Integration of Instance-based Learning and Computed Torque Control for an Effective Assist-as-Needed Support in Human-Exoskeleton Interaction

Pablo Delgado[1*]
Yimesker Yihun[1]
Lisa Vangsness[2]
Hongsheng He[1]
[1]*College of Engineering, Wichita State University, USA*
[2]*Fairmount College of Liberal Arts and Science, Wichita State University, USA*
padelgado@shockers.wichita.edu

Abstract

The physiological responses that arise from human-robot interaction may vary across subjects in magnitude and rate. Such individual variations may require instance-based learning over model-based learning algorithms. For instance, a wearable assist-as-needed exoskeleton may require real-time progress data to provide the appropriate level of support to a specific user. In this study, an instance-based learning algorithm was developed and integrated with a computed torque control law. Real-time bio-signals, in the form of electromyography (EMG), were tracked during a predetermined time window to quantify an adaptive threshold value and to control the torque at the exoskeleton joints. These signals were fed to the algorithm, which instantly learned and determined the support needed to accomplish a desired task. The algorithm was tested on a 5-degree-of-freedom wearable exoskeleton used in the automation of upper-limb therapeutic exercises. Results indicated that the algorithm offered the ability to adjust assist-as-needed support instantly based on the amount of muscle engagement present in the combined motion of the human-exoskeleton system.

Keywords: Assist-as-Needed, Human-Exoskeleton, Control Systems, Nonlinear Systems

1. Introduction

Wearable exoskeletons are used for augmentation (Farris et al., 2022) and rehabilitation exercises (Zheng et al., 2022). In the former case, these devices are used to augment workers' limbs to help them properly do their jobs without risking injury. For example, a wearable exoskeleton may help a worker carry heavy loads in their day-to-day activities and may also help to automate rehabilitation routines. In both applications, proper interaction and synergy must be established between the human body and the exoskeleton system.

Different algorithms have been developed to this end; One approach is the impedance control

Submitted April 26, 2022
Revised August 22, 2022

strategy, which models the relationship between a desired position and force using a damper-spring-mass system (e.g., (Yang et al., 2006). Another approach involves learning through a neural-network-based control model using offline subject-specific required torque data, then using it to render a force field to assist motion by following a target trajectory (Agarwal & Deshpande, 2017). A third method uses offline EMG signals to estimate joint torques and model predictive control to derive the robot joint torques (e.g., (Teramae et al., 2017)). All three methods are time consuming since a learning step must be performed before using the wearable exoskeleton. Also, the performance of these algorithms will be affected by the varying nature of EMG signals. Therefore, it is preferred to have an instance-based learning assist-as-needed algorithm to update the control law. In this work, we present a hybrid instance-based learning computed torque algorithm to update the assistance provided by the wearable exoskeleton. The proposed algorithm is tested on a 5-degree-of-freedom articulated system.

2. Description of exoskeleton

The exoskeleton system, Figure 1, possesses 5 revolute joints; 4 revolute joints that mimic shoulder kinematics, and 1 that emulates the elbow joint. The dynamics needed to develop and implement the controller can be derived by using either the Euler-Lagrange method (Zefran & Bullo, 2005) or the Recursive Newton-Euler Algorithm (RNEA), based on Newton's laws (Khalil, 2011). The former method uses the kinetic and potential energy of the robot to obtain the equations of motion, yielding a simpler and more elegant set of equations. However, as the number of degrees of freedom (DOF) increases, the complexity of the procedure does as well, making it unfeasible. On the other hand, the RNEA formulation allows the calculation of each link twist (Vector of linear and angular velocities) and wrench (vector of forces and moments) at a given time without the need for derivation and is computationally more efficient to implement for robots with many DOFs. Independently of which method is used, the following closed-form dynamic equation can be derived:

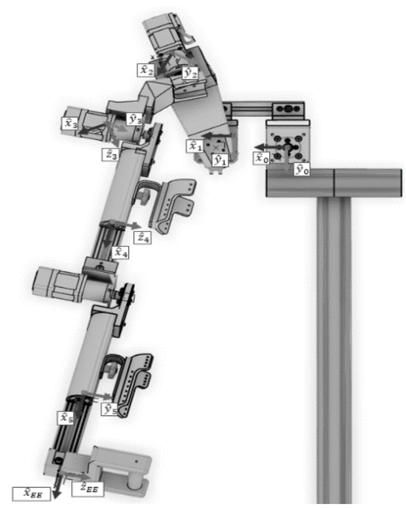


Figure 1. Five-Joint Exoskeleton

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + J(q)^T \mathcal{F} = \tau \tag{1}$$

Where $q \in \mathbb{R}^n$ is a generalized vector of joint coordinates, $M(q) \in \mathbb{R}^{n\times n}$ is the mass matrix, $C(q,\dot{q}) \in \mathbb{R}^{n\times n}$ is the Coriolis & centripetal matrix, $G(q) \in \mathbb{R}^n$ is wrench due to gravity, $J(q) \in \mathbb{R}^{n\times n}$ is the Jacobian matrix, $\mathcal{F} \in \mathbb{R}^n$ is the wrench of forces and moments apply on the environment from the robot at the end-effector coordinates, $\tau \in \mathbb{R}^n$ is a vector of torque and forces exerted by each joint, and n is the number of DOF. Sometimes, it is more convenient to express the summation of $C(q,\dot{q})\dot{q} + G(q) + J(q)\mathcal{F}\ as\ h(q,\dot{q})$, then Equation 1 becomes:

$$M(q)\ddot{q} + h(q,\dot{q}) = \tau \tag{2}$$

In this study, the RNEA is implemented based on forward and backward iteration. In the forward iteration, given a robot with L_{n+1} attached frames from the base to the end-effector, the twist of each link starting from the base frame (L_0) is calculated using Newton's laws and the actual joint positions and rates. Then, using the calculated twists in the backward iteration of the algorithm, the moments and forces exhorted on each link are calculated going backward from the L_{n+1} frame to the L_0 frame. The actual configuration of the robot, the inertia matrices of each link, the link masses, and the center of gravity are required. These parameters need to be defined with respect to the local coordinate frame where each twist is calculated. As shown in Figure 1, each link frame is defined at the center of mass (see (Lynch et al., 2017) for further details).

2.1. Inertia and kinematic parameters of the exoskeleton

The inertia parameters of the exoskeleton are presented in Table 1. Each parameter was estimated from CAD software at the center of mass of each link. In our work, the velocities and forces are derived in spatial coordinates; therefore, the screw axis of each link is derived as well as the initial configuration of the end effector when the joint parameters are set to zero (see Table 2 and Table 3 respectively).

Table 1. Inertia parameters expressed in kg and $kg \cdot m^2 \times 10^{-6}$

Link	1	2	3	4	5
Mass	3.42	0.73	0.73	2.07	1.26
I_{xx}	9,441	1,912	566.5	2,576	595.2
I_{yy}	13,791	639.4	1,720	30,634	9,260
I_{zz}	11,492	1,880	1,533	29,277	9,357
I_{xy}	-1,985	513.7	286.2	-1,618	-719.5
I_{xz}	1,717	1.550	59.34	4,594	354.0
I_{yz}	412.4	-42.91	4.500	353.6	95.58

Table 3. Screw axis of each link

Screw	S_1	S_2	S_3	S_4	S_5
S_x	0.0000	0.0000	-0.3039	-0.9740	-0.9740
S_y	0.0000	0.0000	0.8110	0.2209	0.2209
S_z	1.0000	1.0000	0.5000	0.0498	0.0498
S_{ox}	-0.0223	-0.1740	-0.1704	-0.0476	-0.0565
S_{oy}	-0.0832	-0.1737	-0.1927	-0.2528	-0.3631
S_{oz}	0.0000	0.0000	0.2090	-0.1909	0.5052

Table 2. End-effector initial configuration

$\hat{x}_{ee}$	$\hat{y}_{ee}$	$\hat{z}_{ee}$	$\hat{P}_{ee}$
0.2230	0.0396	-0.9740	0.3454
0.8972	0.3823	0.2209	0.5910
0.3811	-0.9232	0.0498	0.4059
0.0000	0.0000	0.0000	1.0000

3. Computed control law

The robot's equation of motion yields a highly nonlinear system of differential equations that must be linearized to apply linear control system strategies. In the literature, one strategy used to linearize a robot's dynamics is the use of feedback linearization or computed torque control (CTC) (Lynch et al., 2017), (Kelly et al., 2005), (Ullah et al., 2014). The CTC is calculated as follows:

$$\tau_{ctc} = \widehat{M}(q)\left[\ddot{q}_d + k_d\dot{e} + k_p e\right] + \hat{h}(q,\dot{q}) \tag{3}$$

Where $\widehat{M}(q)$ and $\hat{h}(q,\dot{q})$ are the best estimates of the mass matrix, and the vector of forces and torques due to Coriolis & centripetal forces, gravity forces, and end-effector forces. e is the error between the desired joint parameter vector q_d and the actual joint parameter q ($e = q_d - q$). Substituting the computed torque, Equation (3), into the equation of motion of the robot, Equation (2), yields the following equation:

$$\widehat{M}(q)\left[\ddot{q}_d + k_d\dot{e} + k_p e\right] + \hat{h}(q,\dot{q}) - M(q)\ddot{q} - h(q,\dot{q}) = 0 \tag{4}$$

Assuming that $\widehat{M}(q) = M(q)$ and $\hat{h}(q,\dot{q}) = h(q,\dot{q})$ yields the error dynamics set of equations:

$$\ddot{e} + k_d\dot{e} + k_p e = 0 \tag{5}$$

The error dynamics of the system, Equation (5), approaches zero as long as the gain matrices, k_d, and k_p, are positive-definite. Therefore, assuring the stability of the system.

4. Hybrid instance-based learning computed torque algorithm

This study proposes instance-based learning to update an adaptive modifying factor, α, based on current surface electromyography (EMG) data from a subject. This factor calibrates the amount of torque provided by the exoskeleton joints during rehabilitation exercises. Usually, exoskeletons are programmed with given routines – such as elbow flexion – and carry the subject's limb during such exercise. One drawback of such an approach is that the wearer is not motivated to provide any effort to complete the exercise, reducing the efficacy of the rehabilitation session. Incorporating feedback from the wearer into the exoskeleton supplying torque can vary the assistance and improve the synergy of the exoskeleton-human system.

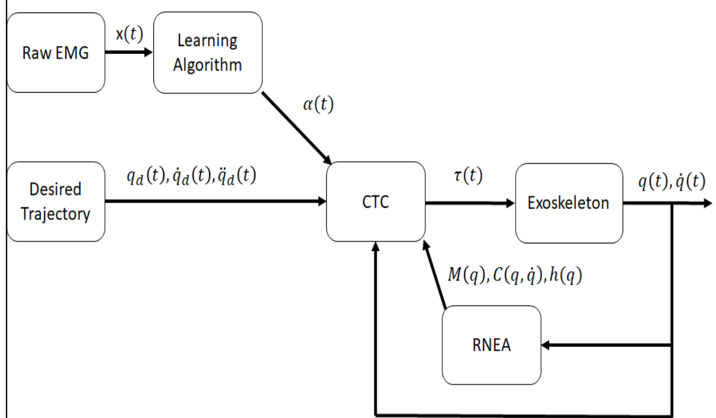
There is evidence that there is a linear relationship between the root mean square (RMS) value of the surface EMG and the contraction force of muscles in the function of the load (Fukuda et al., 2010) (Ullah et al., 2014). The proposed learning algorithm uses a moving RMS window of raw surface EMG data to update the modifying factor (α), then this value is fed back into the control law to modify the calculated torque by CTC in Equation (3). The conditions to update the modifying factor are as follows: First, the maximum RMS value of a given moving RMS time window is extracted, β. Then, this value is compared with new samples, ζ, to determine the value of α, which can be either 1 or ζ/β depending on whether $\beta < \zeta$. This algorithm is shown below along with the proposed closed-loop system. The proposed instance-based learning algorithm is presented in Figure 2a. Depending on the exoskeleton number of DOF, n, there would be *nth* modifying factors, one per each active joint. Grouping these terms together into a modifying diagonal factor matrix, Γ, the desired control torque/force is described in Equation (6).

$$\Gamma = \begin{bmatrix} \alpha_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \alpha_n \end{bmatrix}$$

$$\tau_d = (I - \Gamma) \times \tau_{ctc} \tag{6}$$

Required: Raw EMG data
Results: Updated Modifying Factor
Initialization:
t = time window
β= max (movRMS(raw EMG, t))
***While** (1):*
  if *new data is available:*
    $\zeta =$ *max (movRMS(raw EMG, t))*
    if $\beta < \zeta$ *:*
      $\beta = \zeta$
      $\alpha = 1$
    else*:*
      $\alpha = \frac{\zeta}{\beta}$
    end
  end
  if *is training done:*
    break
  end

(a)



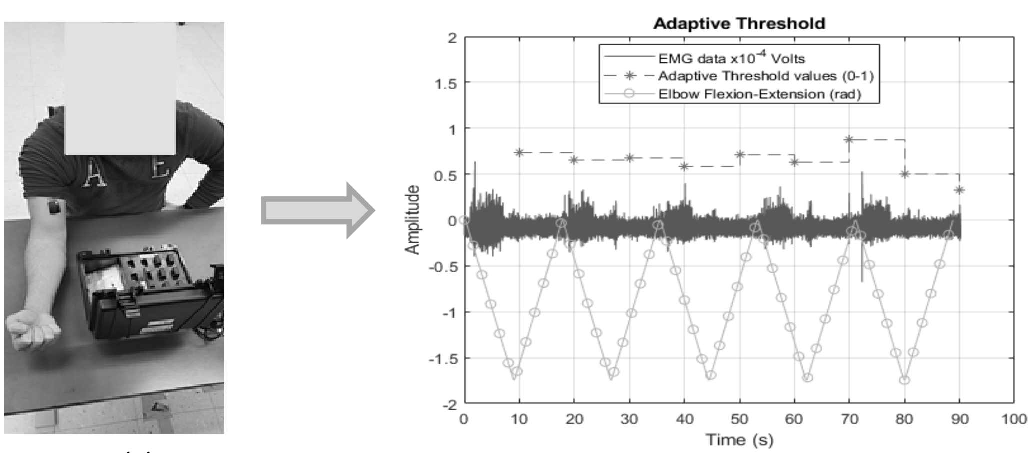
(b)

Figure 2. Proposed Instance-based Algorithm (a), Proposed Closed-loop system (b)

The proposed closed-loop control system is shown in Figure 2b. Where the desired trajectories are fed into the CTC algorithm as well as the learning factor of the targeted muscle, $\alpha(t)$. Also, the system parameters are estimated by the RNEA.

5. Results and discussions

The proposed assist-as-needed algorithm was implemented in an elbow flexion task. The surface EMG signal of the bicep brachii muscle was collected using the Delsys Trigno Wireless EMG system at an approximately sampling rate of $2{,}000Hz$, Figure 3a, when the elbow was flexed from 0° to $100°$. Then, the signal was normalized to have zero mean to remove artifacts due to motion (Konrad, 2005), and filtered with a moving RMS window of 0.125 seconds. The task lasted approximately 90 seconds, then the modifying factor was calculated with an updating rate of 10 seconds, Figure 3b. This information was used in MATLAB where the assist-as-needed algorithm was tested along with the CTC.

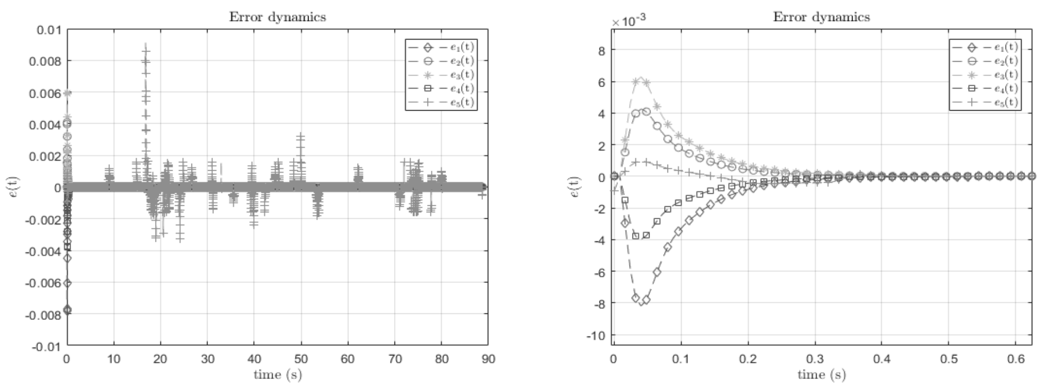


(a) (b)

Figure 3. Data collection system setup (a), Data processing, and learning algorithm (b).

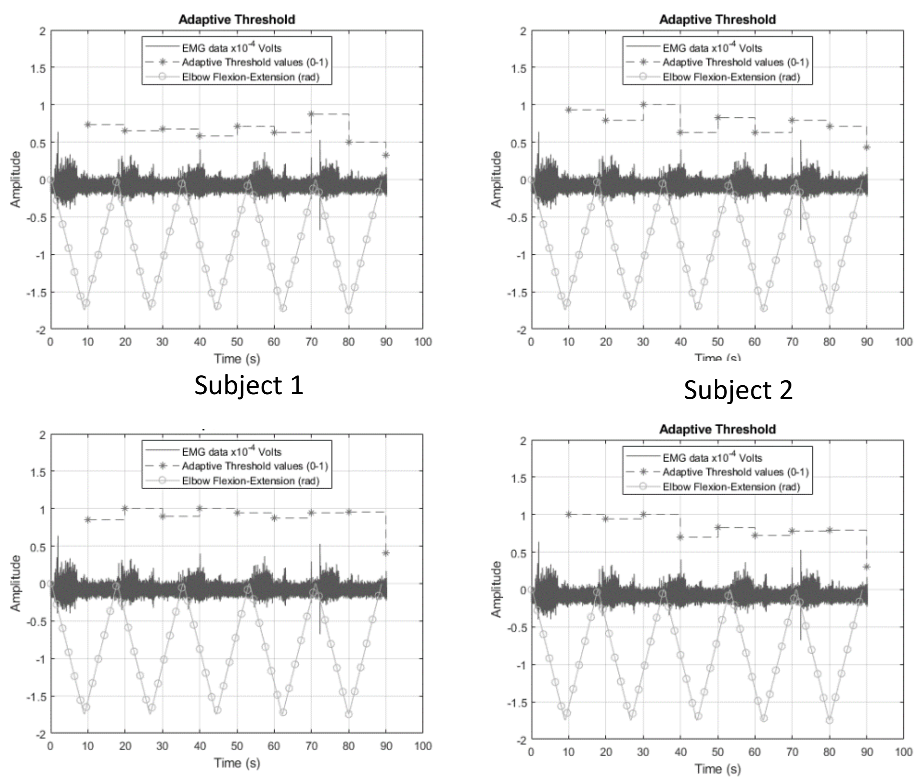
5.1. Implementation without assistance

To demonstrate the effect of adding the EMG instance-based algorithm, the CTC was implemented first to demonstrate that the implemented control law was stable. For this purpose, the gain matrices k_d and k_p were set to $50 \times I^{5\times5}$ and $500 \times I^{5\times5}$, respectably. The error dynamics of the system are shown in Figure 4, as can be inferred from the plot, the system was able to reduce the error dynamics to zero in less than 0.6 seconds.



(a) (b)

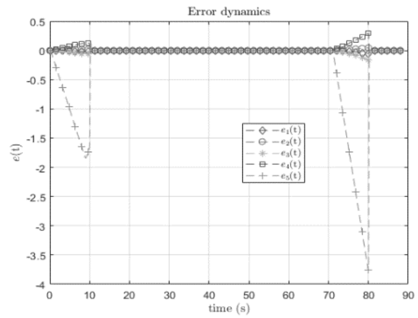
Figure 4. Error Dynamics during a 90-second task (a), Error dynamics during the first 0.6 sec (b)



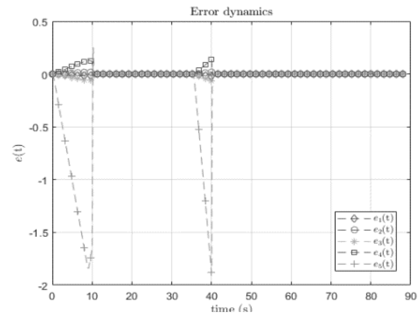
Subject 1 Subject 2

Subject 3 Subject 4

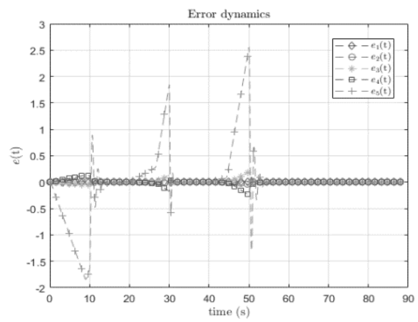
Figure 5. Adaptive Threshold values for each subject



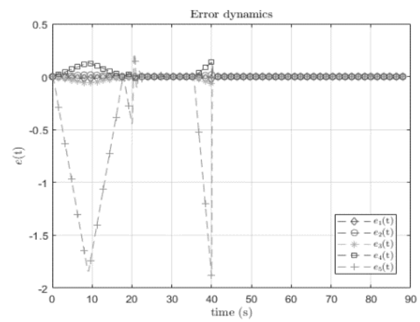
Subject 1



Subject 2

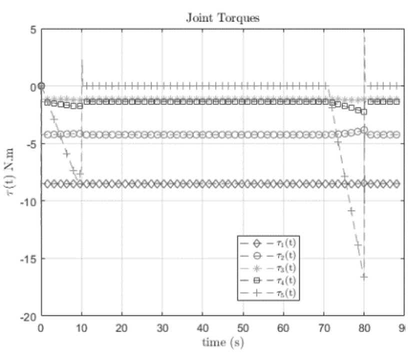


Subject 3

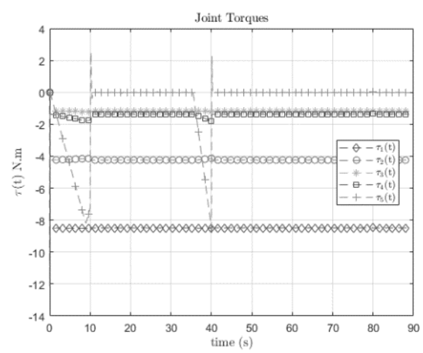


Subject 4

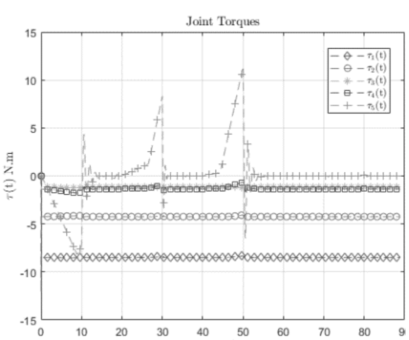
Figure 6. Error dynamics for each subject simulation



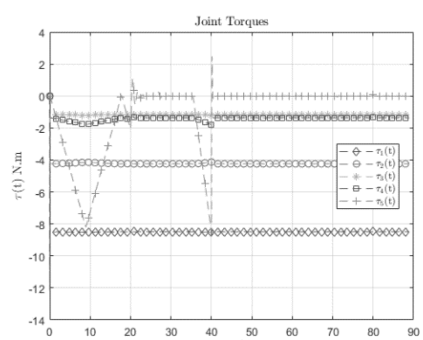
Subject 1



Subject 2



Subject 3



Subject 4

Figure 7. Exoskeleton Joint Torques for each Individual Subject

5.1. Implementation with assistance

Four healthy subjects ages between 18 and 25 volunteered for this study. EMG signals were collected offline for each subject following the above procedure. Modifying factors were calculated, Figure 5, from each subject while performing the elbow flexion task in MATLAB. The error dynamics of the system during each simulation are shown in Figure 6, as well as the joint torque values in Figure 7.

6. Conclusion

In this paper, adaptive EMG-based learning computed torque control algorithm was presented to provide an assist-as-needed control scheme strategy. The proposed method was implemented in MATLAB with data from four different healthy subjects. The assist-as-needed control strategy presented exhibited a dynamic behavior which was expected since the assistance required by each subject varied with the RMS value of the targeted muscle, the bicep brachii. The CTC strategy without assistance could reduce the error dynamics of the system to zero in less than 1 second. On the other hand, when the assist-as-needed control was implemented, the system was capable of correcting the error presented in the desired trajectory for the elbow flexion task in less than 10 seconds for the chosen values of gain matrices k_d and k_p. The error dynamics of the system are expected to be less for real implementation since the simulation does not account for the amount of torque provided by subjects at the moment of the exercise. Despite this, the required torqued from the motors are bounded between reasonable values, $-13N.m$ and $2N.m$.

7. Acknowledgment

This work was supported by WSU Regional Institute on Aging (RIA) and NSF CMMI 2129113 grants.

8. References

Agarwal, P., & Deshpande, A. D. (2017). Subject-specific assist-as-needed controllers for a hand exoskeleton for rehabilitation. *IEEE Robotics and Automation Letters*, *3*(1), 508–515.

Farris, D. J., Harris, D. J., Rice, H. M., Campbell, J., Weare, A., Risius, D., Armstrong, N., & Rayson, M. P. (2022). A systematic literature review of evidence for the use of assistive exoskeletons in defence and security use cases. *Ergonomics*, *just-accepted*, 1–31.

Fukuda, T. Y., Echeimberg, J. O., Pompeu, J. E., Lucareli, P. R. G., Garbelotti, S., Gimenes, R. O., & Apolinário, A. (2010). Root mean square value of the electromyographic signal in the isometric torque of the quadriceps, hamstrings and brachial biceps muscles in female subjects. *J Appl Res*, *10*(1), 32–39.

Kelly, R., Davila, V. S., & Loria, A. (2005). Computed-torque control and computed-torque+ control. *Control of Robot Manipulators in Joint Space*, 227–241.

Khalil, W. (2011). Dynamic modeling of robots using newton-euler formulation. In *Informatics in Control, Automation and Robotics* (pp. 3–20). Springer.

Konrad, P. (2005). The abc of emg. *A Practical Introduction to Kinesiological Electromyography*, *1*(2005), 30–35.

Lynch, K. M., Park, F. C., Lynch, K. M., & Park, F. C. (2017). *Modern Robotics: Mechanics, Planning and Control* (1st ed.). Cambridge University Press.

Teramae, T., Noda, T., & Morimoto, J. (2017). EMG-based model predictive control for physical human--robot interaction: Application for assist-as-needed control. *IEEE Robotics and Automation Letters*, *3*(1), 210–217.

Ullah, M. I., Ajwad, S. A., Islam, R. U., Iqbal, U., & Iqbal, J. (2014). Modeling and computed torque control of a 6 degree of freedom robotic arm. *2014 International Conference on Robotics and Emerging Allied Technologies in Engineering, ICREATE 2014 - Proceedings*, 133–138. https://doi.org/10.1109/iCREATE.2014.6828353

Yang, Y., Wang, L., Tong, J., & Zhang, L. (2006). Arm rehabilitation robot impedance control and experimentation. *2006 IEEE International Conference on Robotics and Biomimetics*, 914–918.

Zefran, M., & Bullo, F. (2005). Lagrangian dynamics. *Robotics and Automation Handbook*, 1–5.

Zheng, L., Hawke, A. L., & Evans, K. (2022). Critical review on applications and roles of exoskeletons in patient handling. *International Journal of Industrial Ergonomics*, *89*, 103290.