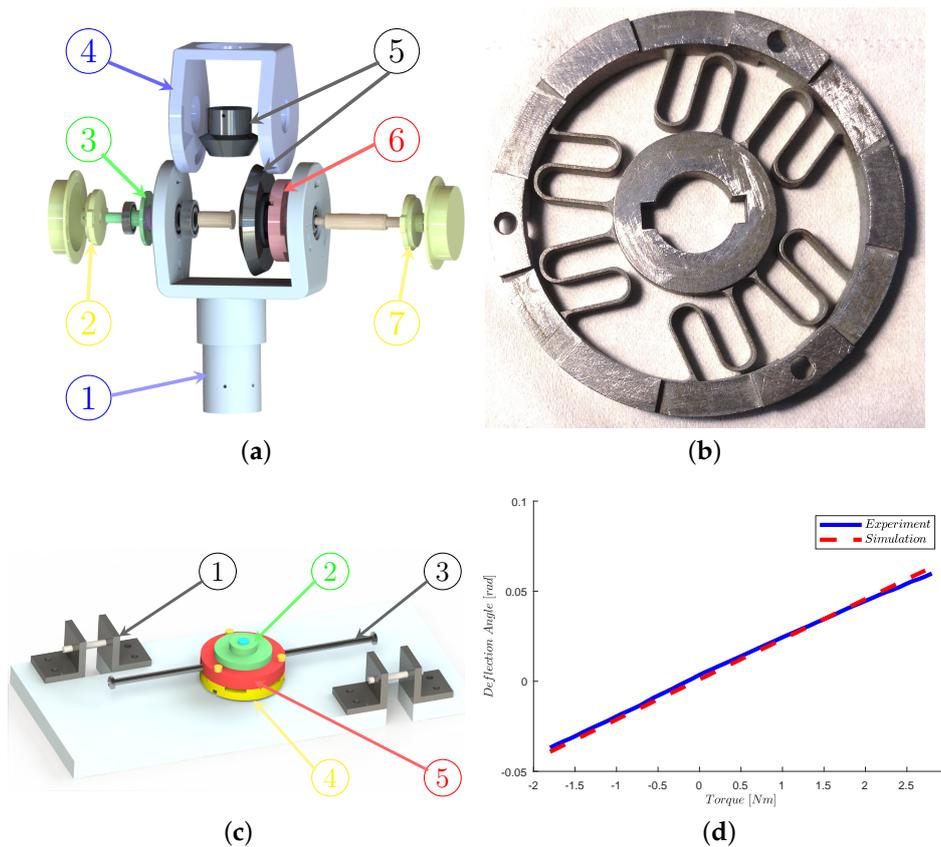


(b)

**Figure 3.** Mechanism of joint power transmission. Considering the scope of the joint torque and compactness of the mechanism, we chose the above scenario (i.e., electric motor with spur gearhead and bevel gears). (a) Electric motor RE40 and spur gearhead (26:1); (b) bevel gears (4:1).

## 2.2. Designing and Sensing of a Compact Elastic Module

In our previous work [27], two elastic modules were applied for compliant control and human intention recognition, separately. In this paper, we only use one elastic module and combine the intrinsic sensing property of the elastic module and the compliant actuation to improve the compactness of mechanical design, presented in Figure 4a. The deflection of the elastic module is measured through the right encoder, while the incremental joint position is obtained by the left encoder. Consequently, the input to the controller is interpreted as the deflection of the elastic module. Moreover, the HRI force is compensated by our proposed EIMC and will be carefully explained in the next section.



**Figure 4.** (a) Configuration of the compact elastic module; 1: Shank segment, the torsional elastic module, 2: Encoder, 3: Encoder support, 4: Thigh segment, 5: Bevel gear pair, 6: Torsional elastic module, and 7: Encoder. (b) Front view of the torsional elastic module; properties of the designed torsional elastic module are given in Table 2. (c) The calibration platform; 1: Bracket, 2: Encoder, 3: Force-bearing bar, 4: Torsional elastic module, 5: Position pin. (d) Calibration result of the elastic module; experimental data is presented in the solid blue line, while the simulation data are given in the red dashed line.

Since the estimation of human motion recognition highly depends on the accuracy and stability of the physical HRI, the elastomer is of great significance in the sensor design. The common options of topology structures are symmetrical and centrosymmetric. The main drawbacks of the symmetrical structures are higher processing costs and less optimized parameters. Based on the analysis in [33], the architecture of spiral and double helix are usually neglected. Additionally, the linearity and the stiffness cannot be guaranteed, owing to the backlash. Therefore, we design a centrosymmetric structure elastic module, as shown in Figure 4b. With the parallelogram-like deflection between the inner circle and outer circle, the human motion is measured by the deflection of the elastic module through the encoder. Moreover, the designed parameters are given in Table 2.

A calibration procedure is required for sensing in a robotic system, and a calibration platform was designed as presented in Figure 4c. The body of the elastic module is grounded on the platform, and the deflection is measured with the encoder mounted on the outer ring. Additionally, the external torque is generated by the force-bearing bar. The measurement signal is obtained from a programmable multi-axis controller (PMAC, Delta, Chatsworth, CA, USA). During the calibration experiment, the loading cell straining on the elastic module ranges from 100 g to 2.5 kg with 50 g increments. The results given in Figure 4d show that the stiffness is  $K_e = 60.2 \text{ Nm/rad}$ . Compared with simulation, the calculation error is 4.5%. Moreover, the resolution 0.1 Nm is guaranteed by the 12-bit resolution magnetic encoder.

**Table 2.** Properties of the elastic module.

Parameters	Values
stiffness	60.2 Nm/rad
diameter of outer circle	60 mm
diameter of inner circle	8 mm
maximum torsion torque	4 Nm
thickness	5 mm
maximum deflection	0.087 rad
resolution	0.1 Nm

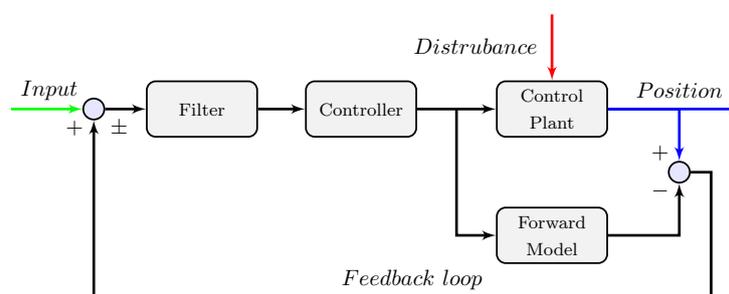
### 3. Evolving Internal Model Control

The internal model control [34,35] is one of the standard applied model-based techniques for nonlinear control systems. However, for lower extremity exoskeleton systems, the dynamic model is not easy to obtain. Although several black-box modeling methods [36,37] have been combined with internal model control for dynamic model identification, the barriers are mainly associated with the curse of dimensionality. In this section, we briefly introduce the essential properties of internal model control and carefully address the modeling issue with our proposed online learning algorithm.

#### 3.1. Internal Model Control with Gaussian Process

For realizing ideal motor control of an exoskeleton which interacts with a pilot, the joint friction, HRI, and inertia should be compensated. Compared with the internal model control, the primary limits of the classical control theory fail to explicitly model the dynamic system according to a classical feedback loop. Thus, to enhance the control performance, internal model control is employed in this paper. In [38], the forward model is learned with a full Gaussian process. The prediction of the Gaussian process provides mean as well as variance. This variance can be seen as the level of confidence of the prediction, which is a crucial advantage compared with neural network or fuzzy models. Note also that the variance is explicitly applied when the data-stream subset should be updated (remove data pair or add data pair). Moreover, owing to the Gaussian process theory, each time step prediction is approximated as a Gaussian process, which is determined by the mean and the variance.

As shown in Figure 5, the reference input to the controller is related to the desired position. The motor command generated by the controller is sent to the control plant and a forward model. Thus, compared with the output from the control plant, disturbance from the environment may cause differences between the predicted positions and the actual system states. The designed filter balances the robustness to the model uncertainties and the desired closed-loop behavior of the control system.



**Figure 5.** A general representation of the internal model control. The design concept of the internal model control is to compensate for errors or make adjustments from the desired outputs. Thus, two primary parts of the control algorithm are the forward model and feedback loop used to enhance the control performance.

Compared with the other state-of-the-art control algorithms, the crucial advantages consist of the following properties.

**Property 1** (Dual Stability Criterion). *If the model is accurate, the stability of both plant and controller is sufficient for overall system stability.*

**Property 2** (Perfect Controller). *If the controller can be achieved as the inverse of the model, the perfect controller can be obtained.*

Since the exoskeleton is a typical robotic system with high coupling and inherent nonlinearity, we apply adaptive computed torque control to address this issue. Thus, without loss of generality, the driven torque is given as:

$$\tau = \hat{H}(\theta)(\ddot{\theta}_d + K_d\dot{e} + K_p e) + \hat{C}(\theta, \dot{\theta})\dot{\theta} + \hat{G}(\theta), \quad (1)$$

where  $\hat{H}$ ,  $\hat{C}$ , and  $\hat{G}$  denote the estimations of the inertial matrix, the centrifugal and Coriolis term, and the gravitational matrix, respectively. Considering the real exoskeleton dynamic model, we have

$$H(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + G(\theta) = \hat{H}(\theta)(\ddot{\theta}_d + K_d\dot{e} + K_p e) + \hat{C}(\theta, \dot{\theta})\dot{\theta} + \hat{G}(\theta), \quad (2)$$

where  $H$ ,  $C$ , and  $G$  denote the inertial matrix, the centrifugal and Coriolis term, and the gravitational matrix, respectively. In light of the linear connection of the inertial parameters, we have

$$\hat{H}(\theta)(\ddot{e} + K_d\dot{e} + K_p e) = \tilde{H}(\theta)\ddot{\theta} + \tilde{C}(\theta, \dot{\theta})\dot{\theta} + \tilde{G}(\theta) \quad (3)$$

$$= Y(\theta, \dot{\theta}, \ddot{\theta})p, \quad (4)$$

with  $\tilde{H} \triangleq H - \hat{H}$ ,  $\tilde{C} \triangleq C - \hat{C}$ , and  $\tilde{G} \triangleq G - \hat{G}$ . Note that only  $p$  includes all the inertial parameters. To demonstrate the stability, we transform the equation into the following form:

$$\dot{x} = \begin{bmatrix} 0 & I \\ -K_p & -K_d \end{bmatrix} x + \begin{bmatrix} 0 \\ -I \end{bmatrix} \Phi p \triangleq Ax + Bp, \quad (5)$$

with  $x = [e^T, \dot{e}^T]^T$  and  $\Phi = \tilde{H}^{-1}Y$ . Therefore, the control system is stable, since both  $K_p$  and  $K_d$  are defined as positive definite matrix. Moreover,  $p$  is evolved according to  $\dot{p} = \Gamma^{-1}\Phi^T B^T R x$ , where  $\Gamma$  is also denoted as a positive definite matrix. Besides, the following Lyapunov equation should be satisfied and  $R$  is the only solution:

$$A^T R + RA = -Q. \quad (6)$$

Since the control system is stable, the first property is easily satisfied. Regarding the second property, the perfect controller which can be designed as the inverse of the dynamic model is quite difficult to obtain, owing to HRI disturbance and model uncertainties. However, if the accurate model can be achieved, the disturbance can be compensated to the next time step control input. Therefore, the precision of the robotic exoskeleton system can be enhanced.

### 3.2. Offline Distributed Learning of Forward Model

Although the Gaussian Process is widely applied in many robotics programs, the computation expense is a strong barrier to the implementation of online model learning. The distributed forward model learning proposed in this paper aims to lower the computational burden by explicitly applying several subsets of the training data. In the past five years, the sparse Gaussian process has been widely used in spatiotemporal modeling [39] as well as robotics reinforcement learning [40]. However, it is inconceivable to use sparse approximations with a data set size of  $N \approx \mathcal{O}(10^7)$  [41]. To further reduce computation expense, we employ distributed computations as an alternative method. In [41],

the application of the distributed Gaussian process for deterministic test point has been validated. In this paper, we extend for a more complicated task (i.e., the distributed Gaussian process for test point with a distribution).

“A Gaussian process is a collection of random variables, any finite number of which have a joint Gaussian distribution” [42]. Consider a regression problem  $y = f(x) + \epsilon \in \mathcal{R}$ , where  $x \in \mathcal{R}^D$ . The likelihood  $p(y|f(x)) = \mathcal{N}(f(x), \sigma_\epsilon^2)$  explains for the i.i.d. measurement noise  $\epsilon \in \mathcal{N}(0, \sigma_\epsilon^2)$ .

The states of the system are joint position  $q$  and angular velocities  $\dot{q}$ . Moreover, the control signal  $u < 40$  Nm is constrained for safety concerns and selected with a random value for training. Thus, the forward model is implemented as a Gaussian regression, where the training inputs are defined as tuples  $x_i^{(t)} = (q_t, \dot{q}_t, u_t)$  and differences  $y_i^{(t)} = \Delta q_t = q_{t+1} - q_t$  as training target, with  $i = 1, \dots, T$ . Throughout this paper, the squared exponential kernel  $k$  with automatic relevance determination is given as

$$k(x_i^{(t)}, x_j^{(t)}) = \alpha^2 \exp\left(-\frac{1}{2}(x_i^{(t)} - x_j^{(t)})^T \Lambda^{-1} (x_i^{(t)} - x_j^{(t)})\right), \quad (7)$$

where  $\theta = \{\Lambda, \alpha^2\}$  (length-scales vector  $\Lambda$ , signal variance  $\alpha^2$ ).

Under the distributed framework, we assume that the  $M$  Gaussian processes (GPs) are independent. Thus, the marginal likelihood  $p(\mathbf{y}|\mathbf{X}, \theta)$  for a full GP training set can be decomposed into the product of  $M$  individual parts

$$p(\mathbf{y}|\mathbf{X}, \theta) = \prod_{k=1}^M p_k(\mathbf{y}^{(k)}|\mathbf{X}^{(k)}, \theta), \quad (8)$$

where each term  $p_k$  is characterized by the  $k$ th GP. In addition, we share a set of hyper-parameters  $\theta$  for all the  $k$  GPs. To optimize the hyper-parameter  $\theta$ , a common method is maximized with the evidence maximization. However, in this paper, we employ more efficient method (i.e., expectation maximization with Monte Carlo sampling).

Additionally, the expectation step aims to compute the posterior distribution of the states  $q(\mathbf{X}) = p(\mathbf{X}^{(k)}|\mathbf{y}^{(k)}, \theta)$ . Since the posterior distribution is not GP and is difficult to compute analytically, we draw several samples from the posterior distribution according to Monte Carlo sampling. Consequently, the true distribution which is not a GP is approximated by a GP distribution. Thus, the marginal likelihood can be given by Monte Carlo integration. For the maximization step, the hyper-parameters are obtained from maximizing the expected likelihood  $\mathcal{E}_q[p(\mathbf{X}^{(k)}|\mathbf{y}^{(k)}, \theta)]$  with stochastic conjugate gradient. The main routine is given in Algorithm 1. Therefore, the learning procedure is implemented with 10 multiprogramming (GP C++ library [43]) under a distributed framework. Note that the data is collected with proportional derivative (PD) control on the above-described system, and each data volume contains 1000 data pairs.

As can be seen in Figure 6a, the learning procedure requires about 30–40 trials each GP. Since the evolution function is initialized with a random condition, the first step penalty may be different. Moreover, during the training, the graphic interpretation of Monte Carlo sampling is given in Figure 7. Consequently, with 20 samples, the distribution can be approximated by a GP distribution. In order to verify the distributed learning forward model, the 600 unused data pairs are tested with the model, given in Figure 6. In addition, as can be seen in the Figure 6, the hip joint model propagates the uncertain input (with system noise) to a next-step output joint position with nearly 0.01 rad in the learning region of interest.

**Algorithm 1:** Calculate Hyper-Parameters Value with Monte Carlo Expectation Maximization

---

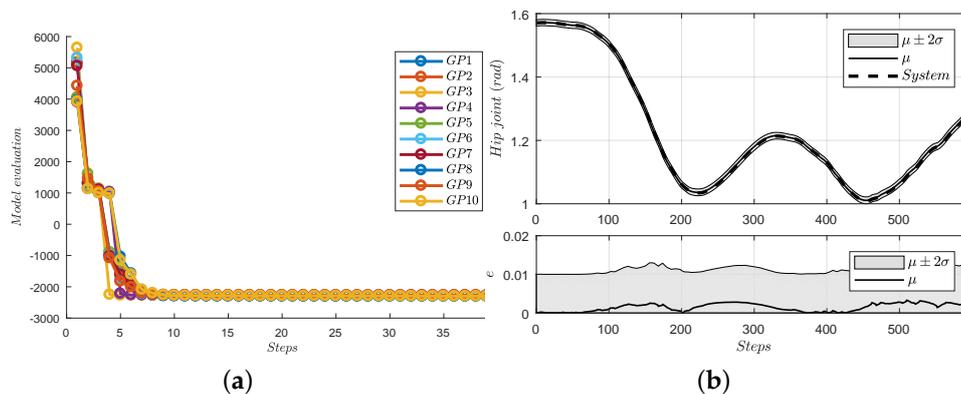
```

input : Training data  $\mathbf{X}^{(k)}, \mathbf{y}^{(k)}$ , where  $k \in 1, \dots, M$ ;
output: Hyper-parameters  $\theta$ ;

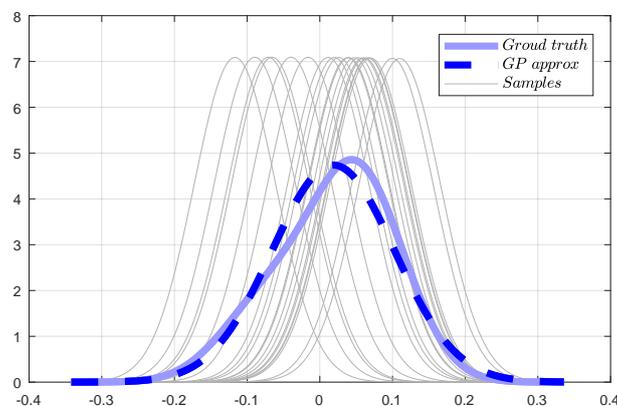
// Initialization;
 $\mathbf{\Pi} = 0, L = 20, I = 500$ ;
for  $i \leftarrow 1$  to  $I$  do
  for  $k \leftarrow 1$  to  $M$  do
    for  $j \leftarrow 1$  to  $L$  do
      // Expectation step;
       $\mathbf{X}_j^{(k)} \sim p(\mathbf{X}^{(k)} | \mathbf{y}^{(k)}, \theta)$ ;
      // The M GPs share the same hyper-parameters;
       $\mathbf{\Pi} = \mathbf{\Pi} + \frac{1}{L} \sum_{j=1}^L p(\mathbf{y}, \mathbf{X}_j^{(k)} | \theta)$ ;
    // Maximization step using stochastic conjugate gradient methods;
     $\theta \in \operatorname{argmax} \log \mathbf{\Pi}$ ;

```

---



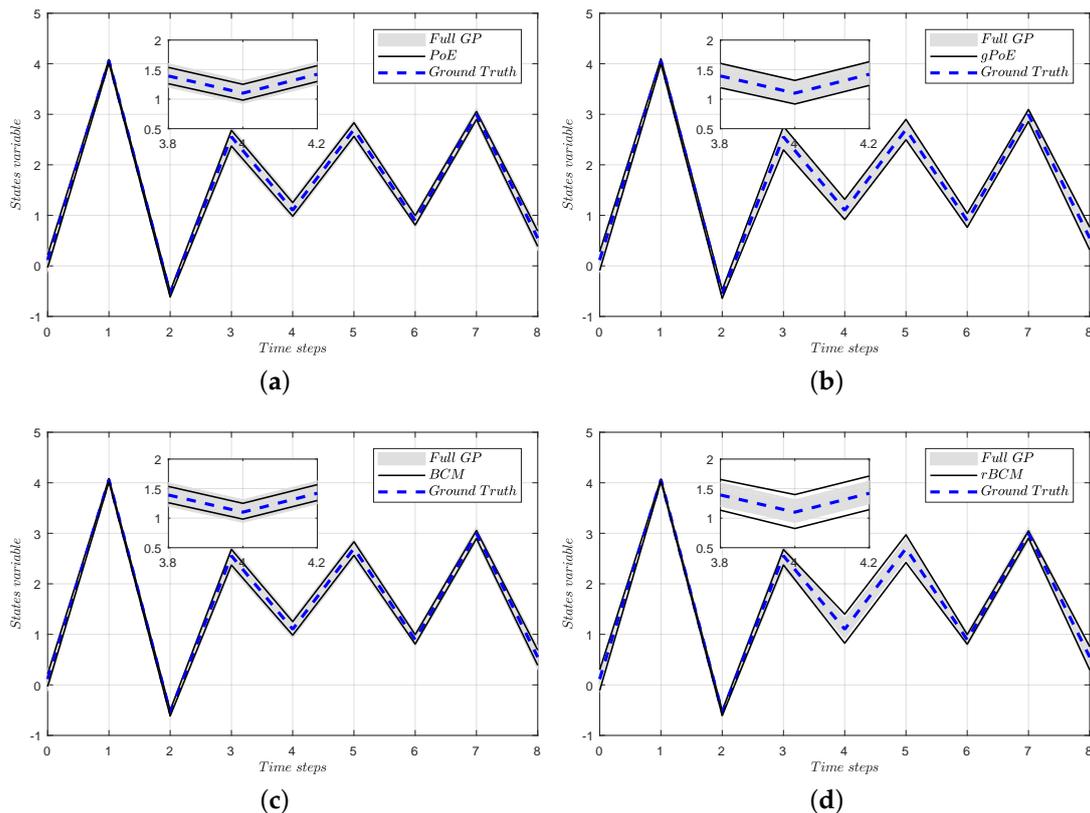
**Figure 6.** Learning procedure and cross-validation of the distributed forward model. (a) Learning procedure hyper-parameters of 10 GPs; (b) cross-validation of learned forward model. The efficient training process has been explained in the above figure. In addition, since the trust region of the trained model is about 0.57 deg, the precision for the practical experiment can be accepted.



**Figure 7.** Graph interpretation of the Monte Carlo sampling. Note that the true distribution is not a GP as given with a dashed blue line, and it is evaluated as a GP shown in the solid blue line with 20 samples.

To combine the  $M$  GPs to obtain a whole prediction, we compare four fusion algorithms; i.e., product-of-GP-experts (PoE) [44], generalized PoE (gPoE) [45], the Bayesian committee machine (BCM) [46], and the robust Bayesian committee machine (rBCM) [41]. To choose a fusion method, we tested one dimension problem [47] with the four algorithms.

As shown in Figure 8 and Table 3, compared to the full GP prediction (negative log-likelihood (NLL):  $-3.12$ ), both PoE (NLL:  $-3.22$ ) and BCM (NLL:  $-3.25$ ) overestimate the variance in the region of interest and result in overconfident precision. Although a more robust solution is desired, the predictive result of the rBCM (NLL:  $-2.54$ ) is too conservative. On the contrary, the gPoE (NLL:  $-2.81$ ) provide a more reasonable prediction mean and variance.



**Figure 8.** Comparison of four data fusion methods (product-of-GP-experts (PoE), generalized PoE (gPoE), Bayesian committee machine (BCM), robust Bayesian committee machine (rBCM)) for the distributed forward model. In each subfigure, the confidence interval of the comparison algorithm is filled between two black lines. Additionally, the full GP to be approximated is shown in the gray shaded area, representing 95% of the confidence interval, and the ground truth is given in blue dashed line. We apply NLL (negative log-likelihood) to evaluate the fusion performance presented in Table 3. (a) PoE; (b) gPoE; (c) BCM; (d) rBCM.

**Table 3.** NLL comparison of four fusion methods with full Gaussian process (GP).

Methods	Full GP	PoE	gPoE	BCM	rBCM
NLL	$-3.12$	$-3.22$	$-2.81$	$-3.25$	$-2.54$

Therefore, considering the above analysis, the gPoE is applied in a distributed inference framework as shown in Algorithm 2. Moreover, propagating through a nonlinear system, the output is not a GP model. Thus, the output distribution can be seen as a Gaussian mixture and evaluated by Monte Carlo approach.

**Algorithm 2:** Inference with Markov Chain Monte Carlo

---

```

input : Training data  $\mathcal{D} = \mathbf{X}^{(k)}, \mathbf{y}^{(k)}$ , where  $k \in 1, \dots, M$ , Hyper-parameters  $\theta$ ;
output: The inference Mean  $\mu_*$  and Variance  $\sigma_*$ ;

// Initialization;
s = 10;
for k  $\leftarrow$  1 to M do
  // Draw samples from GP distribution;
   $p(\mathbf{y}_*^{(k)} | \mu_*, \Sigma_*^{(k)}, \mathcal{D}) \approx \frac{1}{S} \sum_{i=1}^S \mathcal{N}(\mu(x_*^{(i)}), \sigma(x_*^{(i)}))$ ;
  // Fusion with GeneralizedPoE ;
   $\mu_* = \sigma_*^{-2} \sum_k \beta_k \sigma(x_*^{(k)})^{-2} \mu(x_*^{(k)})$ ;
   $\sigma_*^{-2} = \sum_k \beta_k \sigma(x_*^{(k)})$ 

```

---

## 3.3. Online Distributed Evolving of Forward Model

In this subsection, we seek to present an online model learning algorithm. Such an algorithm is desired since the training data collected under the implementation of another control scheme is not entirely precise or when the dynamics model is time-varying. In addition, although the reinforcement learning is applied to several robotic cases [48,49], leaning an exoskeleton dynamic model with human interaction is not recommended due to safety concerns.

For online learning implementation, the critical barrier is computation expense. Based on our distributed prediction scheme in Section 3.2 and inspired by the interesting work in [50], we propose a distributed online evolving model learning methodology, and its main routine is given in Algorithm 3.

**Algorithm 3:** Distributed Online Model Evolving

---

```

input : Observation  $(z_*, \mathbf{y}(z_*))$ , predictive distribution  $\mathcal{N}(\mu(z_*), \sigma^2(z_*))$ ;
output: Updated hyper-parameters  $\theta^{new}$ ;

for i  $\leftarrow$  1 to M do
  // Add new observation data for M GPs;
  if  $\sigma^2(z_*) > threshold_{\sigma^2}$  then
     $\mathcal{I}_i \leftarrow \{\mathcal{I}_i, z_*\}$ ;
  if Length( $\mathcal{I}_i$ ) > maxLength then
    for j  $\leftarrow$  1 to M do
       $\mathcal{I}_i^j \leftarrow \{\zeta_1, \dots, \zeta_{j-1}, \zeta_{j+1}, \dots, \zeta_N\}$ ;
      // Calculate the informative gain of each N - 1 data set;
       $iGains[j] \leftarrow -\ln p(\mathcal{I}_i^j, \theta)$ ;
    ind  $\leftarrow$  -1;
    min  $\leftarrow$   $\infty$ ;
    for k  $\leftarrow$  1 to Length(iGains) do
      if iGains[k] < min then
        ind  $\leftarrow$  k;
        min  $\leftarrow$  iGains[k];
     $\mathcal{I}_i^{new} \leftarrow \{\zeta_1, \dots, \zeta_{ind-1}, \zeta_{ind+1}, \dots, \zeta_N\}$ ;
     $\theta^{new} \leftarrow \text{calcHyperVal}(\mathcal{I}_i^{new}, \theta)$ 

```

---

The general concept of the distributed evolving model can be explained in the following steps: first, for  $M$  initial data sets with trained hyper-parameters, the new observations  $z_*$ , such as sensor

data derived from encoder, inertial measurement unit (IMU), drivers, and deflection of compact module are added to difference  $M$  sets  $\mathcal{I}_i$  for next step evaluation; second, calculate each  $j$  informative gain using  $iGains[j]$  data sets  $\mathcal{I}_i^j$  with data  $j$ , where  $j = 1, \dots, N$ ; according to the informative gains, find the observation with the worst information gains and remove it to form a new data set  $\mathcal{I}_i^{new}$ ; finally, the new hyper-parameter  $\mathcal{I}_i^{new}$  should be recalculated in terms of  $\mathcal{I}_i$  as well as  $\theta$ , and the updated covariance  $K$  is obtained. The procedure is repeated for every incoming loop until a terminate command is received or there is no more available data.

Not akin to the algorithm in [50], the judge condition of adding a new observation only depends on the prediction variance. To be more specific, if the prediction variance is higher than the threshold value, it can be seen that the model is not confident, even though the difference between the predictive mean and its threshold is small enough. Moreover, the hyper-parameters are calculated by the sum of the marginal likelihood of the  $M$  GPs according to Algorithm 1.

Therefore, the whole control scheme as shown in Figure 9 can be introduced as follows.

- (1) First, the desired position  $\theta_m^{desired}$  can be obtained from the deflection  $E_I$  of the elastic module. If the pilot keeps a steady pose, there is no incremental difference between the motor position and the human joint angle. Therefore, the desired position  $\theta_m^{desired}$  is zero.
- (2) Second, the input to the adaptive controller consists of the desired position  $\theta_m^{desired}$  as well as the error position compensation  $\Delta\theta_{com}$  from the feedback loop.
- (3) Then, the same torque command is sent to the distributed online model and the exoskeleton system with the signal amplification by the driver. Moreover, the  $M$  new observation pairs are evaluated with Algorithm 3 and the additional data will be added to the new  $M$  subsets if the condition is satisfied.
- (4) Finally, based on the internal model control framework, the error position  $\Delta\theta_{com}$  is compensated through the feedback loop. Besides, the low-pass filter is applied to enhance the robustness of the control system.

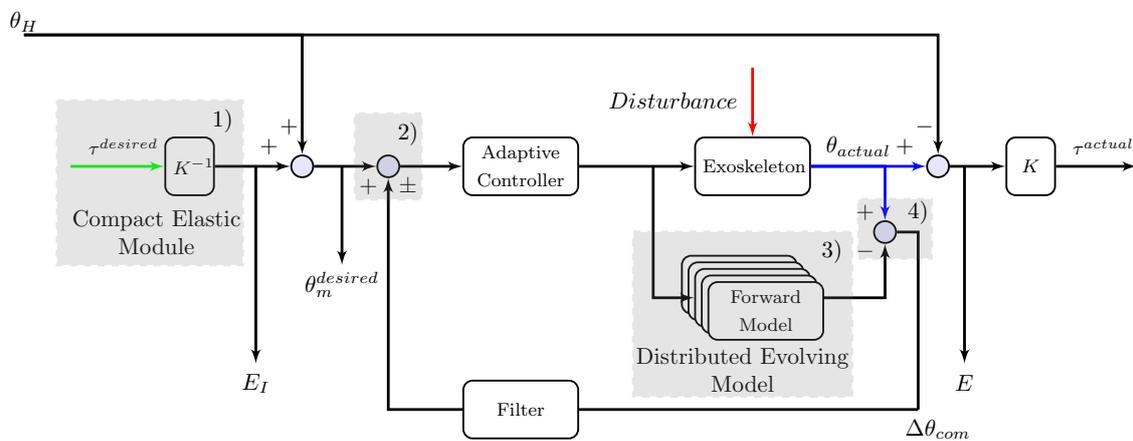


Figure 9. General representation of the evolving internal model control.

**Online Computation Expense:** The main issue of the Gaussian process online learning algorithm is the computation cost. As summarized in Table 4, if  $N$  is defined as the data size, the training and prediction of the full Gaussian process scale in  $\mathcal{O}(N^3)$  and  $\mathcal{O}(N^2)$ . This inherent weakness limits the practical application. In terms of the sparse Gaussian process, the computation burden is lowered by implicitly or explicitly using a subset of the data ( $Q \approx N/10$ ). Nevertheless, it is impossible to apply Gaussian process to a training set size of ( $N \geq \mathcal{O}(N^7)$ ). Hence, to further improve the computation efficiency, “the distributed Gaussian process can handle arbitrary large data sets” with sufficient hierarchical product-of-expert model [41], which is used in our offline model learning. Since we cannot capture all the possible scenarios, the distributed online model evolving is employed in this paper.

Thus, under the distributed framework, computing the inverse covariance of the incoming data stream for  $M$  subsets scales in  $\mathcal{O}(MP^3)$  or  $\mathcal{O}(MP^2)$  for covariance updating or only subset changing in light of [50]. Concerning the inference, our proposed online algorithm scales in  $\mathcal{O}(MP^2)$ .

**Table 4.** Comparison of computation cost.

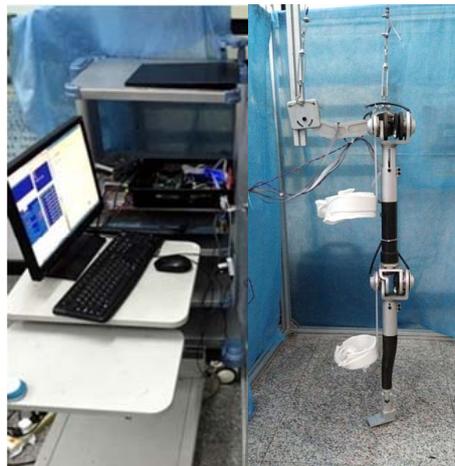
Methods	GP [42]	Sparse GP [51]	Distributed GP [41]	Proposed Method
Training	$\mathcal{O}(N^3)$	$\mathcal{O}(NQ^3)$	$\mathcal{O}(MP^3)$	$\mathcal{O}(MP^3)$ or $\mathcal{O}(MP^2)$
prediction	$\mathcal{O}(N^2)$	$\mathcal{O}(NQ^2)$	$\mathcal{O}(MP^2)$	$\mathcal{O}(MP^2)$

## 4. Experiment

Although the control system for the compact elastic module is designed and explained in the previous part, it is crucial to verify the performance of several experiments. The capabilities of the control system (e.g., the control precision, disturbance rejection, and minimal impedance and back drivability with the human subject) will be demonstrated in the following.

### 4.1. Hardware Configuration

The following experiments are verified on our single-leg exoskeleton robotic platform, as shown in Figure 10. The platform is designed as an ergonomic system for strength augmentation as well as endurance enhancement. The motion range for each joint has been carefully explained in Section 2, along with the mechanical structure of actuation with the compact elastic module.



**Figure 10.** Exoskeleton robotics system. The whole control system consists of the embedded PC, programmable multi-axis controller (PMAC), Copley actuators, power supplement, and required auxiliary facilities integrated into the control enclosure.

For our situation, three kinds of voltage (i.e., 5 V, 12 V, and 24 V) are desired. The overall control scheme is written in embedded PC, and the driven commands are sent to the electric motors through Copley driver. Moreover, the PMAC (programmable multi-axis controller, Delta, Chatsworth, CA, USA) is utilized for motion planning and signal collection.

### 4.2. Target Tasks

This experiment specifically targets three robot-aided tasks.

#### 4.2.1. System Properties

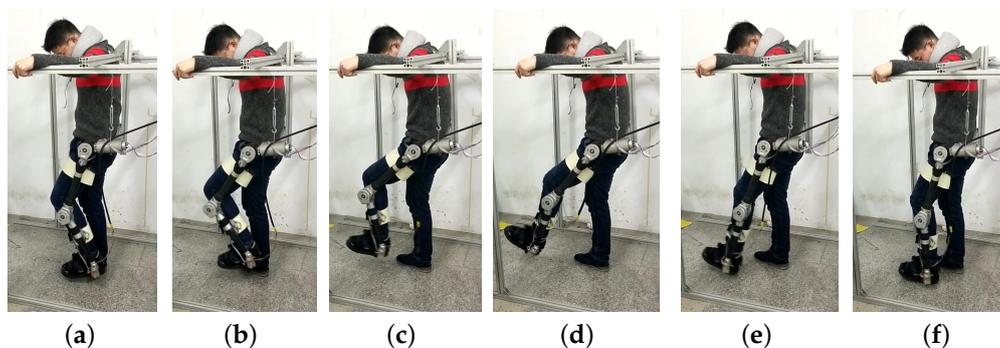
Since the additive part of the input to the controller is derived from the feedback control loop, a low-pass filter is desired to enhance the system robustness. Therefore, the frequency scope of the designed filter should at least cover the range of bandwidth of the human motion. Moreover, the filter should not be sensitive to the system noise in the high-frequency domain. Consequently, the candidates of our filter have the following form:  $F(s) = \frac{\sum_{k=0}^M (\tau s)^k}{(\tau s + 1)^N}$  [52,53]. In our case, the  $N$  and  $M$  are set to be 3 and 1. Moreover, the uncertainties  $\Delta$  are the difference of the exoskeleton plant between stance and swing phase [17].

The system bandwidth is a crucial parameter to evaluate the control performance of the system, since the frequency components of the human motion are up to 4–8 Hz, as detailed in [54]. To respond to the control signal quickly, the bandwidth should at least cover the frequency components of the human motion. In this task, the experimental preparation is the same as detailed previously. Both PD algorithm and our proposed method were tested on our exoskeleton system without a human subject.

To evaluate the control precision, we conducted a simple experiment with specified position signals on the same exoskeleton system. The first-period signal was the zero pose command; the knee joint of the exoskeleton was asked to keep a steady position to evaluate the control performance with existing static friction force and bias force. The second period was to control the exoskeleton with sinu-like position command to test the control performance against the model uncertainties. The compared algorithm was also a classic PD control scheme.

#### 4.2.2. Algorithm Comparison with Human Subject

To verify the output impedance and back drivability of the control system with the compact elastic module, the sway experiment with a human subject (subject A) is presented in this task. The human subject was asked to sway his leg naturally in the experiment, as shown in Figure 11. Note that the HRI was made possible with the compliant cuff.



**Figure 11.** Experiment with the human subject (subject A) during the swing phase of normal locomotion. (a) Heel rise; (b) initial swing ; (c) middle swing; (d) terminal swing; (e) heel strike; (f) full ground contact. With the compliant interaction between the user and exoskeleton, the human subject was asked to sway his left leg naturally. The human intention is recognized by the deflection of the elastic modules mounted on both hip and knee joints.

The compared state-of-the-art algorithms are derived from [17,33]. The reason for choosing these two algorithms for comparison is because the experimental hardware is also a single-leg exoskeleton system with series elastic actuators and the algorithms are also based on the compensation of the human interaction, which is considered as the disturbance of the control system. However, the core idea of such algorithms is disturbance observer. Since the control performance for the exoskeleton robot is difficult to verify, we follow the same evaluation tool (i.e., root-mean-square error (RMSE)) used in [17,33] to test the effectiveness of our proposed method.

#### 4.2.3. Experiment with Different Individuals

Although we aim to provide a control scheme for any human subject since the forward model is first learned offline and designed to be evolved online, the control performance should be verified with different human subjects. Thus, the human subjects' information is briefly given in Table 5. Similar to the comparison task, the human subjects were asked to naturally to sway their leg, and the system hardware was also the same as detailed above.

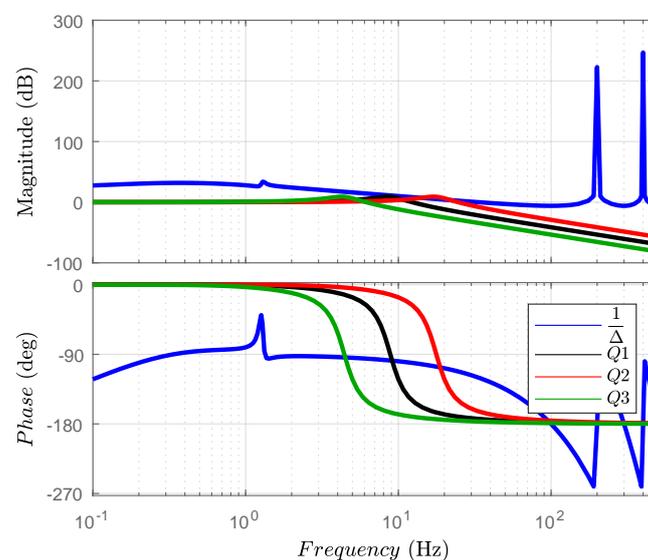
**Table 5.** Human subjects.

Subject	Gender	Age (Years)	Mass (kg)	Height (m)	Status
A	M	25	80	1.83	Healthy
B	M	28	90	1.75	Healthy
C	M	25	70	1.80	Healthy
D	M	26	90	1.80	Healthy

### 4.3. Experimental Results and Discussion

#### 4.3.1. System Properties Results

In terms of the enhancement of the closed-loop robustness, the filter was designed to eliminate high-frequency disturbance. The main issue for such a filter is the balance between the robustness against the disturbance and the disturbance rejection. If the frequency range of the filter is too extensive, the system noise poses a threat to the stability. If the frequency range of the filter is too narrow, the disturbance cannot be well compensated. The three candidate filters with the time constants 0.15, 0.075, and 0.030 are given in Figure 12. Since the magnitude response of the filter should remain below and closed to the uncertainty  $1/\Delta$ . Hence, the filter with the time constant 0.15 (in solid black line) was chosen.

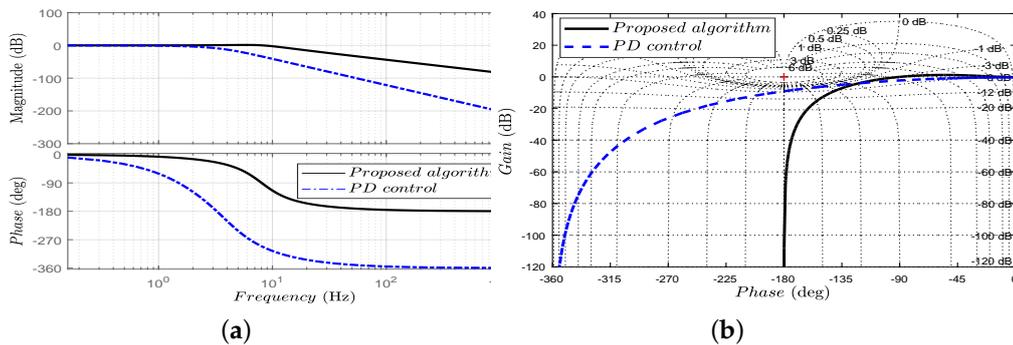


**Figure 12.** The design of the feedback-loop filter. Since the filter is employed to enhance the closed-loop robustness, the frequency scope of the designed filter should cover the range of bandwidth of the human motion. Moreover, the filter is supposed to remain below and closed to the uncertainties  $1/\Delta$ .

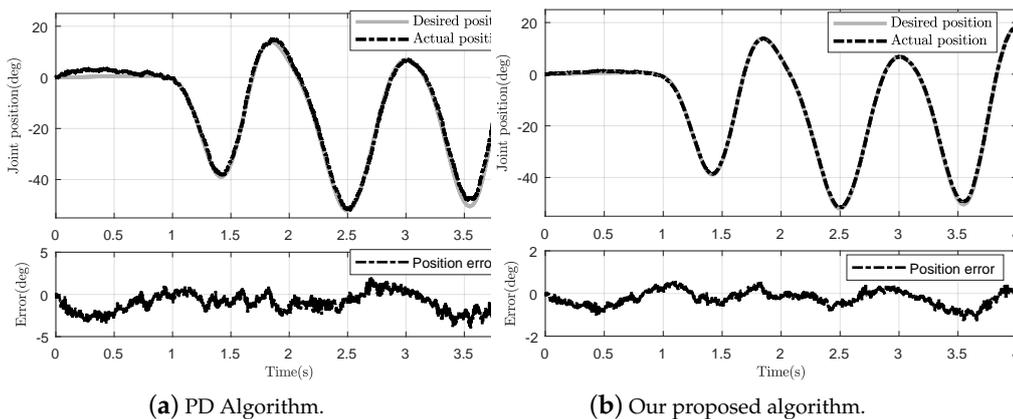
The experimental results as presented in Figure 13a show that the system can generate the desired input up to nearly 11 Hz, which for actuation with the compact elastic module is sufficient. However, the bandwidth of the PD controller is only 4 Hz, and the control performance may be

“conservative”. The rest information of our algorithm given in the Nichols Chart in Figure 13b is the phase margin (89.99 deg), gain margin (100 dB), and peak magnification (1.2 dB). Therefore, the open-loop control system is stable.

In order to test the control precision, the experiment implemented in Figure 14 compares the PD control method with our proposed algorithm. Since the static friction effect and bias force exist all the time, they pose a threat to the control precision. As shown in Figure 14a, the PD controller is strongly influenced by the friction effect or the measurement noise—even if the input to the controller is zero position. However, the same situation is better with our proposed algorithm, owing to disturbance compensation and system error adjustment. Moreover, a tracking error occurs almost at time 3.5 s with 4° in Figure 14a, while at the same high jerk the error is less than 2° in Figure 14b. Consequently, the RSME of the PD control is 5.78%, while the RMSE of the proposed control scheme is 2.07%.



**Figure 13.** The Bode diagram and the Nichols chart of the control system. (a) Frequency response for different control scheme; (b) nichols chart for different control schemes. For verifying the open-loop stability, the Bode diagram and Nichols chart are presented with the PD control algorithm and our proposed algorithm. Although both algorithms are stable in the open-loop, the bandwidth of our proposed algorithm is more extensive and hence responds more quickly. In addition, the control system with the PD controller may lose information of the normal high-frequency motion.



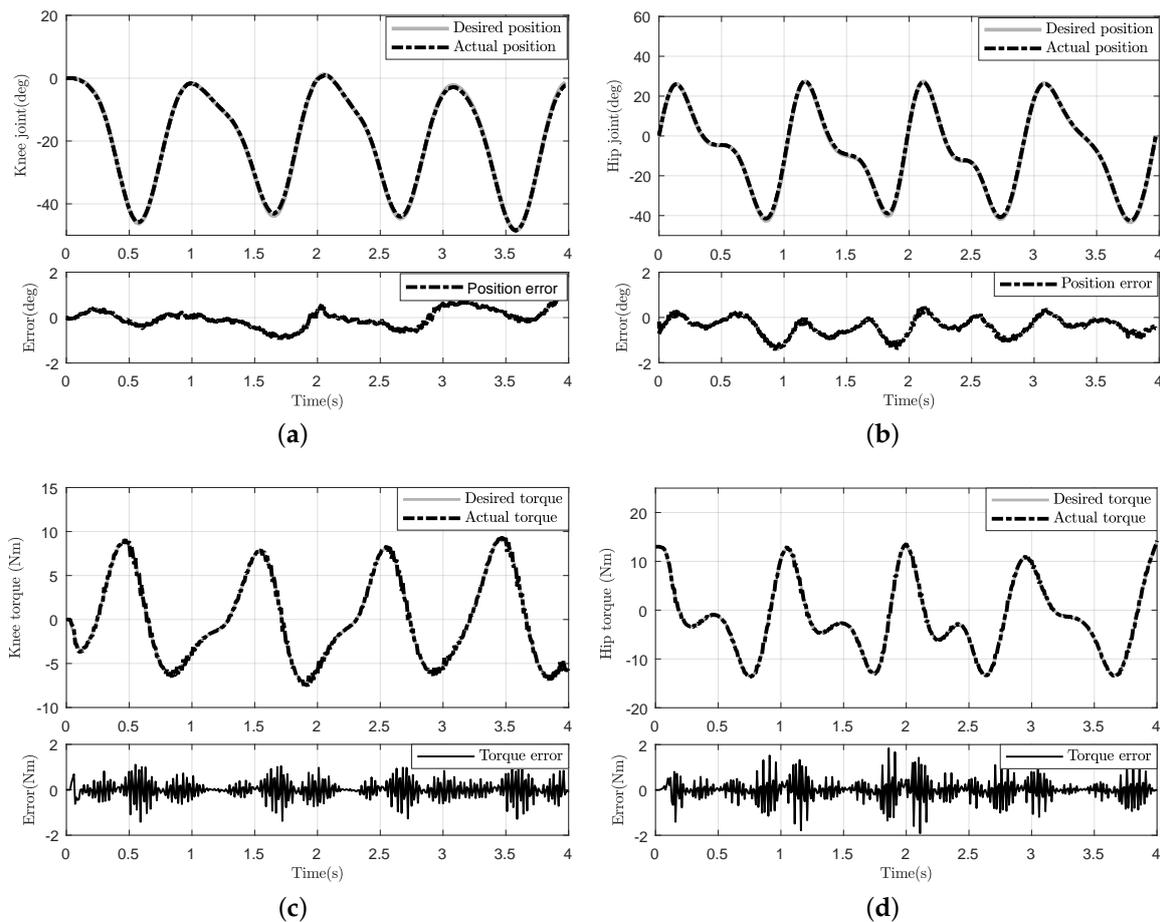
**Figure 14.** Comparison of the PD control method and our proposed algorithm with the same planning trajectory without human factor.

4.3.2. Results of the Algorithm Comparison with Human Subject

Since human locomotion is not always the same, the trajectory given in Figure 15 may be slightly different. Additionally, since our task is to follow the human motion in real-time, the exoskeleton tries to track the joint motion according to the deflection of the elastic module measured by the encoder. Therefore, in both hip and knee trajectories, the error occurs mostly in the high jerk, as shown in

Figure 15a,b. To quantitatively verify the control precision, the RMSE was computed for hip joint 3.26% and knee joint 2.89%, respectively.

Moreover, in order to actuate the exoskeleton effectively and interpret the human intention precisely, the resistive torque should be minimized. Consequently, the human subject will feel more comfortable and natural during the experiment. Thus, the desired torque and actual torque of both joints are given in Figure 15c,d. Additionally, the RMSE for hip joint and knee joint were 4.36% and 4.73%, respectively. Additionally, compared with the other algorithms, the RMSEs of the torque error of the knee joint were 5.5% [17] and 9% [33], as given in Table 6. Thus, our control scheme can provide more precise driven torque and may be more suitable for an exoskeleton application.



**Figure 15.** Experiment with a human subject. (a) Knee joint tracking and corresponding tracking error; (b) hip joint tracking and corresponding tracking error; (c) desired torque and actual torque of knee joint; (d) desired torque and actual torque of hip joint.

**Table 6.** RSME comparison of three algorithms.

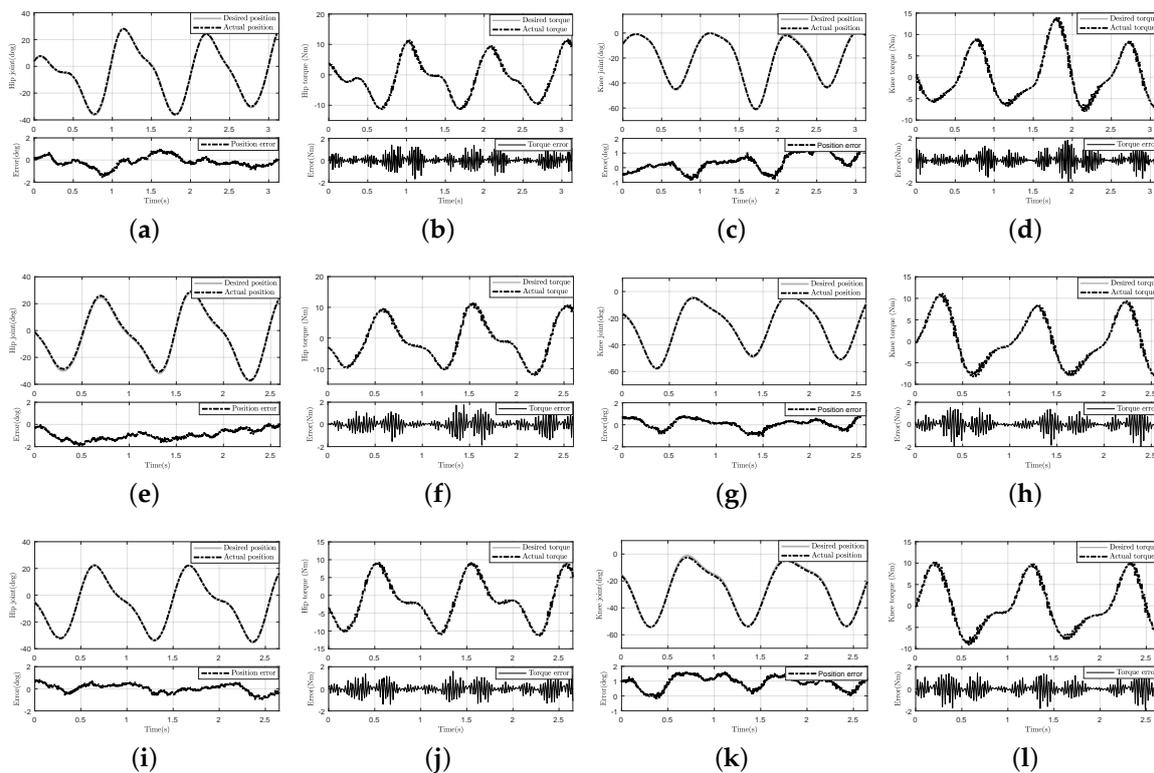
RSME	Control with Disturbance Observer [17]	Force Control for Compact Rotary Series Elastic Actuator [33]	Proposed Control
Hip	Not mentioned	Not mentioned	4.36%
Knee	9%	5.5%	4.73%

#### 4.3.3. Experimental Results with Different Individuals

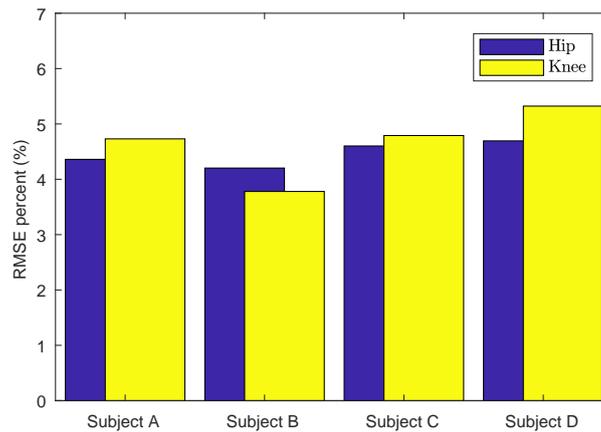
The experimental results of different individuals are reported in Figures 16 and 17. Since the forward model is learned for the exoskeleton dynamic model and evolved online, the control scheme

should be efficient for any human subject. Thus, we added three volunteers' (subjects B, C, and D) experimental results in Figure 16. Comparing the eight figures of the different joint motion of different individuals (i.e., Figure 16a,c,e,g,i,k and Figure 15a,b), although the exoskeleton tracks various human trajectories derived from several human subjects, the control precision can be guaranteed.

Moreover, to verify the online efficiency of the proposed algorithm, the experimental results of the torque tracking performance are given in Figure 16b,d,f,h,j,k and Figure 15c,d. Since the human subjects' trajectories are different, the driven torque for hip and knee joints is not the same. Consequently, to demonstrate the control performance of rejecting the human disturbance for four human subjects, the RMSE comparison of torque error is presented in Figure 17. The maximum of the RMSE percentage was less than 5.32 for the knee joint, while the maximum of the RMSE percentage was 4.69.



**Figure 16.** Experiment with the other three human subjects during the swing phase of normal locomotion. The results of three human subjects (B, C, D) are presented in order from the first row to the third row. (a) Hip joint tracking and corresponding tracking error of human subject B; (b) desired torque and actual torque of hip joint of human subject B; (c) knee joint tracking and corresponding tracking error of human subject B; (d) desired torque and actual torque of knee joint of human subject B; (e) hip joint tracking and corresponding tracking error of human subject C; (f) desired torque and actual torque of hip joint of human subject C; (g) knee joint tracking and corresponding tracking error of human subject C; (h) desired torque and actual torque of knee joint of human subject C; (i) hip joint tracking and corresponding tracking error of human subject D; (j) desired torque and actual torque of hip joint of human subject D; (k) knee joint tracking and corresponding tracking error of human subject D; (l) desired torque and actual torque of knee joint of human subject D.



**Figure 17.** RMSE comparison of torque error for four human subjects.

## 5. Conclusions

In applications related to exoskeletons driven with physical HRI, the precision of the physical sensing is still a substantial barrier to enhancing the control performance. In this paper, an elastic module is proposed to improve the compactness of the mechanical structure. Moreover, with its intrinsic sensing property, a corresponding control scheme—so-called evolving internal model control—is introduced. This scheme aims to improve the control precision with compensation from the difference between the exoskeleton and distributed forward model. Additionally, in order to adjust the forward model in real-time, a distributed online evolving model is presented to address this issue and lower the computation expense. From the experimental results, our proposed control scheme can provide more extensive bandwidth and less tracking errors compared with the PD control algorithm. Moreover, the RMSE value of torque errors also indicates a lower output impedance. Thus, the proposed compact mechanical structure and its control algorithm may provide a feasible solution for human–machine interaction applications. Future work will focus on more complicated specific tasks, such as running, left weighting, and whole body assistance.

**Acknowledgments:** Part of this work received funding from the National Science Foundation of China under Grant No. 51521003. We gratefully acknowledge the constructive comments and suggestions of the reviewers.

**Author Contributions:** Likun Wang and Yi Shen conceived and designed the experiments; Likun Wang, Wei Dong and Zhijiang Du performed the experiments; Likun Wang analyzed the data; Guangyu Zhao contributed experiment platform and devices; Likun Wang and Wei Dong wrote the paper.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

- Schabowsky, C.N.; Godfrey, S.B.; Holley, R.J.; Lum, P.S. Development and pilot testing of HEXORR: Hand EXOskeleton rehabilitation robot. *J. Neuroeng. Rehabil.* **2010**, *7*, 36, doi:10.1186/1743-0003-7-36.
- Esquenazi, A.; Talaty, M.; Packel, A.; Saulino, M. The ReWalk powered exoskeleton to restore ambulatory function to individuals with thoracic-level motor-complete spinal cord injury. *Am. J. Phys. Med. Rehabil.* **2012**, *91*, 911–921.
- Kolakowsky-Hayner, S.A.; Crew, J.; Moran, S.; Shah, A. Safety and feasibility of using the Ekso™ bionic exoskeleton to aid ambulation after spinal cord injury. *J. Spine* **2013**, *4*, 003, doi:10.4172/2165-7939.S4-003.
- Bouzit, M.; Burdea, G.; Popescu, G.; Boian, R. The Rutgers Master II-new design force-feedback glove. *IEEE/ASME Trans. Mechatron.* **2002**, *7*, 256–263.
- Nikolakis, G.; Tzouvaras, D.; Moustakidis, S.; Strintzis, M.G. Cybergrasp and phantom integration: Enhanced haptic access for visually impaired users. In Proceedings of the 9th Conference Speech and Computer, St. Petersburg, Russia, 20–22 September 2004.

6. Kazerooni, H. Exoskeletons for human power augmentation. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2005), Edmonton, AB, Canada, 2–6 August 2005; pp. 3459–3464.
7. Cruciger, O.; Schildhauer, T.A.; Meindl, R.C.; Tegenthoff, M.; Schwenkreis, P.; Citak, M.; Aach, M. Impact of locomotion training with a neurologic controlled hybrid assistive limb (HAL) exoskeleton on neuropathic pain and health related quality of life (HRQoL) in chronic SCI: A case study. *Disabil. Rehabil. Assist. Technol.* **2016**, *11*, 529–534.
8. Kiguchi, K.; Hayashi, Y. An EMG-based control for an upper-limb power-assist exoskeleton robot. *IEEE Trans. Syst. Man Cybern. Part B* **2012**, *42*, 1064–1071.
9. Wang, S.; Wang, L.; Meijneke, C.; Van Asseldonk, E.; Hoellinger, T.; Cheron, G.; Ivanenko, Y.; La Scaleia, V.; Sylos-Labini, F.; Molinari, M.; et al. Design and Control of the MINDWALKER Exoskeleton. *IEEE Trans. Neural Syst. Rehabil. Eng.* **2015**, *23*, 277–286.
10. Jimenez-Fabian, R.; Verlinden, O. Review of control algorithms for robotic ankle systems in lower-limb orthoses, prostheses, and exoskeletons. *Med. Eng. Phys.* **2012**, *34*, 397–408.
11. Bouteraa, Y.; Abdallah, I.B. Exoskeleton robots for upper-limb rehabilitation. In Proceedings of the 13th International Multi-Conference on Systems, Signals & Devices (SSD), Leipzig, Germany, 21–24 March 2016; pp. 1–6.
12. Tsukahara, A.; Hasegawa, Y.; Sankai, Y. Gait support for complete spinal cord injury patient by synchronized leg-swing with HAL. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, San Francisco, CA, USA, 25–30 September 2011; pp. 1737–1742.
13. Carignan, C.R.; Cleary, K.R. Closed-loop force control for haptic simulation of virtual environments. *Electron. J. Haptics Res.* **2000**, *1*, 2.
14. Bae, J.; Kong, K.; Tomizuka, M. Gait phase-based control for a rotary series elastic actuator assisting the knee joint. *J. Med. Devices* **2011**, *5*, 31010, doi:10.1115/1.4004793.
15. Bae, J.; Tomizuka, M. A gait rehabilitation strategy inspired by an iterative learning algorithm. *Mechatronics* **2012**, *22*, 213–221.
16. Vallery, H.; Ekkelenkamp, R.; Van Der Kooij, H.; Buss, M. Passive and accurate torque control of series elastic actuators. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2007), San Diego, CA, USA, 29 October–2 November 2007; pp. 3534–3538.
17. Kong, K.; Bae, J.; Tomizuka, M. Control of rotary series elastic actuator for ideal force-mode actuation in human–robot interaction applications. *IEEE/ASME Trans. Mechatron.* **2009**, *14*, 105–118.
18. Ugurlu, B.; Nishimura, M.; Hyodo, K.; Kawanishi, M.; Narikiyo, T. Proof of concept for robot-aided upper limb rehabilitation using disturbance observers. *IEEE Trans. Hum. Mach. Syst.* **2015**, *45*, 110–118.
19. Kim, S.; Bae, J. Force-mode control of rotary series elastic actuators in a lower extremity exoskeleton using model-inverse time delay control (MiTDC). *IEEE/ASME Trans. Mechatron.* **2017**, *22*, 1392–1400.
20. Calanca, A.; Fiorini, P. Human-adaptive control of series elastic actuators. *Robotica* **2014**, *32*, 1301–1316.
21. Calanca, A.; Muradore, R.; Fiorini, P. Impedance control of series elastic actuators: Passivity and acceleration-based control. *Mechatronics* **2017**, *47*, 37–48.
22. Calanca, A.; Fiorini, P. A Rationale for Acceleration Feedback in Force Control of Series Elastic Actuators. *IEEE Trans. Robot.* **2018**, *34*, 48–61.
23. Veneman, J.; Ekkelenkamp, R.; Kruidhof, R.; Van Der Helm, F.; van der Kooij, H. Design of a series elastic-and Bowden cable-based actuation system for use as torque-actuator in exoskeleton-type training. In Proceedings of the 9th International Conference on Rehabilitation Robotics (ICORR 2005), Chicago, IL, USA, 28 June–1 July 2005; pp. 496–499.
24. Zinn, M.; Roth, B.; Khatib, O.; Salisbury, J.K. A new actuation approach for human friendly robot design. *Int. J. Robot. Res.* **2004**, *23*, 379–398.
25. Agarwal, P.; Fox, J.; Yun, Y.; O'Malley, M.K.; Deshpande, A.D. An index finger exoskeleton with series elastic actuation for rehabilitation: Design, control and performance characterization. *Int. J. Robot. Res.* **2015**, *34*, 1747–1772.
26. Agarwal, P.; Deshpande, A.D. Series Elastic Actuators for Small-Scale Robotic Applications. *J. Mech. Robot.* **2017**, *9*, 31016, doi: 10.1115/1.4035987.
27. Long, Y.; Du, Z.J.; Chen, C.F.; Wang, W.D.; Dong, W. Development of a lower extremity wearable exoskeleton with double compact elastic module: preliminary experiments. *Mech. Sci.* **2017**, *8*, 249–258.

28. Accoto, D.; Carpino, G.; Sergi, F.; Tagliamonte, N.L.; Zollo, L.; Guglielmelli, E. Design and characterization of a novel high-power series elastic actuator for a lower limb robotic orthosis. *Int. J. Adv. Robot. Syst.* **2013**, *10*, 359, doi:10.5772/56927.
29. Zhang, L.; Xu, D.; Makhous, M.; Lin, F. Stiffness and viscous damping of the human leg. In Proceedings of the 24th Annual Meeting of the American Society of Biomechanics, Chicago, IL, USA, 19–22 July 2000.
30. Kirtley, C. CGA Normative Gait Database. 2006. Available online: <http://guardian.curtin.edu.au/cga/data/> (accessed on 7 March 2018).
31. Winter, A. Biomechanical Data Resources, Gait Data. International Society of Biomechanics. 2006. Available online: <http://www.isbweb.org/data/> (accessed on 7 March 2018)
32. Motor Maxon. Maxon Motor Company. 2015. Available online: <https://www.maxonmotor.com> (accessed on 7 March 2018).
33. Kong, K.; Bae, J.; Tomizuka, M. A compact rotary series elastic actuator for human assistive systems. *IEEE/ASME Trans. Mechatron.* **2012**, *17*, 288–297.
34. Garcia, C.E.; Morari, M. Internal model control. A unifying review and some new results. *Ind. Eng. Chem. Process Des. Dev.* **1982**, *21*, 308–323.
35. Rivera, D.E.; Morari, M.; Skogestad, S. Internal model control: PID controller design. *Ind. Eng. Chem. Process Des. Dev.* **1986**, *25*, 252–265.
36. Rivals, I.; Personnaz, L. Nonlinear internal model control using neural networks: Application to processes with delay and design issues. *IEEE Trans. Neural Netw.* **2000**, *11*, 80–90.
37. Li, S.; Gu, H. Fuzzy Adaptive Internal Model Control Schemes for PMSM Speed-Regulation System. *IEEE Trans. Ind. Inform.* **2012**, *8*, 767–779.
38. Gregorcic, G.; Lightbody, G. Internal model control based on a Gaussian process prior model. In Proceedings of the American Control Conference, Denver, CO, USA, 4–6 June 2003; Volume 6, pp. 4981–4986.
39. Luttinen, J.; Ilin, A. Efficient Gaussian process inference for short-scale spatio-temporal modeling. In Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics, La Palma, Spain, 21–23 April 2012; pp. 741–750.
40. Long, Y.; Du, Z.j.; Dong, W.; Wang, W.d. Human Gait Trajectory Learning Using Online Gaussian Process for Assistive Lower Limb Exoskeleton. In *Wearable Sensors and Robots*; Springer: Singapore, 2017; pp. 165–179.
41. Deisenroth, M.P.; Ng, J.W. Distributed gaussian processes. In Proceedings of the 32nd International Conference on Machine Learning, Lille, France, 6–11 July 2015.
42. Rasmussen, C.E.; Williams, C.K. *Gaussian Processes for Machine Learning*; MIT Press: Cambridge, MA, USA, 2006; Volume 1.
43. Blum, M. Gaussian Process Library for Machine Learning. 11th November 2012 version 0.1.4. Available online: <https://github.com/mblum/libgp> (accessed on 7 March 2018).
44. Ng, J.W.; Deisenroth, M.P. Hierarchical mixture-of-experts model for large-scale Gaussian process regression. *arXiv* **2014**, *arXiv:1412.3078*.
45. Cao, Y.; Fleet, D.J. Generalized product of experts for automatic and principled fusion of Gaussian process predictions. *arXiv* **2014**, *arXiv:1410.7827*.
46. Tresp, V. A Bayesian committee machine. *Neural Comput.* **2000**, *12*, 2719–2741.
47. Kitagawa, G. Monte Carlo filter and smoother for non-Gaussian nonlinear state space models. *J. Comput. Graph. Stat.* **1996**, *5*, 1–25.
48. Cui, Y.; Matsubara, T.; Sugimoto, K. Pneumatic artificial muscle-driven robot control using local update reinforcement learning. *Adv. Robot.* **2017**, *31*, 397–412.
49. Muelling, K.; Boularias, A.; Mohler, B.; Peters, J. Learning strategies in table tennis using inverse reinforcement learning. *Biol. Cybern.* **2014**, *108*, 603–619.
50. Petelin, D.; Kocijan, J. Control system with evolving Gaussian process models. In Proceedings of the IEEE Workshop on Evolving and Adaptive Intelligent Systems (EAIS), Paris, France, 11–15 April 2011; pp. 178–184.
51. Snelson, E.; Ghahramani, Z. Sparse Gaussian processes using pseudo-inputs. In Proceedings of the 18th International Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 5–8 December 2005; pp. 1257–1264.

52. Chen, W.H. Disturbance observer based control for nonlinear systems. *IEEE/ASME Trans. Mechatron.* **2004**, *9*, 706–710.
53. Chen, W.H.; Ballance, D.J.; Gawthrop, P.J.; O'Reilly, J. A nonlinear disturbance observer for robotic manipulators. *IEEE Trans. Ind. Electron.* **2000**, *47*, 932–938.
54. Winter, D.A. *Biomechanics and Motor Control of Human Movement*; John Wiley & Sons: Hoboken, NJ, USA, 2009.



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).