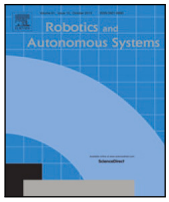
Robotics and Autonomous Systems 171 (2024) 104555

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

journal homepage: www.elsevier.com/locate/robot



Shared autonomy and positive power control for powered exoskeletons ☆

Benjamin Beiter [a,*], Divya Srinivasan [b], Alexander Leonessa [a]

[a] Virginia Tech, Blacksburg, 24060, VA, USA
[b] Clemson University, Clemson, 29634, SC, USA

ARTICLE INFO

Keywords:
Powered exoskeleton
Whole Body Control
Shared control

ABSTRACT

Powered exoskeletons have been shown to significantly reduce physical workload during occupational tasks. Due to this they have great potential impact on future labor practices. However, powered exoskeleton controllers must first be developed that are able to directly assist with task objectives and truly collaborate with users. To address this need we present a shared autonomy control framework that separates the human and exoskeleton control objectives. This allows for creating a feedback-based exoskeleton controller that will help accomplish the user's task goals rather than just assisting in specific motions. We introduce Positive Power control for the human-based controller that is designed to allow the human to directly command work that accomplishes the desired task. While a standard robotic controller can be used for the feedback-based control, we introduce 'acceptance' as a measure of how well the exoskeleton's control objective matches the human's. Both control objectives are implemented in an optimization-based Whole-Body-Control structure. Finally, we introduce a method to update the exoskeleton's objectives to match the humans' such that after operating for a time the exoskeleton can learn to assist the user in accomplishing their task. This framework is implemented on a 10-DoF upper-body powered exoskeleton. The results verify that the control framework works as desired and can form a basis for developing extended methods for directly improving cooperative control for powered exoskeletons.

1. Introduction

Powered or active exoskeletons are robots that are worn by a user in order to assist or enhance the strength and mobility of their body. These exoskeletons have numerous applications, but can primarily be sorted into rehabilitation and occupational exoskeletons [1,2]. Rehabilitation exoskeletons are designed to assist users to achieve mobility that has been lost permanently or to aid in recovering that mobility during rehabilitation. These exoskeletons come in lower-limb exoskeletons, which are primarily focused on assisting in locomotion [3,4], and upper-limb exoskeletons that are focused on providing joint assistance to accomplish manipulation tasks in the presence of some injury [5,6]. Occupational exoskeletons are instead designed to assist the user in accomplishing industrial tasks either by augmenting their strength [7] or helping avoid injury from repetitive action [3,6]. There are many lower-body exoskeletons designed for gait and strength augmentation [4,8,9], as well as several whole-body-exoskeletons in development [10–12], however, most occupational exoskeletons focus on the upper body [13]. Active occupational exoskeletons can significantly reduce the workload when accomplishing tasks [14], and therefore have the potential to revolutionize the experience of labor in the future, with a large market for industrial powered exoskeletons. Despite the increased interest and usage of upper-body powered exoskeletons, one area that has lagged behind in development is the control strategies for exoskeletons. Numerous review papers [2,3,15–19] present the same control strategies with only minor improvements in the performance of the human robot system. There is a need for controllers that can incorporate a new understanding and strategy for human–robot interaction that provides more than just physical assistance to the human, all while maintaining a high level of cooperation and performance of the human–robot system.

A way of organizing the variety of available control approaches for powered exoskeletons is as robotics control approaches and human-aware approaches. Robotics control approaches are standard control strategies for robots such as admittance control, using user input at a force sensor to command a desired velocity for an end-effector or joint [20–22], and impedance control, governing the output force of the robot's motions based on the relationship between the current and desired states of the robot [23–25]. Human-aware control approaches center around calculating exoskeleton action to coordinate

☆ This work was funded in part by the National Science Foundation through the WEAVE project (Grant No. 1839946) and the LEARNER's Project (Grant No. 2033592).

* Corresponding author.
E-mail address: bbeiter1@vt.edu (B. Beiter).

https://doi.org/10.1016/j.robot.2023.104555
Received 11 March 2023; Received in revised form 10 June 2023; Accepted 1 October 2023
Available online 13 October 2023

with the user's musculoskeletal behavior. This includes both Electromyography (EMG) and model-based approaches. EMG approaches use sensor measurements of muscle activity to govern when to activate the exoskeleton's assistance. The simplest method of this is direct EMG-to-Torque control, where the measured muscle effort on a human joint causes the corresponding exoskeleton joint to activate with a proportional assistive torque, scaled by some gain [26]. Model-based approaches use a necessarily simplified model of the human's musculoskeletal system to calculate joint torques and control the exoskeleton to respond accordingly. This varies in complexity from merely estimating the joint torque to better perform human to exoskeleton joint torque scaling, as in the EMG-to-torque case [27], to optimization methods that minimize the human muscle efforts during activity via the action of the exoskeleton [28]. There are also approaches that combine the model-based and robotic control strategies [29]. More sophisticated robot control algorithms have been applied to assist exoskeletons in considering unknowns of human state and intention, such as optimal control [30], adaptive control [31], and LQR [32]. These established exoskeleton control strategies, however, are still limited in scope and are not flexible enough to be applied to uncertain situations with changing objectives [3,18]. There is a distinct need for improved control approaches for powered exoskeletons that features greater cooperation between the human and robot.

One field of Human-Robot-Interaction (HRI) concerned with improved human–robot cooperation is shared control, which focuses on how a human and robot can jointly control a system. Many approaches do this by mapping lower-dimensional human commands to higher-dimensional control inputs to a robot [33]. Alternately, in [34] the authors defined an approach to dynamically allocate different roles between a cobot and its' human partner while trying to accomplish a shared task. This, along with an "agreement indicator", allows the cobot to employ its' own controller when the cobot and human are successfully cooperating, and allows the human to act more freely when they are not. This and other shared control approaches are primarily implemented on robot arms and have not been extended to powered exoskeletons, which are more dynamically coupled than other cooperative robots. The only known similar approach is in [27], where a changing 'fictitious gain' balanced a human-behavior-based controller with a robotic impedance controller for scaling the amount of joint torque assistance. Another approach to improved human-exoskeleton cooperation is shown in [28], where EMG input is used, not as direct EMG-to-torque control, but rather as learning feedback for a pre-designed adaptive controller. This controller can adapt to the user's objectives to the point of off-loading all user muscle effort to accomplish the repetitive task. Inversely, in [33] a user directly controls an assistive robot, while the robot infers the task that the human is attempting to accomplish. When the human presses a button, the robot switches to autonomous control and completes the inferred task. These approaches show the effectiveness of different roles for exoskeleton-human system controllers but are not dynamically allocated roles like the shared control approaches. The importance of measuring and adapting to human acceptance of shared control with a physically coupled robot has been highlighted as one of the three main pillars of physical human–robot interaction [35]. Therefore, we seek to create a shared autonomy control framework for powered exoskeletons that dynamically balances direct human-input and exoskeleton-autonomous control in response to the human's actions during operation. The specific gaps found in the scientific literature are:

- Powered exoskeleton controllers that can provide assistance in accomplishing a task directly.
- Incorporation of the shared control approaches of explicitly defined roles for the human and robot in powered exoskeleton control.
- Empowering the exoskeleton controller with knowledge of when its' objectives do not match the user's and the ability to fully offload the effort required to accomplish the task when the objectives do match.

To address this gap we start by implementing a Whole-Body-Control (WBC) approach. WBC is the go-to strategy for controlling highly dynamic rigid-body systems. It works by setting up a quadratic programming (QP) optimization problem with the linear instantaneous equations of motion of the system in order to accomplish multiple, overlapping control tasks at the same time. This control approach was originally developed for humanoid robots, such as our previous work with THOR [36], or the famous MIT Atlas [37]. WBC has been leveraged to realize many complex behaviors, from straight-leg walking [38] to teleoperation of a humanoid robot by a human user [39]. WBC can be applied to a variety of systems as well, including parallel-structured robots [40], assistive robots [41], and our prior work with powered exoskeletons [42].

With this WBC, we propose a Shared Autonomy control framework to address the needed improvements in powered exoskeleton control. This framework separates the behavior of a human–robot system into feedback-based and human-autonomous controllers. It uses a measure of how well the exoskeleton's control objectives match the human's objectives to balance the different controllers in the WBC. The specific contributions are:

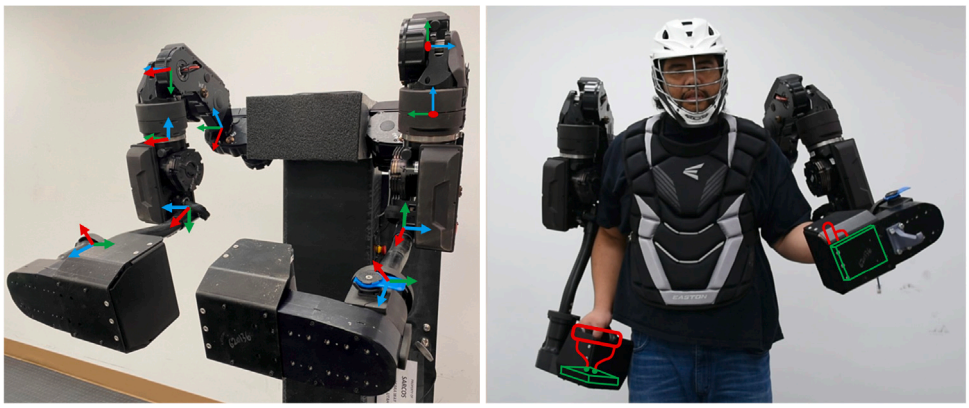
- Separate human-based and feedback-based controllers in a modifiable framework that can shift the required autonomy during repetitive task completion.
- Propose a new, positive-power, human-based controller for direct task assistance from the exoskeleton.
- Introduce "acceptance" as a measure of how well the exoskeleton's control objectives match the human's objectives.
- Define a learning method to update the exoskeleton control objectives to match the human's task objectives.

The framework results in the novel capability of the exoskeleton to perform the task autonomously, without any human effort needed, once it has learned the objective from the user, while still being able to switch back to human control when required. We foresee the application of this control framework in occupational exoskeletons, allowing for increased assistance in accomplishing partially unknown, repetitive tasks, as make up a high proportion of industrial applications. Additionally, by instilling a give-and-take between the authority and knowledge of the human and the robot into the controller itself, this approach is able to generate more complex human–robot interactions, which widens the potential applications of the field in the future.

This paper is structured as follows: Section 2 will give an overview of the human–robot system. Section 3 will present the mathematical model of the robot and the background of the controller. Section 4 lays out the design of the WBC controller. Section 5 presents the details of the Shared Autonomy framework. Section 6 presents the experimental implementation of the controller and its' results. Section 7 is the conclusion.

2. System overview

The system we are using is an upper-body powered exoskeleton. Under feedback-based control the exoskeleton acts as a high-intervention device, guiding the user to pre-planned motions. Under human-based control, the exoskeleton acts as an assistive device while accomplishing a task. The upper-body exoskeleton we use consists of a pair of grounded 5-DoF robot arms designed by Sarcos Robotics [11], shown in Fig. 1(a) . Mass properties, joint limits, and link lengths were obtained from Sarcos to create an open kinematic chain model of the exoskeleton that is used by the controller. The exoskeleton is fixed to a stationary stand so that its' weight is not borne by the user. The exoskeleton contains multiple microcontrollers that provide processing and joint control for each joint-link pair. To operate the exoskeleton, a user stands inside the exoskeleton with their back fixed to the back-plate of the exoskeleton and their hands on the handles. These handles are



(a) (b)

Fig. 1. (a) 10-DOF Empty Powered Exoskeleton with each joint highlighted by coordinate systems (z-axis is joint axis). Left back shoulder joint not pictured. (b) Powered Exoskeleton during use. Handles through which the user interacts are shown highlighted in red, with the FT sensor that measures interaction in green. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

attached to each end-effector of the robot and are equipped with a 6-Degree Of Freedom (DOF) Force-Torque sensor. All interaction forces between the exoskeleton and the user are measured and used as control inputs. This exoskeleton is different than many others in that it does not attach to the human's arm anywhere other than the handles. This means that there can be a mismatch of joint position and link length between the human and exoskeleton without a problem. This also enables the exoskeleton to generate forces at its' end-effector without loading the human's arm at all. A helmet and chestpad are worn as PPE while operating this prototype exoskeleton.

The exoskeleton controller is designed using two sets of open-source software. IHMC Open Robotics Software (ORS) contains the WBC and additional modeling and control algorithms for humanoids and exoskeletons [43]. IHMC Simulation Construction Set (SCS) is a cross-platform simulation and analysis environment, developed in Java, and equipped with tools for modeling robots and testing their controllers [44]. ORS also contains tools for real-time thread applications, which is necessary for consistent controller and communication time steps [45]. The high-level modeling, control, and simulation run on a desktop PC, which communicates to the joint micro-controllers over socket communication.

2.1. Hierarchical control approach

The total control approach for this powered exoskeleton is separated into two parts, high-level and low-level control [46,47], as explained in Fig. 2. The high-level control uses WBC to achieve the exoskeleton's various control objectives. These objectives, referred to as tasks, are arbitrary functions of the robot state and equations of motion. On humanoid robots, these tasks consist of things like dynamic balance, foot pose control, pelvis orientation, arm configuration control, and interaction force control. The dynamics of this exoskeleton are simpler, as it has a fixed base, so the WBC is instead used to share autonomy between the exoskeleton and the user during operation. This WBC consists of two main tasks: the user's control objective, which is set through instantaneous force input at the end effector, and the exoskeleton's autonomous task control, which is designed prior to operation. The output of the high-level control is desired torques for the exoskeleton to realize at each joint that would generate the desired motions. The exoskeleton also comes with joint controllers that run on the microcontrollers within the exoskeleton. These are current controllers that track desired torque commands at every joint and communicate joint states to a high-level controller.

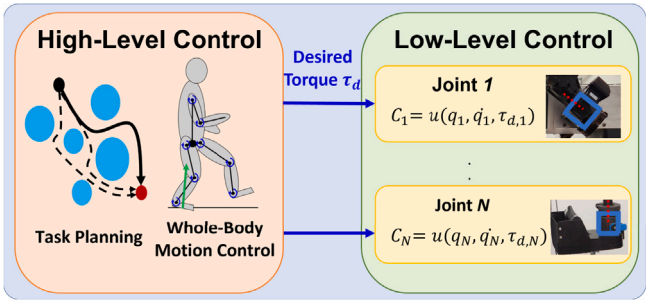


Fig. 2. Multi-level control approach showing the high-level control planning motions via joint torques, and the low-level current control tracking those desired torques.

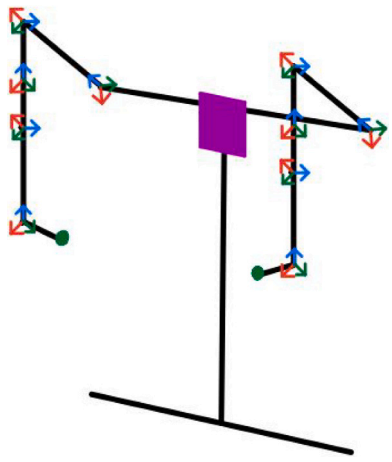


Fig. 3. Schematic diagram of the exoskeleton. Joint coordinate systems are highlighted, matching Fig. 1(a). The purple outline is where a user would stand and the green dots represent the handles the user would hold onto. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

3. Mathematical formulation

In this section we present the mathematical representation of the robotic system presented above, the derivation of the Whole-Body-Controller used here, and the combination of both to output the desired torques required for control. As illustrated in Fig. 3, the exoskeleton

can be modeled as a fixed-base system of rigid bodies with n actuated degrees of freedom (DoF). Each DoF has a coordinate system attached such that positive joint rotation and joint torque is about the z-axis. We define the vector of joint variables $\mathbf{q}$ and the vector of joint torques $\boldsymbol{\tau}$ as

$$\mathbf{q} \in \mathbb{R}^n, \qquad \boldsymbol{\tau} \in \mathbb{R}^n. \tag{1}$$

Using Featherstone inverse dynamics algorithms [48], the standard equation of motion for a robot or any system of rigid bodies can be efficiently found and written as

$$\mathbf{H}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) = \boldsymbol{\tau} + \sum_c \mathbf{J}_{\mathrm{e}}^T \mathbf{w}_{\mathrm{e}}. \tag{2}$$

Here $\mathbf{H} \in \mathbb{R}^{n\times n}$ is the mass-inertia matrix, $\mathbf{C} \in \mathbb{R}^{n\times n}$ is the Coriolis matrix, $\mathbf{G} \in \mathbb{R}^n$ is the gravity-matrix, c is the number of external contact points in the rigid body system, $\mathbf{J}_{\mathrm{e}} \in \mathbb{R}^{6\times n}$ is the jacobian of the external contact points in the inertial frame, and $\mathbf{w}_{\mathrm{e}} \in \mathbb{R}^6$ is the external contact wrench at each point.

3.1. Whole body control

WBC is an approach for controlling highly dynamic rigid-body systems by setting up a quadratic optimization problem with the linear instantaneous equations of motion (Eq. (2)) to control the system to accomplish multiple, overlapping control tasks at the same time. A task is a specific control objective to be accomplished. There are many different kinds of tasks, but all tasks can be encoded as achieving a desired task velocity

$$\mathbf{v}_k = \mathbf{J}_k \dot{\mathbf{q}}, \tag{3}$$

and a corresponding task acceleration

$$\dot{\mathbf{v}}_k = \dot{\mathbf{J}}_k \dot{\mathbf{q}} + \mathbf{J}_k \ddot{\mathbf{q}}, \tag{4}$$

where $\mathbf{J}_k$ is the task Jacobian. Here $k = 1, 2 \ldots n_{\mathrm{T}}$ and n_{T} is the number of tasks. For each task, a task controller

$$\dot{\mathbf{v}}_{k,\mathrm{des}} = \boldsymbol{\pi}_k(\mathbf{q}, \dot{\mathbf{q}}), \tag{5}$$

is designed to give the desired acceleration. This controller $\boldsymbol{\pi}_k$ can be designed using any arbitrary strategy. This means that each task is defined as the desired rate of change of some value $\mathbf{v}_k$, and the task Jacobian $\mathbf{J}_k$ is the mapping from the joint variables to the task value. These tasks can be joint accelerations, spatial acceleration of certain points such as end-effectors, or any other measure that can be encoded as a desired acceleration.

To include each task in a WBC, a quadratic cost function is defined for each task that relates the actual and desired task accelerations

$$\begin{aligned} J_k &= \|\mathbf{C}_k(\dot{\mathbf{v}}_{k,\mathrm{des}} - \dot{\mathbf{v}}_k)\|^2 \\ &= \|\mathbf{C}_k(\boldsymbol{\pi}_k(\mathbf{q}, \dot{\mathbf{q}}) - \dot{\mathbf{J}}_k\dot{\mathbf{q}} - \mathbf{J}_k\ddot{\mathbf{q}})\|^2, \end{aligned} \tag{6}$$

Where $\mathbf{C}_k$, is a weighting matrix (usually diagonal) that gives task prioritization. These tasks are combined into a total WBC cost-function

$$J = \sum_{k=1}^{n_T} J_k + \lambda_{\ddot{\mathbf{q}}}^T ||\ddot{\mathbf{q}}||^2, \tag{7}$$

where $\lambda_{\ddot{\mathbf{q}}} \in \mathbb{R}^N$ is a vector of weights penalizing the effort, and $||\ddot{\mathbf{q}}||^2$ is a vector of squared joint acceleration magnitudes. This cost function J can be minimized using the joint accelerations $\ddot{\mathbf{q}}$ as the optimization variables, and subject to a single inequality constraints: the upper and lower bounds on joint positions $\underline{\mathbf{q}}$ and $\bar{\mathbf{q}}$. These can be defined as

$$\underline{\mathbf{q}} \le \mathbf{q} + T\dot{\mathbf{q}} + \frac{1}{2}T^2\ddot{\mathbf{q}} \le \bar{\mathbf{q}} \tag{8}$$

where T is a time constant that governs how abruptly the joints shall avoid their range of motion limits. If the total cost function (7) is minimized to 0, each controller would be perfectly implemented. This is only the case for non-overlapping tasks. In most cases, including this paper, there are overlapping tasks and constraints so the minimization implements each controller to the greatest extent that it can, highlighting the importance of the task prioritization matrices.

3.2. Optimization problem formulation

To solve this optimization problem, the cost function (7) and constraints need to be written in a standard form to input the problem into a standard optimization solver. The standard form of a Quadratic Programming problem is

$$\begin{aligned} &\min_{\ddot{\mathbf{q}}} \quad \frac{1}{2}\ddot{\mathbf{q}}^T\boldsymbol{\mathcal{H}}\ddot{\mathbf{q}} + \mathbf{f}^T\ddot{\mathbf{q}} \\ &\text{subject to: } \mathbf{A}_{\mathrm{ineq}}\ddot{\mathbf{q}} \le \mathbf{b}_{\mathrm{ineq}}. \end{aligned} \tag{9}$$

To achieve this, each task cost-function (6) is stacked into task matrices

$$\mathbf{b} = \begin{bmatrix} \boldsymbol{\pi}_1(\mathbf{q}, \dot{\mathbf{q}}) \\ \dots \\ \boldsymbol{\pi}_{n_{\mathrm{T}}}(\mathbf{q}, \dot{\mathbf{q}}) \end{bmatrix}, \qquad \bar{\mathbf{J}} = \begin{bmatrix} \mathbf{J}_1 \\ \dots \\ \mathbf{J}_{n_{\mathrm{T}}} \end{bmatrix}, \qquad \dot{\bar{\mathbf{J}}} = \begin{bmatrix} \dot{\mathbf{J}}_1 \\ \dots \\ \dot{\mathbf{J}}_{n_{\mathrm{T}}} \end{bmatrix}, \tag{10}$$

such that the total cost function (7) can be written as

$$J = \|\mathbf{C}(\mathbf{b} - \dot{\bar{\mathbf{J}}}\dot{\mathbf{q}} - \bar{\mathbf{J}}\ddot{\mathbf{q}})\|^2 + \lambda_{\ddot{\mathbf{q}}}^T||\ddot{\mathbf{q}}||^2, \tag{11}$$

where the block-diagonal, composite task-prioritization matrix is

$$\mathbf{C} = \begin{bmatrix} \mathbf{C}_1 & & 0 \\ & \ddots & \\ 0 & & \mathbf{C}_{n_{\mathrm{T}}} \end{bmatrix}. \tag{12}$$

To write the composite cost-function (11) in the standard form (9), it is rewritten as

$$\begin{aligned} J &= \|\mathbf{C}(\mathbf{b} - \dot{\bar{\mathbf{J}}}\dot{\mathbf{q}} - \bar{\mathbf{J}}\ddot{\mathbf{q}})\|^2 + \lambda_{\ddot{\mathbf{q}}}^T\ddot{\mathbf{q}} \\ &= ((\mathbf{b} - \dot{\mathbf{J}}\dot{\mathbf{q}})^T - \ddot{\mathbf{q}}^T\mathbf{J}^T)\mathbf{C}^T\mathbf{C}((\mathbf{b} - \dot{\bar{\mathbf{J}}}\dot{\mathbf{q}}) - \bar{\mathbf{J}}\ddot{\mathbf{q}}) + \lambda_{\ddot{\mathbf{q}}}^T\ddot{\mathbf{q}} \\ &= (\mathbf{b} - \dot{\bar{\mathbf{J}}}\dot{\mathbf{q}})^T\mathbf{Q}(\mathbf{b} - \dot{\bar{\mathbf{J}}}\dot{\mathbf{q}}) + \ddot{\mathbf{q}}^T\bar{\mathbf{J}}^T\mathbf{Q}\bar{\mathbf{J}}\ddot{\mathbf{q}} + \lambda_{\ddot{\mathbf{q}}}^T\ddot{\mathbf{q}} - 2(\mathbf{b} - \dot{\bar{\mathbf{J}}}\dot{\mathbf{q}})^T\mathbf{Q}\bar{\mathbf{J}}\ddot{\mathbf{q}} \\ &= \mathcal{E} + \frac{1}{2}\ddot{\mathbf{q}}^T\boldsymbol{\mathcal{H}}\ddot{\mathbf{q}} + \mathbf{f}^T\ddot{\mathbf{q}} \\ &\text{Where: } \mathcal{E} = (\mathbf{b} - \dot{\bar{\mathbf{J}}}\dot{\mathbf{q}})^T\mathbf{Q}(\mathbf{b} - \dot{\bar{\mathbf{J}}}\dot{\mathbf{q}}), \qquad \boldsymbol{\mathcal{H}} = 2\bar{\mathbf{J}}^T\mathbf{Q}\bar{\mathbf{J}} + \boldsymbol{\Lambda}_{\ddot{\mathbf{q}}}, \\ &\qquad\qquad \mathbf{f}^T = -2(\mathbf{b} - \dot{\bar{\mathbf{J}}}\dot{\mathbf{q}})^T\mathbf{Q}\bar{\mathbf{J}}. \end{aligned} \tag{13}$$

Here $\boldsymbol{\Lambda}_{\ddot{\mathbf{q}}} = \mathrm{diag}(\lambda_{\ddot{\mathbf{q}}})$. Note that $\mathcal{E}$ is a constant here, and so need not be included in the following minimization:

$$\min_{\ddot{\mathbf{q}}} J - \mathcal{E} = \min_{\ddot{\mathbf{q}}} \frac{1}{2}\ddot{\mathbf{q}}^T\boldsymbol{\mathcal{H}}\ddot{\mathbf{q}} + \mathbf{f}^T\ddot{\mathbf{q}}. \tag{14}$$

In the above derivation, the squared task-prioritization matrix is

$$\mathbf{Q} = \mathbf{C}^T\mathbf{C} = \begin{bmatrix} \mathbf{C}_1^T\mathbf{C}_1 & & 0 \\ & \ddots & \\ 0 & & \mathbf{C}_{n_{\mathrm{T}}}^T\mathbf{C}_{n_{\mathrm{T}}} \end{bmatrix} = \begin{bmatrix} \mathbf{Q}_1 & & 0 \\ & \ddots & \\ 0 & & \mathbf{Q}_{n_{\mathrm{T}}} \end{bmatrix}, \tag{15}$$

where $\mathbf{Q}_k$ is the squared task-prioritization for the kth task.

Finally, any inequality constraint is to be written in the form $\mathbf{A}_{\mathrm{ineq}}\ddot{\mathbf{q}} \le \mathbf{b}_{\mathrm{ineq}}$. The upper and lower bounds on joint positions (8) can be written in the form

$$\begin{bmatrix} -\mathbf{I}_n \\ \mathbf{I}_n \end{bmatrix} \ddot{\mathbf{q}} \le \begin{bmatrix} -(\underline{\mathbf{q}} - \mathbf{q} - T\dot{\mathbf{q}})\frac{2}{T^2} \\ (\bar{\mathbf{q}} - \mathbf{q} - T\dot{\mathbf{q}})\frac{2}{T^2} \end{bmatrix}. \tag{16}$$

The resulting optimization problem can be solved with any readily available QP solving algorithm. Once the optimization has a solution $\ddot{\mathbf{q}}^*$ the joint torques can be recovered from the inverse dynamics:

$$\boldsymbol{\tau} = \mathbf{H}(\mathbf{q})\ddot{\mathbf{q}}^* + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) - \sum_e \mathbf{J}_e^T\mathbf{w}_e. \tag{17}$$

These torques are then commanded to the exoskeleton joint controllers to realize the desired motion.

4. Controller design

In this section, we present the choice of tasks included in our WBC problem, as well as the associated task controllers. With WBC, the complexity of this choice of task usually arises from under-actuated robots with multiple limbs and many interaction points. For this exoskeleton,

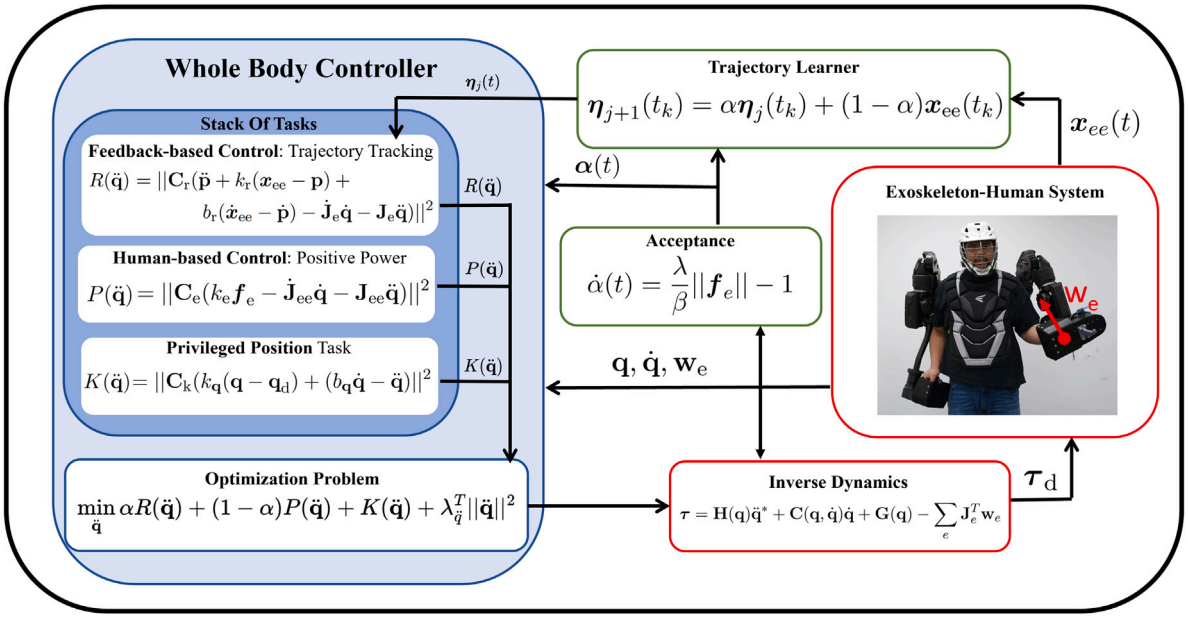


Fig. 4. Diagram of the structure of the full controller.

the dynamics are not as complex, as it is a fixed base system of rigid bodies. Instead, the complexity arises from the control application. In this case, there are two competing control objectives for the robot: following all user inputs with human-based control, and achieving its' own learned task objectives with feedback-based control. Therefore the exoskeleton is operated with three WBC tasks active, a human-based task, a feedback-based task, and a supporting privileged position task. To illustrate the application within a WBC as well as the flexibility of the shared autonomy framework, we present two possible human-based tasks that are interchangeable. These tasks are integral to the full controller structure shown in Fig. 4. This Figure also shows the integration into the Shared Autonomy Framework, which is explained in more detail in Section 5.

4.1. Feedback-based task: PD trajectory tracking

Any exoskeleton-autonomous control objective can be represented as a WBC task. Here, a trajectory tracking task is used. For the WBC, the trajectory is treated as a series of positions at any timestep given by the trajectory $\mathbf{p} = \boldsymbol{\eta}(t)$, with the derivatives also available. The task controller regulates displacement between the end effector position $\boldsymbol{x}_{ee}$ and the desired position $\mathbf{p}$. We define a task-space position feedback control law for the end effector to track a desired trajectory. The proportional–derivative (PD) controller for the desired end effector acceleration is

$$\ddot{\boldsymbol{x}}_{ee,des} = k_r(\mathbf{p} - \boldsymbol{x}_{ee}) + b_r(\dot{\mathbf{p}} - \dot{\boldsymbol{x}}_{ee}), \tag{18}$$

where k_r and b_r are gains. The corresponding WBC task is

$$\boldsymbol{\pi}_2 = \ddot{\boldsymbol{x}}_{ee,des}, \qquad \mathbf{J}_2 = \mathbf{J}_e, \tag{19}$$

where $\mathbf{J}_{ee}$ is the end effector jacobian. The overall cost function of the task is

$$R(\ddot{\mathbf{q}}) = J_r = \|\mathbf{C}_r(\ddot{\mathbf{p}} + k_r(\mathbf{p} - \boldsymbol{x}_{ee}) + b_r(\dot{\mathbf{p}} - \dot{\boldsymbol{x}}_{ee}) - \dot{\mathbf{J}}_e\dot{\mathbf{q}} - \mathbf{J}_e\ddot{\mathbf{q}})\|^2, \tag{20}$$

where $\mathbf{C}_r \in \mathbb{R}^{NxN}$ and $N = 3$ for considering linear force and motion or $N = 6$ when also considering moment and rotational motion.

4.2. Human-based task: Positive power

The positive power task controls the exoskeleton to instantaneously increase the power the robot and human are outputting together. It is similar to admittance control, but instead of realizing an output velocity, it tracks a desired output acceleration at the end-effector of the robot $\ddot{\boldsymbol{x}}_e$. The end effector is fixed relative to the point that the user applies an input force $\boldsymbol{f}_e$. The desired acceleration output is proportional to this force input

$$\ddot{\boldsymbol{x}}_{ee,des} = k_e\boldsymbol{f}_e, \tag{21}$$

where k_e is the force-to-motion scaling gain. The Task for this controller is then

$$\boldsymbol{\pi}_1 = k_e\boldsymbol{f}_e, \qquad \mathbf{J}_1 = \mathbf{J}_{ee}, \tag{22}$$

where $\mathbf{J}_{ee}$ is the Jacobian of the end effector. The overall cost function of the task is

$$P(\ddot{\mathbf{q}}) = J_e = \|\mathbf{C}_e(k_e\boldsymbol{f}_e - \dot{\mathbf{J}}_{ee}\dot{\mathbf{q}} - \mathbf{J}_{ee}\ddot{\mathbf{q}})\|^2, \tag{23}$$

where $\mathbf{C}_e \in \mathbb{R}^{NxN}$ and $N = 3$ for considering linear force and motion or $N = 6$ when also considering moment and rotational motion.

4.3. Alternative human-based control task: Force output

To explore the performance of positive power control we compare it to force output control, which is an alternative control approach similar to approaches for augmentation exoskeletons [7]. In this control task, a desired wrench at an end-effector is controlled to be a scaled output of the input wrench,

$$\mathbf{w}_{e,des} = k_f\boldsymbol{f}_e, \tag{24}$$

where k_f is the force input to force output scaling gain. Here the goal is to produce a larger force at the end effector than the user does, thus the exoskeleton does more work. This task, however, cannot be included in the WBC approach in the same way as the other tasks. Instead, the optimization problem is first solved for desired joint accelerations, $\ddot{\mathbf{q}}^*$, with any other active tasks. Then the inverse dynamics of the system (Eq. (17)) can be used to find the desired output torque that would realize this output wrench.

$$\boldsymbol{\tau} = \mathbf{H}(\mathbf{q})\ddot{\mathbf{q}}^* + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) - \sum_e \mathbf{J}_e^T\mathbf{w}_{e,des}. \tag{25}$$

In this way the robot is attempting to move the end effector at a desired acceleration while producing a desired output force. These objectives,

however, are conflicting. Instead of being balanced by the optimization, the objectives are balanced by how much force is being exerted on the end effector from the environment. The expected cooperation would be less.

4.4. Privileged position task

In addition to the two autonomy tasks, there is a third task. The upper body exoskeleton arms are redundant manipulators, meaning in certain states the arms could generate an end effector output force or motion in multiple ways. Therefore, there could be multiple similar solutions to the QP problem. To address this potential issue, we define a privileged position task, which will provide guidance to the QP problem in which solution to choose if there are multiple possibilities. This task consists of a set of privileged joint positions $\mathbf{q}_{\mathrm{d}}$, and a joint-space PD controller for the desired joint acceleration

$$\ddot{\mathbf{q}}_{\mathrm{d}} = k_{\mathbf{q}}(\mathbf{q}_{\mathrm{d}} - \mathbf{q}) - b_{\mathbf{q}}\dot{\mathbf{q}}, \tag{26}$$

that tracks the privileged joint positions, with gains $k_{\mathbf{q}}$ and $b_{\mathbf{q}}$. The privileged joint positions are defined as a known, safe position for the exoskeleton to be in. The WBC task is

$$\boldsymbol{\pi}_3 = \ddot{\mathbf{q}}_{\mathrm{d}}, \qquad \mathbf{J}_3 = \mathbf{I}_n. \tag{27}$$

The overall cost function of the task is

$$K(\ddot{\mathbf{q}}) = J_{\mathrm{k}} = \|\mathbf{C}_{\mathrm{k}}(k_{\mathbf{q}}(\mathbf{q} - \mathbf{q}_{\mathrm{d}}) + (b_{\mathbf{q}}\dot{\mathbf{q}} - \ddot{\mathbf{q}}))\|^2, \tag{28}$$

where $\mathbf{C}_k \in \mathbb{R}^{nxn}$ is the weighting matrix. Weighting this task much less than the other tasks in the QP problem will bias all solutions towards the given set of privileged joint positions, without interfering with the other objectives. A properly chosen set of privileged joint positions can improve the safety of the system by biasing all solutions towards a safe and comfortable position for a user.

5. Shared autonomy

We present a shared autonomy framework for balancing human and exoskeleton autonomy while accomplishing a task. The objective of the framework is to allow the exoskeleton to assist in accomplishing a task directly rather than just assisting in specific motions. This includes empowering the robot to accomplish the task autonomously once it knows the specific objectives of the task. Two additions are needed for this, a way for the exoskeleton to learn the task objectives that the user knows, and a way of knowing when its' internal task objectives are wrong and need to be updated. The former is done on a task-by-task basis, and the latter is done with a measure of *acceptance*. This acceptance is used as an additional weight in the WBC for the human and exoskeleton autonomous tasks.

5.1. Acceptance

Acceptance is an estimated value of how much the user accepts the exoskeleton's internal control objectives as matching their own task objectives. The acceptance is measured as an integral of the force input into the robot at each end effector, and is a scalar value on the interval $[0, 1]$. It is a measure of the amount of physical effort that the user is exerting over time to control the robot, and is expected to correlate with the cognitive workload or "cost" for controlling the movement. At zero acceptance, it is assumed the exoskeleton does not know the task objective, and so is operated under the human control input task. At full acceptance, the exoskeleton operates fully autonomously and does not follow the human's inputs. This happens when the user is no longer inputting force into the handles to correct the exoskeleton's motions towards the user's objectives. Operating at an intermediate acceptance value results in shared control autonomy between the user and exoskeleton through the balance of the WBC tasks. Conceptually, the acceptance $\alpha(t)$ is an integral of the recent effort put forth by the user to correct the robot. As such it is measured by setting the derivative of the acceptance proportional to the force measured from the user

$$\dot{\alpha}(t) = \frac{\lambda}{\beta}\|\boldsymbol{f}_e\| - 1, \tag{29}$$

where λ is a learning rate, and β is a normalization value. This normalization value is set such that when the magnitude of the force input is greater than the value, the acceptance increases, and when it is less the acceptance decreases. Both λ and β can be tuned. Additionally, the acceptance is practically bounded within the interval $[0.1, 1]$ such that the exoskeleton can have full autonomy, but its' task will never fully be overridden by the user so it can keep indicating to the user what its' objectives are.

The acceptance is added to the WBC problem as an adjustment to the weights of the control tasks. The overall cost function (7), rather than being written as a simple sum of the task costs, becomes

$$\min_{\ddot{\mathbf{q}}} \alpha R(\ddot{\mathbf{q}}) + (1-\alpha)P(\ddot{\mathbf{q}}) + K(\ddot{\mathbf{q}}) + \lambda_{\ddot{q}}^T ||\ddot{\mathbf{q}}||^2. \tag{30}$$

This weighting can also be implemented in the standard form of the optimization problem by simply scaling the task weighting matrices (15) by the acceptance

$$\mathbf{Q} = \begin{bmatrix} (1-\alpha)\mathbf{Q}_{\mathrm{e}} & 0 & 0 \\ 0 & \alpha\mathbf{Q}_{\mathrm{r}} & 0 \\ 0 & 0 & \mathbf{Q}_{\mathrm{k}} \end{bmatrix}. \tag{31}$$

5.2. Task learning

To demonstrate the application of the Shared Autonomy control approach, it is applied to a trajectory tracking task, though a learning method can be implemented for any type of control task. For the trajectory tracking task shown in Section 4.1, we conceptualize the user's objective as to move the exoskeleton's end-effector such that it follows a desired trajectory $\boldsymbol{\phi}(t)$ defined on a period $t \in [t_i, t_f]$. The trajectory task is assumed to be repetitive, with the same goal trajectory $\boldsymbol{\phi}(t)$ each cycle. The feedback-based task is to follow a desired trajectory $\boldsymbol{\eta}(t)$. The exoskeleton's learning objective is to converge the control trajectory to the human's desired trajectory

$$\boldsymbol{\eta}_j(t) \to \boldsymbol{\phi}(t), \tag{32}$$

where $j = 0, 1, \ldots$ is the cycle number. Each desired trajectory is defined by n_p way-points. The learning procedure is to begin with an initial desired trajectory $\boldsymbol{\eta}_1(t)$. When it does not match the user's objective, the user exerts force to direct the exoskeleton. This both causes the acceptance to decrease so the user can perform their desired task under their own autonomy, and moves the exoskeleton from its' initial desired trajectory. When this happens, the exoskeleton updates its' desired trajectory for the next cycle based on this user-corrected motion.

There are two ways for the learning to occur. The first is *trajectory direct overwrite*, in which the trajectory that is to be tracked in the next cycle is set to match the realized trajectory from this cycle

$$\boldsymbol{\eta}_{j+1}(t_k) = \boldsymbol{x}_{\mathrm{ee}}(t_k) \qquad j, k = 1, 2, \ldots, n_p, \tag{33}$$

at a time-step t_k associated with each trajectory way-point, within the period of cycle number j. The second learning method is *trajectory soft-update*. With this strategy, the desired autonomous trajectory is updated towards the achieved trajectory, scaled by the acceptance measure

$$\boldsymbol{\eta}_{j+1}(t_k) = \alpha\boldsymbol{\eta}_j(t_k) + (1-\alpha)\boldsymbol{x}_{\mathrm{ee}}(t_k) \qquad j, k = 1, 2, \ldots, n_p. \tag{34}$$

At full acceptance the exoskeleton will operate autonomously and will not update its' objectives. At zero acceptance the exoskeleton will act fully under human autonomy and will act like direct overwrite. The soft-update strategy performs better, as it makes the final learned trajectory more robust to both disturbances from the environment and error in low-level control tracking.

Overall, the final expected behavior of the system is that the user begins to use the exoskeleton, which knows what type of task must be

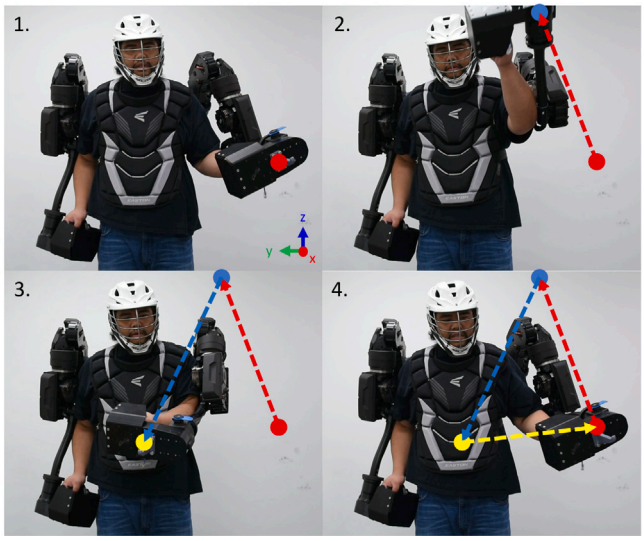


Fig. 5. User performing the trajectory tracking task with the exoskeleton. Only the left arm is active for testing. View is of the YZ Plane in the task-space.

accomplished, and therefore which WBC task to use, but does not know the exact objectives (desired end-effector positions). The user begins to command the robot under full human autonomy to accomplish the task. As it does so, the exoskeleton learns the user's objective. The user has to correct the exoskeleton less, so the acceptance increases until eventually the exoskeleton is operating fully autonomously, accomplishing the same task the user was. When the acceptance is high, and the exoskeleton is operating autonomously, the total effort required of the human will be lower. Without having to focus on operating the exoskeleton, the user is freed to spend more effort on observing the environment, planning, and other higher-level decisions related to their task. At any time the user can exert more force to correct the exoskeleton's autonomous performance of the task, either for small adjustments of priority or for fully changing the task. In this way the user's effort is only high when directing the exoskeleton to achieve a new objective or respond to the environment.

6. Results and discussion

To Illustrate the operation of the controller we implemented the Shared Autonomy framework on the trajectory tracking task with the Sarcos Upper-Body Powered Exoskeleton. Fig. 4 shows the diagram of the final controller structure. All trials are done with only linear motion and force tasks, not moment or angular motion. All trials are done with only the left arm active in order to simplify this validation experiment. The right arm could be operated under an identical controller, or a new task controller could be designed for bimanual manipulation, but it is left dormant here. Additionally, all trials are done with no load on the end-effector to test achieving of desired motions. The privileged position is defined as the zero position for all the joints, shown in Fig. 3, except for the elbow, which is at 90°. We begin by validating the functionality of both the human-based and feedback-based tasks independently. Then the full framework is implemented to show how it achieves the desired outcomes. Fig. 5 shows the exoskeleton during operation along with the trajectory tracking task.

6.1. Feedback-based control

To validate the feedback-based control task, the exoskeleton is operated with only this task active and no human input. This matches the shared autonomy case where the acceptance is $\alpha = 1$. The trajectory given is a triangle in the yz frame, a forearm's width in the x direction from the base of the robot. The gains for the PD controller are $k_r = 175$ and $b_r = 12$. These gains were tuned through experimentation to provide small or no oscillations during tracking, so as not to cause any unexpected motions for a user. The trade-off for this is longer settling times and higher tracking error. The task weights are given in Table 2. The desired trajectory and the accuracy of the tracking task are shown in Fig. 6.

The desired trajectory is tracked successfully, with the exoskeleton repeating the same trajectory over and over, even if it is not perfectly accurate. The variation between the 5 trajectory cycles shown in this trial are given in Table 1. The total variance between cycle i and j is defined as

$$v_{i,j} = \int_{t_o}^{t_f} \|\boldsymbol{x}_{ee,i}(t) - \boldsymbol{x}_{ee,j}(t)\| dt. \tag{35}$$

From these results it can be seen that the variance between the trajectory tracked in each cycle is very small compared to the total distance traveled in the cycle. The tracking error over the trial averages to $e_{avg} = 0.0193$ m, which comes from the fact that the PD controller for trajectory tracking does not have a feed-forward or integral term to drive its error to zero. As it is, these results show that the feedback-based task works, shows close to accurate tracking, and is repeatable so that the human can depend on the exoskeleton to do the same thing without variation if it is in autonomous mode.

Table 1
Variance of the end-effector position while repeatedly tracking a stationary trajectory. Total travel is the total distance moved during the cycle, while total variance is the difference between the actual end-effector trajectory in the final cycle and each previous cycle, integrated over the cycle. The percent variance is the amount of variance compared to the total distance traveled.

	Cycle 1	Cycle 2	Cycle 3	Cycle 4	Cycle 5
Total Travel (m)	2.35	2.33	2.34	2.33	2.38
Total Variance (mm)	37	14	13	14.3	0
Percent Variance	1.57%	0.60%	0.55%	0.62%	0.0%

Table 2
Task Weighting Values for the Various Task Implementations.

	$\mathbf{Q}_e$	$\mathbf{Q}_r$	$\mathbf{Q}_k$
Human-Based Control Only - Positive Power	$100 \cdot \mathbf{I}_3$	$0 \cdot \mathbf{I}_3$	$5 \cdot \mathbf{I}_3$
Human-Based Control Only - Force Output	$0 \cdot \mathbf{I}_3$	$0 \cdot \mathbf{I}_3$	$5 \cdot \mathbf{I}_3$
Feedback-Based Control Only	$0 \cdot \mathbf{I}_3$	$1200 \cdot \mathbf{I}_3$	$5 \cdot \mathbf{I}_3$
Shared Autonomy	$100(1-\alpha) \cdot \mathbf{I}_3$	$1000\alpha \cdot \mathbf{I}_3$	$5 \cdot \mathbf{I}_3$

6.2. Human-based control

To demonstrate the performance of the human-based control, trials of operating the exoskeleton were done with no feedback-based control task, as if acceptance were always $\alpha = 0$. This was done with both the Positive Power Task and Force Output Task approaches for comparison. Under Positive Power the force to motion gain is $\mathbf{k}_e = 1.1$. Under Force Output the force to force gain is $\mathbf{k}_f = 4.0$. The privileged position task is the only active WBC task in this mode. The relative weights for each task is given in Table 2. During the trials, the user was allowed to move about freely. During this time they moved randomly at high and low speeds, and tried to track an approximate triangle trajectory. The actions are not exactly synced between the positive power and force output task trials, but occur in the same window of 1 min. Details of the exoskeleton motion are shown in Fig. 7. The first observation is that increased exoskeleton input forces correlate with increased velocities. The instantaneous power shown is

$$P = \dot{\boldsymbol{x}}_{ee} \cdot \boldsymbol{f}_e, \tag{36}$$

which is a measure of how well the exoskeleton's output motion correlates with the user's intention communicated though the force

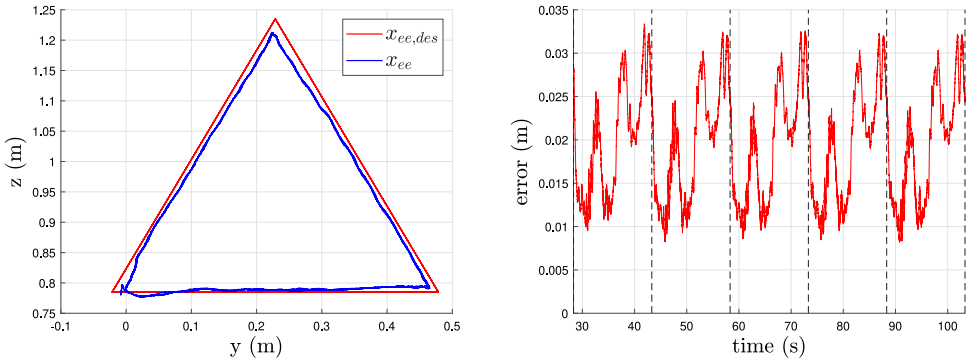


Fig. 6. (a) Desired and actual trajectory of the exoskeleton's end-effector under feedback-based control only. (b) Accuracy of the tracking task through this motion. Dotted black lines indicate the beginning of each cycle.

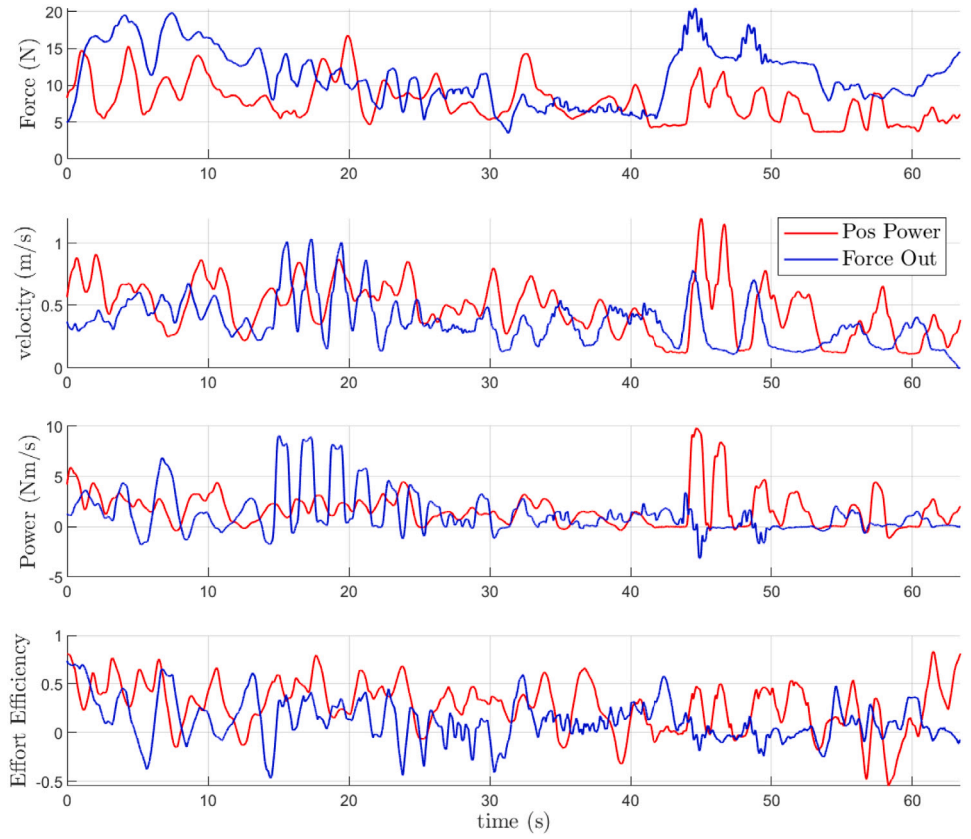


Fig. 7. User input force magnitude, left-arm end-effector velocity magnitude, instantaneous power over time, and a measure of the effort efficiency during free motion under the human-based task controller. For clarity, data is plotted under a moving average filter 333 time steps or 1 s. Data between trials is not synchronized as motions were not performed at the same rate.

input. During the Positive Power Task the average power is $P_{\text{PP,avg}} = 1.74$ Nm/s and the power is positive in 75.1% of timesteps. During the Force Output Task the average power is $P_{\text{FO,avg}} = 1.31$ Nm/s and the power is positive in 56.9% of timesteps. This is under the influence of noisy velocity and force sensor reading. On average, the positive instantaneous power indicates that the human-based task works to generate exoskeleton motions that follow the human's input directly, and thus will fulfill its role in the shared autonomy framework. Additionally, the higher average power and correlation percentage of the positive power task indicates that the task succeeds in its aim, increasing the power output towards the action the user wishes to take and therefore increasing the cooperation of the exoskeleton and the user.

To examine the power behavior further, it must be considered along with the magnitude of the Force and velocity. From Fig. 7, it can be seen that often the force output task requires a higher force input to achieve a similar or smaller resulting velocity compared to the positive power task. This is true in 40.0% of timesteps, while the reverse, the positive power task requiring a larger force for lesser resulting velocity, is only true in 9.0% of timesteps. This indicates that the positive power task also requires less effort from the user to achieve a desired motion. The varying magnitudes of force and velocity can make it harder to compare the performance of the two tasks, but it is still desired to compare how well both are performing. Therefore, we introduce an additional measure of how well the exoskeleton and human cooperate during human-based control, referred to as 'effort efficiency'. This measure

$$E = \frac{\dot{\boldsymbol{x}}_{\text{ee}}}{\|\dot{\boldsymbol{x}}_{\text{ee}}\|} \cdot \frac{\boldsymbol{f}_{\text{e}}}{\|\boldsymbol{f}_{\text{e}}\|}, \tag{37}$$

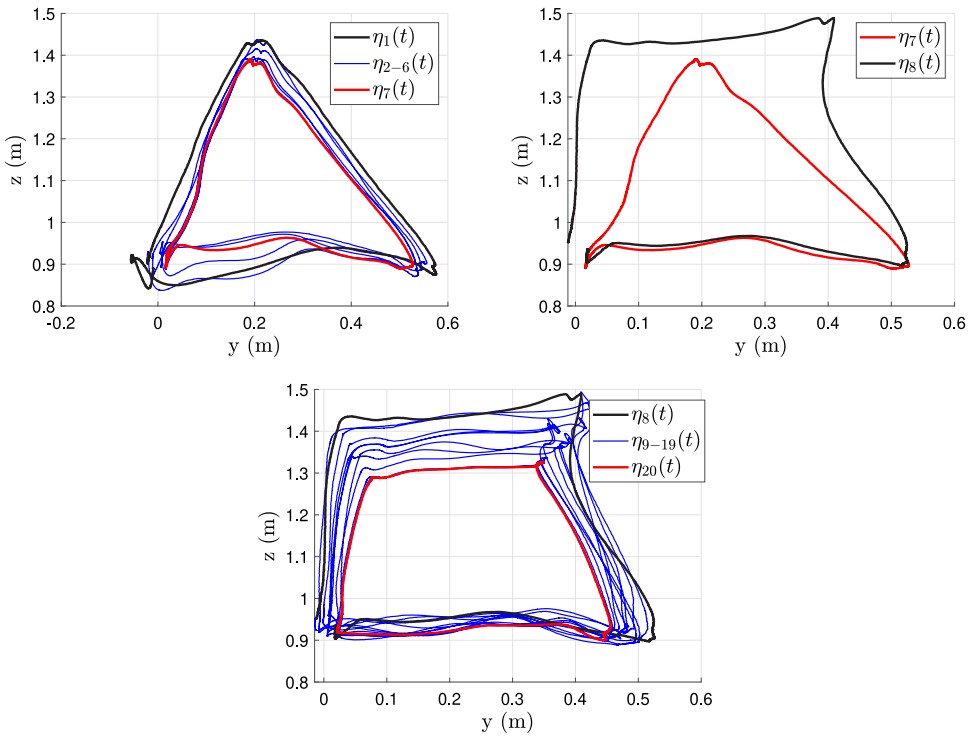


Fig. 8. Trajectory behavior during operation with the shared autonomy framework. (a) Highlights the taught trajectory $\boldsymbol{\eta}_1(t)$ and the final triangle trajectory $\boldsymbol{\eta}_7(t) = \boldsymbol{\phi}_{tri}(t)$ while showing each trajectory in between. (b) Shows the transition between the last triangle trajectory and first square trajectory. (c) Highlights the first $\boldsymbol{\eta}_8(t)$ and last square trajectory $\boldsymbol{\eta}_{20}(t) = \boldsymbol{\phi}_{squ}(t)$.

is the dot product of the unit vectors representing the direction of the force and velocity; essentially a non-dimensional measure $E \in [0, 1]$ indicating how well the robot and human are acting together. For both tasks, this is shown in Fig. 7. The percentage of time that the effort efficiency is positive is the same as the power, however, the fact that the measure is normalized allows it to be better compared between the two tasks. The average efficiency is $E_{\text{PP,avg}} = 0.278$ for the positive power task and $E_{\text{FO,avg}} = 0.125$ for the force output task. On average, the positive power task has a higher effort efficiency than the force output task does, and specifically this is true in 66.5% of timesteps. This new measure for exoskeleton control supports the claim that the positive power task generates more cooperation between the human and robot.

6.3. Shared autonomy control

After validating the performance of the individual control tasks, the shared autonomy framework combining them was implemented. Fig. 5 shows the user operating the exoskeleton. The controller gains used match the individual task implementations, with the task weights shown in Table 2. For measuring the acceptance, the learning rate is set $\lambda = 1.0$, and the normalization force value is $\beta = 33$ N. To begin the shared autonomy, the exoskeleton is initialized with a user-taught trajectory $\boldsymbol{\eta}_1(t)$. Then, operating with the soft-update strategy, the user attempts to track a simple desired trajectory $\boldsymbol{\phi}(t)$. The user starts with a triangle trajectory, similar to the trajectory tracking task in Section 6.1. The user corrects the exoskeleton continuously to adjust the trajectory over time from the initial until it converges from the initial trajectory $\boldsymbol{\eta}_1(t)$ to one they accept. Then, to test the responsiveness of the shared autonomy framework, the desired trajectory $\boldsymbol{\phi}(t)$ changes to be a square over the same period. The human begins to correct the trajectory to follow a square instead of a triangle and is able to do so quickly. After converging to an acceptable square trajectory, the acceptance increases once more as the human gives autonomy back to the exoskeleton to follow the taught square trajectory on its' own. Fig. 8 shows the trajectory behavior throughout this trial, while Table 3 gives the details of these trajectories varying over time.

For analysis purposes, the desired triangle trajectory $\boldsymbol{\phi}_{tri}(t)$ is set to be the final tracked triangle trajectory $\boldsymbol{\eta}_7(t)$. Similarly, the desired square trajectory $\boldsymbol{\phi}_{squ}(t)$ is set to be the final tracked square trajectory $\boldsymbol{\eta}_{20}(t)$. During trajectories 1–7, when a triangle is being tracked, it can be seen in the decreasing variance from $\boldsymbol{\phi}_{tri}(t)$ that the trajectories slowly converge to be closer to the desired. The acceptance begins high after the first trajectory, decreases initially due to the human exerting effort to correct the trajectory, and then increases again back up to $\alpha = 1.0$ as the trajectory approaches the desired triangle trajectory. When the desired trajectory changes to the square in trajectory 8, the exoskeleton responds immediately to the user's inputs, decreasing the acceptance and adjusting the tracked motion. We then see the same behavior twice more, of the acceptance decreasing while the user corrects the trajectory, and then increasing back to 1.0 once it has reached an acceptable trajectory. This happened twice in this case because the user settled to one trajectory, $\boldsymbol{\eta}_9(t)$, first and then continued to learn towards the final square trajectory $\boldsymbol{\eta}_{20}(t)$. It can be seen that in trajectories 6–7 and trajectories 18–20, once full acceptance has been reached, the exoskeleton repeats the same trajectory autonomously, as intended. The variance from the previous trajectory becomes as small as that between the trajectories under only feedback-based control. All together, the behavior displayed here verifies that the shared autonomy framework works to accomplish its' objective: allowing an exoskeleton to operate with less effort from the user while still being responsive to correction when the user's task objective changes.

Achieving the intended behavior, we look closer at the dynamics of the WBC that realizes it. Fig. 9 shows performance of the controller through the same trial. The figure first shows the tracking error of the three control tasks, not scaled by optimization weights or acceptance. Due to the tuned gains, the privileged position task has the smallest error, while the exoskeleton and human-based tasks have similar values for tracking error. It was found that having the feedback-based task tuned to have a slightly higher error was more effective for operation.

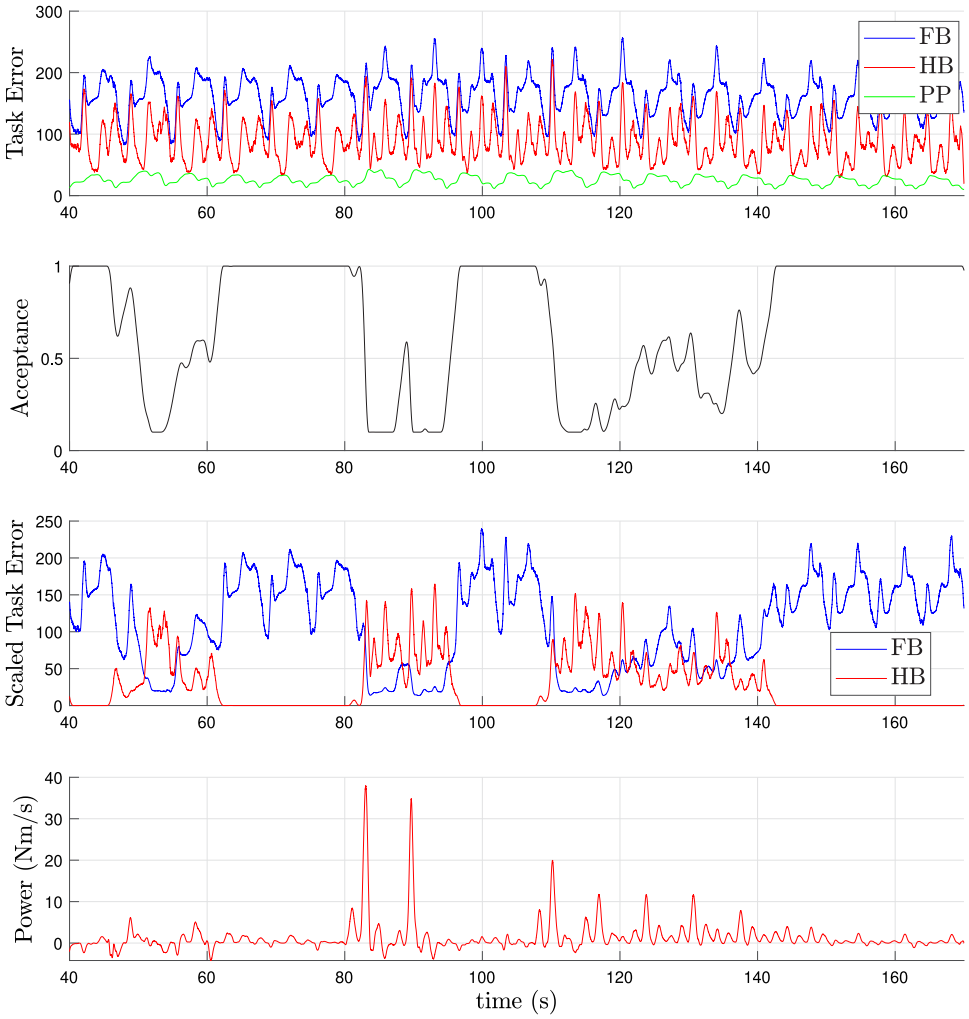


Fig. 9. The performance of the WBC over the course of the trial. The task error is the cost function associated with the feedback-based (FB, Eq. (18)), human-based (HB, Eq. (21)), and privileged position (PP, Eq. (26)) task controllers. The scaled task errors are scaled by the acceptance, as in Eq. (30). The power output at the end-effector matches Eq. (36). For clarity, data is plotted under a moving average filter of 167 time steps or 0.5 s.

Table 3
End-effector trajectories during Shared Autonomy. Variance is from each trajectory to the initial trajectory $\boldsymbol{\eta}_1(t)$, the previous trajectory $\boldsymbol{\eta}_{i-1}(t)$, the last triangle trajectory $\boldsymbol{\phi}_{tri}(t)$, or the last square trajectory $\boldsymbol{\phi}_{squ}(t)$.

Cycle #	Total Travel (m)	Var. From $\boldsymbol{\eta}_1(t)$ (mm)	Var. From $\boldsymbol{\eta}_{i-1}(t)$ (mm)	Average acceptance	Var. From $\boldsymbol{\phi}_{tri}(t)$ (mm)	Var. From $\boldsymbol{\phi}_{squ}(t)$ (mm)
1	2.454	0	0	0.00	1205	–
2	2.444	1562	1562	0.98	436	–
3	2.497	1571	243	0.55	522	–
4	2.465	1335	362	0.37	283	–
5	2.363	1235	202	0.86	96	–
6	2.346	1216	84	1.00	23	–
7	2.338	1205	23	1.00	0	–
8	2.645	1433	573	0.50	–	937
9	2.658	1358	399	0.20	–	846
10	2.517	1138	369	0.73	–	532
11	2.544	1137	104	1.00	–	476
12	2.544	1171	309	0.51	–	646
13	2.470	1154	265	0.18	–	642
14	2.349	1058	221	0.47	–	437
15	2.400	1052	135	0.41	–	365
16	2.240	1005	158	0.46	–	231
17	2.166	1014	170	0.95	–	71
18	2.144	1006	56	1.00	–	29
19	2.141	1011	26	1.00	–	12
20	2.120	1011	12	1.00	–	0

However, scaling these two tasks by the acceptance, as in Eq. (30), results in a closer balance between the two tasks. When acceptance is high, the error for the feedback-based task is dominant as the human-based task is negligible. The reverse is true when acceptance is low. A higher error here means that the WBC will prioritize tracking that task, and thus realize the behavior seen in Fig. 8. This is supported by the instantaneous power from the user, shown in Fig. 9, which is much higher when acceptance is low. This shows that the exoskeleton is successfully switching from tracking its' own trajectory with little variance when acceptance is high, to better cooperation with the user in accomplishing the task when acceptance is low.

For simplicity, in this trial we operated the shared autonomy framework with a learning rate of $\lambda = 1.0$ and a normalization force value of $\beta = 33$ N. These parameters are set relatively arbitrarily to show that the framework works, but it was found that changes in these parameters can drastically change the human's experience of operating the exoskeleton. Based on how the framework works, when a user wants to make a change in the exoskeleton's motion, they need to exert a large force at first to decrease the acceptance such that they can take authority. This is helpful because the small interaction forces that the user inputs while holding onto the handles of the robot, even when the exoskeleton is operating autonomously, do not decrease the acceptance or effect the functioning of the robot. However, the large force to "take control" does make it harder to start to direct the robot.

Lowering the normalization value β decreases this threshold for correcting the exoskeleton. For example, when we operate the same Shared Autonomy trial with $\beta = 22$ N, the user experience was that it took much less force to take authority back from the exoskeleton. A higher normalization value makes the exoskeleton require more force and be less responsive. The learning rate affects this too, but the users remarked more on the speed of acceptance change with a varied λ rather than the amount of force required. With a higher learning rate $\lambda = 1.25$ it was found that the acceptance changed much quicker and there was a higher proportion of time with acceptance at its minimum or maximum values rather than at an intermediate, shared autonomy value. With a lower learning rate $\lambda = 0.75$, the acceptance changed slower and the user had to be more conscious of the time it takes for the acceptance to increase once the exoskeleton has learned the proper objectives. In one trial with $\beta = 22$ N and $\lambda = 0.75$, an instability in the behavior was discovered where the lower learning rate slowed the acceptance changing, which required the user to maintain motions longer for the exoskeleton to learn. This was coupled with the lower normalization value such that a lower force input would correct the exoskeleton. However, the same force that changes the acceptance is used in the human-based task, so lower inputs will command less motion to the end effector, and thereby allowing the feedback-based task to be more dominant than the user wishes, especially if the acceptance is increasing slowly. Higher corrective forces will command more motion from the human-based task, but will then decrease the acceptance more, restarting the correction cycle. This is just an example of how a poor selection of parameters can negatively impact the user's experience. A parameter search or optimization is very hard to do as the proper tuning is also dependent on the user's behaviors and will likely vary between different individuals. A more thorough user study will be necessary to explore these interaction dynamics between the controller and the user.

The observed behavior of the system where a user must exert a large force to "take control" back from the exoskeleton is also key to the contribution of the shared autonomy framework. This force is not dedicated to the task completion, but rather to communicate the user's intention to the controller. This is a categorical difference from other exoskeleton controllers, because most controllers focus on making powered exoskeletons as intuitive and transparent as possible. Instead, here we propose a controller that the user must interact with directly, essentially including an interpretation layer with the controller that chooses how to respond to the user's input. This interaction and communication between the human and controller is complex, and it highlights the complex interaction between the independent agents of any powered exoskeleton and the human using it. Shared Autonomy, as presented here, provides the beginnings of a framework for separating these robot and human control interests and actions. Separating them is the first step towards controllers that allow a human and powered exoskeleton to cooperate in better ways.

7. Conclusion

In this paper we presented a Shared Autonomy framework that decomposes the behavior of a user operating a powered exoskeleton into a feedback-based and a human-based task. We use Whole-Body-Control as the approach to implementing these overlapping control objectives at the same time. We introduce 'acceptance' as a measure of how well the human's and exoskeleton's objectives match, as well as an approach for the exoskeleton to learn the human's objectives for the specific task of trajectory tracking. We present task-based positive power control as the human-based control approach both as an improvement on the state-of-the-art in multi-joint powered exoskeleton control, and as it is better able to be implemented in the WBC approach.

The framework's functionality is validated by implementation on a upper-body powered exoskeleton. Due to the complexity of the human–robot interaction, the necessary next step in understanding the resulting behavior of the framework is to conduct a thorough user study to examine it. This study will focus on not just the effect of parameters on performance, but also the ergonomic and cognitive experience of operating the exoskeleton. These types of effects of powered exoskeleton control are seldom explored. Overall, however, the results shown so far already indicate that the shared autonomy framework functions as intended. Tracking of desired trajectories is not perfect, but perfect performance is never reached with human-in-the-loop control. The ability to sense the environment, synthesize information, and make decisions to adjust during operation makes the inclusion of the framework worth the decrease in control accuracy. The framework augments these strengths by allowing the controller to be relied upon for assistance in completing repetitive tasks directly, rather than simply providing physical assistance in certain motions. Finally, in this work, a no-load trajectory tracking task was the only task implemented in the shared autonomy framework. Future work will include implementing more tasks, including a load-lifting task and a bimanual manipulation task. To implement any new task, we must only implement an autonomous controller to accomplish the type of task, and a learning method for the exoskeleton to infer the user's task objectives. With these two steps, the presented control framework can be implemented for any industrial task or controller.

CRediT authorship contribution statement

Benjamin Beiter: Developed the ideas for this work, implemented it, and was the primary author for this paper. **Divya Srinivasan:** Contributed to the overall concept development and advised on the experimental validation approaches. **Alexander Leonessa:** Advised on all aspects of this work, assisted in developing ideas and the writing this paper.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Benjamin Beiter reports financial support was provided by National Science Foundation.

Data availability

Data will be made available on request.

References

[1] R. Bogue, Robotic exoskeletons: A review of recent progress, Ind. Robot: Int. J. 42 (1) (2015) 5–10, http://dx.doi.org/10.1108/ir-08-2014-0379.

[2] A. Zaroug, J.K. Proud, D.T.H. Lai, K. Mudie, D. Billing, R. Begg, Overview of Computational Intelligence (CI) techniques for powered exoskeletons, in: Computational Intelligence in Sensor Networks, Springer, Berlin Heidelberg, 2019, pp. 353–383, http://dx.doi.org/10.1007/978-3-662-57277-1_15.

[3] A.J. Young, D.P. Ferris, State of the art and future directions for lower limb robotic exoskeletons, IEEE Trans. Neural Syst. Rehabil. Eng. 25 (2) (2017) 171–182, http://dx.doi.org/10.1109/tnsre.2016.2521160.

[4] W. Chen, J. Li, S. Zhu, X. Zhang, Y. Men, H. Wu, Gait recognition for lower limb exoskeletons based on interactive information fusion, Appl. Bionics Biomech. 2022 (2022) 1–19, http://dx.doi.org/10.1155/2022/9933018.

[5] A.M. Stewart, C.G. Pretty, M. Adams, X. Chen, Review of upper limb hybrid exoskeletons, in: 2017 International Federation of Automatic Control.

[6] M.A. Gull, S. Bai, T. Bak, A review on design of upper limb exoskeletons, Robotics 9 (1) (2020) 16, http://dx.doi.org/10.3390/robotics9010016.

[7] L. Lang, J. Xiao, Y. Sun, H. Lu, Z. Zhou, C. Yang, Scale force control of an exoskeleton for human performance augmentation, J. Intell. Robot. Syst. 106 (1) (2022) http://dx.doi.org/10.1007/s10846-022-01611-6.

[8] D. Lim, W. Kim, H. Lee, H. Kim, K. Shin, T. Park, J. Lee, C. Han, Development of a lower extremity exoskeleton robot with a quasi-anthropomorphic design approach for load carriage, in: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, IEEE, http://dx.doi.org/10.1109/iros.2015.7354132.

[9] A. Chu, H. Kazerooni, A. Zoss, On the biomimetic design of the Berkeley Lower Extremity Exoskeleton (BLEEX), in: 2005 IEEE International Conference on Robotics and Automation, IEEE, http://dx.doi.org/10.1109/robot.2005.1570789.

[10] S. Kim, D. Srinivasan, M.A. Nussbaum, A. Leonessa, Human gait during level walking with an occupational whole-body powered exoskeleton: Not yet a walk in the park, IEEE Access 9 (2021) 47901–47911, http://dx.doi.org/10.1109/access.2021.3068836.

[11] S. Robotics, Guardian XO full-body powered exoskeleton, 2022, January 18 2023, URL https://www.sarcos.com/products/guardian-xo-powered-exoskeleton/.

[12] S. Marcheschi, F. Salsedo, M. Fontana, M. Bergamasco, Body extender: Whole body exoskeleton for human power augmentation, in: 2011 IEEE International Conference on Robotics and Automation, IEEE, http://dx.doi.org/10.1109/icra.2011.5980132.

[13] S. De Bock, J. Ghillebert, R. Govaerts, B. Tassignon, C. Rodriguez-Guerrero, S. Crea, J. Veneman, J. Geeroms, R. Meeusen, K. De Pauw, Benchmarking occupational exoskeletons: An evidence mapping systematic review, Appl. Ergon. 98 (2022) 103582, http://dx.doi.org/10.1016/j.apergo.2021.103582, URL https://www.ncbi.nlm.nih.gov/pubmed/34600307.

[14] M.P. De Looze, T. Bosch, F. Krause, K.S. Stadler, L.W. O'Sullivan, Exoskeletons for industrial application and their potential effects on physical work load, Ergonomics 59 (5) (2016) 671–681, http://dx.doi.org/10.1080/00140139.2015.1081988, URL https://ulir.ul.ie/bitstream/10344/5646/1/OSullivan_2016_exoskeletons.pdf.

[15] H.S. Lo, S.Q. Xie, Exoskeleton robots for upper-limb rehabilitation: State of the art and future prospects, Med. Eng. Phys. 34 (3) (2012) 261–268, http://dx.doi.org/10.1016/j.medengphy.2011.10.004, URL https://www.ncbi.nlm.nih.gov/pubmed/22051085.

[16] J. Gunasekara, R. Gopura, T. Jayawardane, S. Lalitharathne, Control methodologies for upper limb exoskeleton robots, in: 2012 IEEE/SICE International Symposium on System Integration, SII, IEEE, http://dx.doi.org/10.1109/sii.2012.6427387.

[17] S. Dalla Gasperina, L. Roveda, A. Pedrocchi, F. Braghin, M. Gandolla, Review on patient-cooperative control strategies for upper-limb rehabilitation exoskeletons, Front. Robot. AI 8 (2021) 745018, http://dx.doi.org/10.3389/frobt.2021.745018, URL https://www.ncbi.nlm.nih.gov/pubmed/34950707.

[18] R.A.R.C. Gopura, K. Kiguchi, D.S.V. Bandara, A brief review on upper extremity robotic exoskeleton systems, in: 2011 6th International Conference on Industrial and Information Systems, IEEE, http://dx.doi.org/10.1109/iciinfs.2011.6038092.

[19] K. Anam, A.A. Al-Jumaily, Active exoskeleton control systems: State of the art, Procedia Eng. 41 (2012) 988–994, http://dx.doi.org/10.1016/j.proeng.2012.07.273.

[20] Q. Wu, X. Wang, B. Chen, H. Wu, Patient-active control of a powered exoskeleton targeting upper limb rehabilitation training, Front. Neurol. 9 (2018) 817, http://dx.doi.org/10.3389/fneur.2018.00817, URL https://www.ncbi.nlm.nih.gov/pubmed/30364274.

[21] S. Bai, S. Christensen, M.R.U. Islam, An upper-body exoskeleton with a novel shoulder mechanism for assistive applications, in: 2017 IEEE International Conference on Advanced Intelligent Mechatronics, AIM, IEEE, http://dx.doi.org/10.1109/aim.2017.8014156.

[22] C. Liang, T. Hsiao, Admittance control of powered exoskeletons based on joint torque estimation, IEEE Access 8 (2020) 94404–94414, http://dx.doi.org/10.1109/access.2020.2995372.

[23] H. Mehdi, O. Boubaker, Stiffness and impedance control using Lyapunov theory for robot-aided rehabilitation, Int. J. Soc. Robot. 4 (S1) (2012) 107–119, http://dx.doi.org/10.1007/s12369-011-0128-5.

[24] H. Krebs, M. Ferraro, S.P. Buerger, M.J. Newbery, A. Makiyama, M. Sandmann, D. Lynch, B.T. Volpe, N. Hogan, Rehabilitation robotics: Pilot trial of a spatial extension for MIT-manus, J. NeuroEng. Rehabil. 1 (1) (2004) 5, http://dx.doi.org/10.1186/1743-0003-1-5.

[25] G. Aguirre-Ollinger, J.E. Colgate, M.A. Peshkin, A. Goswami, Active-impedance control of a lower-limb assistive exoskeleton, in: 2007 IEEE 10th International Conference on Rehabilitation Robotics, IEEE, http://dx.doi.org/10.1109/icorr.2007.4428426.

[26] T. Lenzi, S.M.M. De Rossi, N. Vitiello, M.C. Carrozza, Proportional EMG control for upper-limb powered exoskeletons, in: 2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society, IEEE, http://dx.doi.org/10.1109/iembs.2011.6090139.

[27] K. Kong, M. Tomizuka, Control of exoskeletons inspired by fictitious gain in human model, IEEE/ASME Trans. Mechatronics 14 (6) (2009) 689–698, http://dx.doi.org/10.1109/tmech.2009.2032685.

[28] L. Peternel, T. Noda, T. Petrič, A. Ude, J. Morimoto, J. Babič, Adaptive control of exoskeleton robots for periodic assistive behaviours based on EMG feedback minimisation, PLoS One 11 (2) (2016) e0148942, http://dx.doi.org/10.1371/journal.pone.0148942, URL http://dirros.openscience.si/Dokument.php?id=1030&dn=.

[29] K. Kiguchi, Y. Hayashi, An EMG-based control for an upper-limb power-assist exoskeleton robot, IEEE Trans. Syst. Man Cybern. B 42 (4) (2012) 1064–1071, http://dx.doi.org/10.1109/tsmcb.2012.2185843.

[30] J. Wang, O.R. Barry, Inverse optimal robust adaptive controller for upper limb rehabilitation exoskeletons with inertia and load uncertainties, IEEE Robot. Autom. Lett. 6 (2) (2021) 2171–2178, http://dx.doi.org/10.1109/lra.2021.3061361.

[31] A.M. Khan, D.-W. Yun, M.A. Ali, J. Han, K. Shin, C. Han, Adaptive impedance control for upper limb assist exoskeleton, in: 2015 IEEE International Conference on Robotics and Automation, ICRA, IEEE, http://dx.doi.org/10.1109/icra.2015.7139801.

[32] S. Mghames, C.D. Santina, M. Garabini, A. Bicchi, A neuromuscular-model based control strategy to minimize muscle effort in assistive exoskeletons, IEEE Int. Conf. Rehabil. Robot. 2019 (2019) 963–970, http://dx.doi.org/10.1109/ICORR.2019.8779456, URL https://www.ncbi.nlm.nih.gov/pubmed/31374754.

[33] S. Bustamante, G. Quere, K. Hagmann, X. Wu, P. Schmaus, J. Vogel, F. Stulp, D. Leidner, Toward seamless transitions between shared control and supervised autonomy in robotic assistance, IEEE Robot. Autom. Lett. 6 (2) (2021) 3833–3840, http://dx.doi.org/10.1109/lra.2021.3064449.

[34] A. Mörtl, M. Lawitzky, A. Kucukyilmaz, M. Sezgin, C. Basdogan, S. Hirche, The role of roles: Physical cooperation between humans and robots, Int. J. Robot. Res. 31 (13) (2012) 1656–1674, http://dx.doi.org/10.1177/0278364912455366, URL http://eprints.lincoln.ac.uk/31162/9/Moertl-IJRR12-RoleOfRoles.pdf.

[35] D.P. Losey, C.G. Mcdonald, E. Battaglia, M.K. O'Malley, A review of intent detection, arbitration, and communication aspects of shared control for physical human–robot interaction, Appl. Mech. Rev. 70 (1) (2018) 010804, http://dx.doi.org/10.1115/1.4039145, URL https://appliedmechanicsreviews.asmedigitalcollection.asme.org/data/journals/amread/936715/amr_070_01_010804.pdf.

[36] M.A. Hopkins, D.W. Hong, A. Leonessa, Compliant locomotion using whole-body control and divergent component of motion tracking, in: 2015 IEEE International Conference on Robotics and Automation, ICRA, IEEE, http://dx.doi.org/10.1109/icra.2015.7140001.

[37] T. Koolen, S. Bertrand, G. Thomas, T. de Boer, T. Wu, J. Smith, J. Englsberger, J. Pratt, Design of a momentum-based control framework and application to the humanoid robot Atlas, Int. J. Humanoid Robotics 13 (01) (2016) http://dx.doi.org/10.1142/s0219843616500079.

[38] R.J. Griffin, G. Wiedebach, S. Bertrand, A. Leonessa, J. Pratt, Straight-leg walking through underconstrained whole-body control, in: 2018 IEEE International Conference on Robotics and Automation, ICRA, IEEE, http://dx.doi.org/10.1109/icra.2018.8460751.

[39] L. Penco, B. Clement, V. Modugno, E. Mingo Hoffman, G. Nava, D. Pucci, N.G. Tsagarakis, J.B. Mouret, S. Ivaldi, Robust real-time whole-body motion retargeting from human to humanoid, in: 2018 IEEE-RAS 18th International Conference on Humanoid Robots, Humanoids, IEEE, http://dx.doi.org/10.1109/humanoids.2018.8624943.

[40] D. Mronga, S. Kumar, F. Kirchner, Whole-body control of series-parallel hybrid robots, in: 2022 International Conference on Robotics and Automation, ICRA, IEEE, http://dx.doi.org/10.1109/icra46639.2022.9811616.

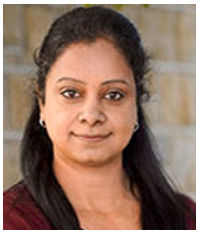
[41] M. Iskandar, G. Quere, A. Hagengruber, A. Dietrich, J. Vogel, Employing whole-body control in assistive robotics, in: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, IEEE, http://dx.doi.org/10.1109/iros40897.2019.8967772.

[42] B. Beiter, C. Herron, A. Leonessa, Whole body control for haptic interaction with VR, in: 2022 American Control Conference, ACC, IEEE, http://dx.doi.org/10.23919/acc53348.2022.9867505.

[43] I. Robotics, Open Robotics Software (ORS) source code, 2022, December 5th 2022, URL https://github.com/ihmcrobotics/ihmc-open-robotics-software.

[44] I. Robotics, Simulation Construction Set (SCS) source code, 2022, December 5th 2022, URL https://github.com/ihmcrobotics/simulation-construction-set.
[45] J. Smith, D. Stephen, A. Lesman, J. Pratt, Real-time control of humanoid robots using OpenJDK, in: 2014 Proceedings of the 12th International Workshop on Java Technologies for Real-Time and Embedded Systems, JTRES '14, ACM Press, http://dx.doi.org/10.1145/2661020.2661027.
[46] C.W. Herron, Z.J. Fuge, M. Kogelis, N.J. Tremaroli, B. Kalita, A. Leonessa, Design and validation of a low-level controller for hierarchically controlled exoskeletons, Sensors 23 (2) (2023) 1014, http://dx.doi.org/10.3390/s23021014.
[47] S. Welch, C. Runyon, B. Beiter, C. Herron, B. Kalita, A. Leonessa, A mapping approach for torque control of parallel-actuated systems, in: 2022 International Mechanical Engineering Congress and Exposition.
[48] R. Featherstone, Rigid Body Dynamics Algorithms, Springer, 2014.

Ben Beiter is a Ph.D. student in mechanical engineering at Virginia Tech. Research interests center primarily around controls for powered exoskeletons and humanoid robots, which extends broadly to the integration of sophisticated robotic platforms into the modern world. This includes developments in dynamics and control theory, implementation and tuning of robotic systems, human–robot interaction and cooperation, and bioinspired robots for locomotion and interaction.



Dr. Divya Sinivasan is a Professor of Industrial Engineering and Bioengineering at Clemson University. Research interests include human performance assessments (strength, motion, postures, individual differences), biomechanics and movement control, and motor learning and skills training. Dr. Srinivasan heads the Biomechanics, Ergonomics, Safety, and Training (BEST) Lab (https://cecas.clemson.edu/ergo/).

Dr. Alexander Leonessa is a Professor of Mechanical Engineering and Virginia Tech, and is the Associate Department Head for Strategic Initiatives. Research Interests include control theory and robotics, autonomous ground, aerial, surface, and underwater vehicles, rehabilitation robotics, brain–computer interfaces, and sensory-motor neuroprosthesis, and augmented and virtual reality interfaces, exoskeletons, and haptic feedback. Dr. Leonessa heads the Terrestrial Robotics Engineering and Controls (TREC) Lab (https://www.trecvt.org/).