



Computational modeling and simulation of closed chain arm-robot multibody dynamic systems in OpenSim

Matthew Green¹ · Yoon No Gregory Hong² · Jinsook Roh² · Benjamin J. Fregly¹ 

Received: 16 January 2022 / Accepted: 4 October 2022 / Published online: 7 November 2022
© The Author(s) 2022

Abstract

Rehabilitation robot efficacy for restoring upper extremity function post-stroke could potentially be improved if robot control algorithms accounted for patient-specific neural control deficiencies. As a first step toward the development of such control algorithms using model-based methods, this study provides general guidelines for creating and simulating closed chain arm-robot models in the OpenSim environment, along with a specific example involving a three-dimensional arm moving within a two degree-of-freedom upper extremity rehabilitation robot. The closed chain arm-robot model developed in OpenSim was evaluated using experimental robot motion and torque data collected from a single healthy subject under four conditions: 1) active robot alone, 2) active robot with passive arm, 3) passive robot with active arm, and 4) active robot with active arm. Computational verification of the combined model was performed for all four conditions, whereas experimental validation was performed for only the first two conditions since torque measurements were not available for the arm. For the four verification problems, forward dynamic simulations reproduced experimentally measured robot joint angles with average root-mean-square (RMS) errors of less than 0.3 degrees and correlation coefficients of 1.00. For the two validation problems, inverse dynamic simulations reproduced experimentally measured robot motor torques with average RMS errors less than or equal to 0.5 Nm and correlation coefficients between 0.92 and 0.99. If patient-specific muscle–tendon and neural control models can be successfully added in the future, the coupled arm-robot OpenSim model may provide a useful testbed for designing patient-specific robot control algorithms that facilitate recovery of upper extremity function post-stroke.

Keywords Rehabilitation robotics · Musculoskeletal modeling · Neuromusculoskeletal modeling · OpenSim · Kinarm · Stroke neurorehabilitation

1 Introduction

Strokes are one of the most prevalent serious health conditions in society today with one in four people over the age of 25 experiencing a stroke in their lifetime [1]. Eighty percent

✉ B.J. Fregly
fregly@rice.edu; url: <https://rcnl.rice.edu>

¹ Department of Mechanical Engineering, Rice University, Houston, TX, USA

² Department of Biomedical Engineering, University of Houston, Houston, TX, USA

of strokes lead to hemiparesis of the contralateral upper limb with 40% becoming chronic [2]. Hemiparesis can significantly affect a patient's ability to perform activities of daily living by limiting the ability to reach and grasp [3, 4]. This fact helps explain why stroke is a leading cause of long-term disability [5], with over 50% of stroke survivors aged 18 to 64 years reporting a stroke-related work disability [6]. Although the high prevalence and serious consequences of stroke have motivated extensive research seeking to develop more effective treatments, the statistics above indicate that existing clinical interventions are often unable to restore lost arm and hand function to the extent that patients desire.

Rehabilitation robotics is a growing research area that is seeking to fill the gap between what existing clinical treatments can achieve and the amount of upper extremity function that patients would like to recover [7]. Rehabilitation robots utilize a wide range of different assistive cooperative control algorithms that fall into three general categories: passive control, triggered passive control, and partially assistive control [8]. Partially assistive control, or assist-as-needed control, is the most common and seeks to help the patient complete the desired motion while requiring the most effort possible from the patient [9]. When calculating the necessary robot control torques, many assist-as-needed controllers are complex enough to account for dynamic contributions from the subject's arm, but none to date account for patient-specific neural control deficits or desired neural control changes. Compared to patients receiving conventional therapy, those receiving upper extremity robotic therapy do not generally exhibit larger improvements in their ability to perform activities of daily living [10, 11]. While the reasons for this lack of effectiveness remain unclear, the uniqueness of each patient's neural control deficits and the generic nature of existing rehabilitation robot control methods are likely contributing factors [12–14].

One way to account for the uniqueness of each patient's neural control deficits is through the use of personalized neuromusculoskeletal models. Coupling such models with rehabilitation robot models could facilitate the use of model-based methods for designing patient-specific robot control algorithms. Personalized neuromusculoskeletal models have recently matured to the point where they can represent patient-specific neural control characteristics post-stroke and predict a patient's function under new conditions [15, 16]. Furthermore, generic neuromusculoskeletal models have already proven to be effective tools to support the design of exoskeleton robots [17–19] and end-effector robot control systems [20]. However, coupling a personalized neuromusculoskeletal model with a rehabilitation robot model creates closed kinematic chains that are challenging to model and simulate [19–21] with the commonly-used musculoskeletal modeling software OpenSim [22, 23]. Although three previous studies have coupled some form of upper extremity musculoskeletal model with some form of robot model [18, 19, 27], none of these studies provided general guidelines for modeling and simulating the resulting closed kinematic chains within the OpenSim environment, and none performed extensive computational verification and experimental validation of their coupled arm-robot models.

As a first step toward designing patient-specific robot control algorithms for upper extremity neurorehabilitation post-stroke using model-based methods, this study addresses the challenges involved in creating and simulating coupled arm-robot models possessing multiple closed kinematic chains using OpenSim. These challenges include determining the number of degrees of freedom in a closed chain model, constructing a closed chain model using available OpenSim constraints, and simulating a closed chain model using available OpenSim dynamic simulation methods. Specifically, after presenting general guidelines for how to construct and simulate multibody dynamic models possessing closed kinematic chains within the OpenSim environment, we present a specific example involving the Kinarm (Kinarm Corporation, Kingston, ON, Canada) upper extremity rehabilitation robot. Creation of

a three-dimensional musculoskeletal arm-Kinarm robot model in OpenSim required modeling not only the closed kinematic chain involving the arm and robot but also multiple closed kinematic chains within the robot. The coupled arm-robot OpenSim model was verified computationally and validated experimentally using data collected from a single subject performing shoulder–elbow reaching tasks within the Kinarm robot, confirming the reliability and utility of the proposed general guidelines. If patient-specific muscle–tendon and neural control models can be successfully added in the future, then the final arm-robot OpenSim model may eventually provide a useful testbed for designing patient-specific rehabilitation robot control algorithms that facilitate recovery of upper extremity function post-stroke.

2 General guidelines

Before describing the specific arm-robot model application, we begin by providing helpful general guidelines for constructing and simulating multibody dynamic models possessing closed kinematic chains within the OpenSim environment. These general guidelines are intended to provide guidance to researchers who wish to perform similar closed chain model construction and simulation tasks to those presented here but using a different robot than the Kinarm.

Rigid body dynamic systems can be categorized as either “open chain” or “closed chain.” The most common category is an open-chain system, where one end of the system is attached to a base body by a joint while the other end of the system is completely unconstrained. In contrast, a closed-chain system arises when both ends of the system are attached to the same base body by a joint. Closed-chain systems are more difficult to analyze kinematically and dynamically since the additional joint at the far end of the system constrains (or limits) the achievable configurations of the system. In an open-chain system, all of the generalized coordinates defining the configuration of the system are independent of one another. However, in a closed-chain system, the constraints imposed by the additional joint make some of the system’s generalized coordinates dependent on the values of the other generalized coordinates. In this case, the values of the dependent generalized coordinates are no longer required to determine the configuration of the system but rather can be found from the values of the independent generalized coordinates.

2.1 Closed-chain model degrees of freedom

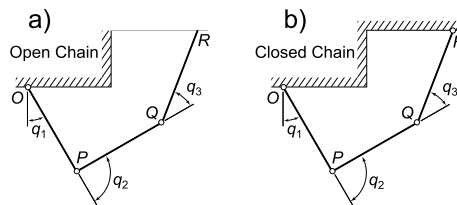
The key to developing a rigid body dynamic model with one or more closed kinematic chains is determining first, how many independent degrees of freedom (DOFs) the model should have, and second, which types of joints should be used throughout the model to achieve the desired number of independent DOFs. The answer to the first question is determined subjectively based on knowledge of the system being modeled. In contrast, the answer to the second question can be determined objectively based on knowledge of some basic mathematical relationships. Specifically, the number of DOFs n in a rigid body dynamic model without or with closed kinematic chains can be calculated for a two-dimensional (2D) or three-dimensional (3D) system by knowing the number of rigid bodies b in the model and the number of constraints c imposed by joints in the model:

$$n = 3b - c \quad \text{for a 2D system,} \quad (1)$$

$$n = 6b - c \quad \text{for a 3D system,} \quad (2)$$

Table 1 Summary of constraints for different types of joints in two-dimensional (2D) and three-dimensional (3D) rigid body dynamic models

Dimensions	Joint type	Translational constraints	Rotational constraints	Total number of constraints	Total number of DOFs
2D	Weld	2	1	3	0
	Pin	2	0	2	1
	Slider	1	1	2	1
	Planar	0	0	0	3
3D	Weld	3	3	6	0
	Pin	3	2	5	1
	Slider	2	3	5	1
	Universal	3	1	4	2
	Cylindrical	2	2	4	2
	Gimbal	3	0	3	3
	Planar	1	2	3	3
	Ball	3	0	3	3
	Bearing	2	0	2	4
6-DOF	0	0	0	6	

**Fig. 1** a) Open- and b) closed-chain planar kinematic models composed of three rigid links connected by pin joints. The open-chain model possesses 3 DOFs, whereas the closed-chain model possesses 1 DOF. Points O , P , and Q are locations of pin joints for both models, whereas point R is the added pin joint for the closed-chain model. Generalized coordinates q_1 , q_2 , and q_3 are independent for the open-chain model, whereas any of the three coordinates can be chosen as the one independent coordinate for the closed-chain model

where the ground body is never included in the number of bodies. For a 2D system, each unconstrained rigid body with no joints possesses 3 DOFs, whereas for a 3D system, each unconstrained rigid body possesses 6 DOFs. Adding joints to a body constrains the body's motion and reduces these initial numbers of DOFs. The number of constraints provided by each joint is determined by the translations and rotations that the joint does not permit. For example, a pin joint does not allow 2 translations for a 2D system, whereas for a 3D system, it does not allow 3 translations and 2 rotations. For reference, the number of constraints provided by different types of common joints are listed in Table 1.

Some simple examples are helpful for demonstrating how to calculate the number of DOFs for 2D and 3D open- and closed-chain rigid body dynamic models. Consider first a planar open-chain three-bar linkage, where the first body is connected to ground by a pin joint, the second body to the first body by a pin joint, and the third body to the second body by a pin joint (Fig. 1a), with all pin joint axes being perpendicular to the plane of motion. Thus this model possesses three bodies and three pin joints. If we consider the model to be

a 2D system, then the number of DOFs is found from Eq. (1):

$$n = 3(3) - 2(3) = 3.$$

Since each pin joint in a 2D system prevents 2 translations, the 3 pin joints impose 6 constraints on the motion of the system. Similarly, if we consider the model to be a 3D system, then the number of DOFs is found from Eq. (2):

$$n = 6(3) - 5(3) = 3.$$

In a 3D system, each pin joint prevents 3 translations and 2 rotations, and thus 3 pin joints impose 15 constraints. Either way, Eqs. (1) and (2) indicate that the model possesses 3 DOFs, as one would expect.

A less obvious result occurs when the open-chain three-bar linkage is turned into a closed-chain four-bar linkage by pinning the far end of the chain to the ground, where the axis of the new pin joint is again perpendicular to the plane of motion (Fig. 1b). If we consider the resulting model to be a 2D system, then the number of DOFs is again found from Eq. (1):

$$n = 3(3) - 2(4) = 1.$$

Thus “closing the chain” with an additional pin joint yields a model with only 1 DOF, as one would expect. However, if the model is considered to be a 3D system, then the number of DOFs found from Eq. (2) becomes

$$n = 6(3) - 5(4) = -2.$$

Though initially surprising, a negative number of DOFs indicates that the model is overconstrained and may lock up or not assemble successfully. Since the desired number of DOFs is 1 for a four-bar linkage, -2 DOFs indicate that the model possesses 3 excess constraints. To eliminate 3 redundant constraints, one can replace the additional pin joint possessing 5 constraints with a bearing joint possessing only 2 translational constraints. The bearing joint constrains the far end of the three-bar linkage the same way as in the 2D case but without adding a redundant out-of-plane reaction force and two redundant in-plane reaction moments. With this change, the resulting number of DOFs becomes

$$n = 6(3) - [5(3) + 2(1)] = 1,$$

as expected.

2.2 Closed-chain model construction in OpenSim

These theoretical concepts provide the foundation needed for constructing coupled arm-robot models in OpenSim, where the primary challenge is how to model closed kinematic chains. Overcoming this challenge requires understanding two additional OpenSim concepts. The first concept involves the difference between joints and constraints in OpenSim. Joints are used to create open kinematic chains by connecting individual bodies together. Joints in OpenSim follow a parent–child relationship, where every parent body can be connected by a joint to multiple child bodies (i.e., a parent can have multiple children), but every child body can be connected by a joint to only one parent body (i.e., a child can

have only one parent). In contrast, constraints are used to create closed kinematic chains by connecting one body in an open chain to another body in the same chain. For a constraint, there is no concept of a parent or child body but rather only two bodies. OpenSim supports three types of constraints that can be used to close a kinematic chain: `PointConstraint`, `PointOnLineConstraint`, and `WeldConstraint`. Each of these constraints can be viewed as a replacement for a corresponding joint. A `PointConstraint` imposes the same constraints as a pin (2D) or gimbal (3D) joint, a `PointOnLineConstraint` the same constraints as a pin+slider (2D) or bearing (3D) joint, and a `WeldConstraint` the same constraints as a weld (2D or 3D) joint.

The second concept involves deciding where to place a constraint to close a kinematic chain in OpenSim. Multiple options exist for where a constraint can be used to replace a joint and close the chain. When choosing the location of the constraint, one should keep in mind four recommendations to maximize the likelihood that the model will assemble and simulate properly. First, each closed kinematic chain should, if possible, possess only one constraint. Second, each body should possess at most one constraint. Third, no constraints should be used where coordinate actuators, prescribed motions, or measured orientations would be used for the equivalent joint. Fourth, prescribed motions should be applied to only as many joints as independent DOFs in the closed-chain model, though coordinate actuators can be applied to as many joints as desired. Because of these four recommendations, the locations of constraints must be chosen strategically when constructing an OpenSim model with closed kinematic chains. Once a closed-chain model is constructed in OpenSim, the number of DOFs should be calculated from Eq. (1) or (2), where the number of constraints imposed by each OpenSim constraint should be added to the number of constraints. The calculated number of DOFs should then be compared to the desired number of DOFs to verify that the selected joints and constraints function as intended.

2.3 Closed-chain model simulation in OpenSim

This theoretical background informs not only closed-chain model construction in OpenSim but also closed-chain model dynamic simulations in OpenSim. Dynamic simulations fall into two general categories, “forward” and “inverse.” Performing forward dynamic simulations with a closed-chain model is straightforward in OpenSim, since one can place coordinate actuators on as many joints as desired without overconstraining the model. In contrast, performing inverse dynamic simulations with a closed-chain model is more complicated in OpenSim. In closed-chain models, the coordinates can be partitioned into independent and dependent sets, where the number of independent coordinates equals the number of DOFs in the model. During an inverse dynamic analysis, OpenSim calculates an inverse dynamic load for every coordinate in the model, even unactuated dependent coordinates in closed kinematic chains. To force OpenSim to lump inverse dynamic loads at only the selected independent coordinates, one can perform a “mixed” dynamic simulation that combines forward and inverse dynamics in a single simulation [24]. A mixed dynamic simulation is formulated in OpenSim by including prescribed coordinates (which are differentiated twice with respect to time automatically by OpenSim) in a forward dynamic simulation, which causes an associated OpenSim joint reaction analysis to calculate inverse dynamic loads for the prescribed coordinates. When a model possesses closed kinematic chains, prescribing the motion of the independent coordinates in a forward dynamic simulation produces inverse dynamic loads for only those coordinates, with no inverse dynamic loads being calculated for the dependent coordinates. Consequently, by changing which coordinates are prescribed one can control which coordinates are treated as independent and thus where OpenSim calculates inverse dynamic loads from a forward dynamic simulation. Furthermore, coordinate

Table 2 Summary of experimental conditions. Active indicates that control torques were provided by the robot or arm, whereas passive indicates that no control torques were provided robot or arm

Condition	Robot	Arm
1	Active	None
2	Active	Passive
3	Passive	Active
4	Active	Active

actuators applied to dependent coordinates will affect the calculated inverse dynamic loads for the independent coordinates.

3 Specific example

These general guidelines were used to construct and simulate a coupled arm-robot model for the Kinarm robot, where simulations were performed to verify and validate the proposed modeling approaches.

3.1 Experimental data collection

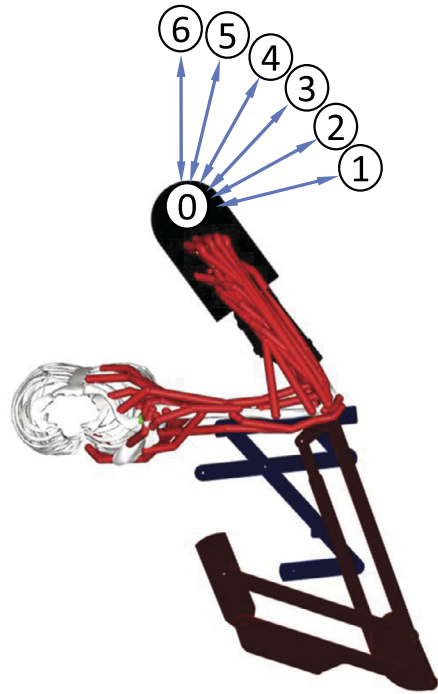
A Kinarm exoskeleton robot (Kinarm Corporation, Kingston, ON, Canada) was utilized for all experimental and modeling tasks performed in this study. The Kinarm robot possesses 2 DOFs in the transverse plane, fully supports the weight of the arm, and uses multiple closed kinematic chains to reproduce anatomical shoulder and elbow motion. For test conditions where the robot was active (see below), the robot was used in position control mode. During experiments, the robot reports its two commanded motor torques (accurate to within 5%, or 0.2 Nm for our experiments), estimates of the two associated frictional torques, and the two associated motor angles (accurate to within 0.2°).

Experimental movement data were collected under four conditions (Table 2) using a Kinarm robot in position control mode without and with the right arm of a single healthy male subject (age 23 yrs, height 182 cm, mass 82 kg) in the robot. The study was approved by the institutional review boards of the University of Houston and Rice University, and the subject gave written informed consent. The four experimental conditions involved 1) active robot with no arm, 2) active robot with passive arm, 3) passive robot with active arm, and 4) active robot with active arm (i.e., cooperative control). For each experimental condition, two repetitions of six reaching motions were performed. For each reaching motion, the robot (and arm if present) started in a neutral pose (shoulder angle 0°, elbow angle 90°), reached out to one of six targets, and then returned to the neutral pose (Fig. 2). The outbound and inbound motion segments were each performed in 1 second. For the third condition, the time to complete the outbound and inbound motion segments was close to 1 second but varied since the robot was passive and the arm provided all of the control. Thus 12 different motion segments (6 targets with 2 motion segments each) were performed twice for each experimental condition, resulting in a total of 24 motion segments per experimental condition treated as separate experimental trials.

3.2 Arm-robot model development

A multibody dynamic model of the Kinarm exoskeleton robot was created in OpenSim [22, 23]. Kinarm Corporation provided information on the joint center locations and mass

Fig. 2 Visual depiction of the experimental task. The experiment was carried out with specified 1 second reach times following 0, 1, 0, 2, 0, 3, etc. trajectories



properties for each link in the robot. Computer-aided design (CAD) geometric models of all robot links were reverse engineered using a combination of physical measurements made on the Kinarm robot used in this study and reference images available from the company. The kinematic structure of the Kinarm robot possesses multiple closed kinematic chains modeled using pin joints to connect the robot links and point constraints to close kinematic chains (Fig. 3). Point constraints were strategically located such that each closed kinematic chain possessed only one point constraint (with one unavoidable exception), each robot link possessed at most one point constraint, and no point constraints were used where a robot or arm control torque was applied or where a robot or arm joint angle was measured. Because the original Kinarm robot model was developed for the left arm but the experimental data and selected OpenSim upper extremity model were for a right arm (see below), custom Matlab (The Mathworks, Natick, MA) code (available on Simtk.org; see Data Availability Statement below) was developed that mirrored the original Kinarm OpenSim model to create a right-sided version.

Since the combined arm-robot model needed to function in 3D space, we used Eq. (2) to confirm that the Kinarm robot model with multiple closed kinematic chains possessed 2 DOFs as desired. As shown in Fig. 3a, the Kinarm robot model consisted of 12 bodies, 12 pin joints, and 5 point constraints. Thus, based on Eq. (2) and the constraints listed in Table 1, the number of DOFs was

$$n = 6(12) - [5(12) + 3(5)] = -3.$$

Thus, the initial model possessed $2 - (-3) = 5$ constraints too many, which led to assembly failures. We resolved this problem by replacing 5 pin joints with 5 cylindrical joints, each of which imposed 4 rather than 5 constraints by allowing out-of-plane translation. Applying

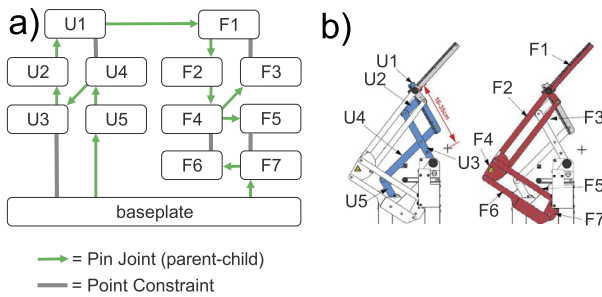


Fig. 3 a) Flowchart describing the parent–child tree structure utilized to define joints and constraints in the Kinarm exoskeleton robot model. b) Visualization of body names in the Kinarm rehabilitation robot. All joints except ground to baseplate are pin joints. U3, F3, F4, and F5 also have an additional translational DOF perpendicular to the plane of motion to meet OpenSim assembly tolerance requirements. The ground-to-baseplate joint is a 6-DOF joint to allow the robot model to be positioned and orientated anywhere relative to the ground coordinate system. All constraints used in the model are point constraints

Eq. (2) to the modified model yielded

$$n = 6(12) - [5(7) + 4(5) + 3(5)] = 2,$$

exactly as desired. This modification eliminated the initial assembly failures.

The MoBL-ARMS upper extremity musculoskeletal model developed in OpenSim [25] was selected for use in the coupled arm-robot model. This upper extremity model is three-dimensional and represents the torso and right arm. The present study used the version of the model where the torso is fixed to ground. This version possesses 7 DOFs consisting of a 3-DOF shoulder, 1-DOF elbow, 1-DOF forearm, and 2-DOF wrist. The shoulder model includes multiple coordinate couplers to couple scapuloclavicular motion to shoulder motion. For the purposes of the present study, all muscle, ligament, and coordinate limit forces were deactivated but retained for use in future studies. The generic upper extremity model was scaled to the dimensions of the experimental subject using physical measurements of segment lengths made on the subject, where the scaling process adjusted both the segment lengths and inertia properties. All coordinate couplers in the shoulder were removed from the scaled model, the previously coupled joints were locked in positions consistent with the Kinarm robot, and the shoulder was replaced with a 3-DOF ball-and-socket joint. These changes were required to eliminate incorrect model calculation of inverse dynamic shoulder muscle torques (see Discussion).

The Kinarm robot model was coupled with the scaled modified upper extremity model in a manner that mimicked the experimental setup. First, the robot and arm OpenSim models were combined automatically into a single model using custom Matlab code (available on Simtk.org) that eliminated the manual tasks normally required to combine two OpenSim models. Second, the position and orientation of the robot model with respect to ground were manually adjusted so that the robot’s shoulder and elbow axes were closely aligned with those of the upper extremity model. Third, the hand of the upper extremity model was rigidly connected to the hand rest of the Kinarm model using a weld constraint, which produced an additional closed kinematic chain and necessitated small additional adjustments of the robot position and orientation with respect to ground. In the resulting combined model, all 7 coordinates in the upper extremity remained unlocked except for the shoulder elevation coordinate, which was locked at 90° to represent the horizontal position of the arm in the robot.

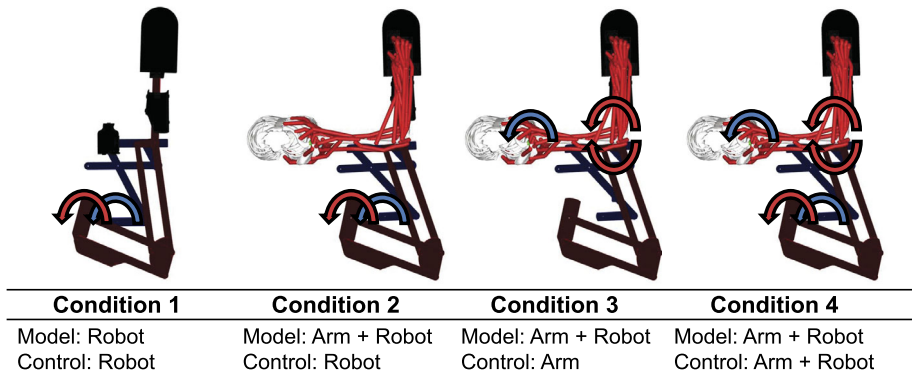


Fig. 4 Visual depiction of the four experimental conditions used in the study. Blue (lighter) curved arrows indicate where robot and/or arm shoulder torques were generated, whereas red (darker) curved arrows represent where robot and/or arm elbow torques were generated. All four conditions were used for robot and combined arm-robot model verification, whereas only the first two conditions were used for model validation

To confirm that the coupled arm-robot model possessed 2 DOFs as desired, we used Eq. (2) to calculate the number of DOFs in the final model. Prior to the addition of the weld constraint, the closed-chain Kinarm model possessed 2 DOFs whereas the open-chain arm model possessed 7 DOFs for a total of 9 DOFs. Adding a weld constraint to connect the two models and locking the shoulder elevation coordinate introduced 7 constraints. Thus, based on Eq. (2), the final combined model possessed $9 - 7 = 2$ DOFs, again exactly as desired.

3.3 Arm-robot model verification

To gain confidence in our robot and coupled arm-robot OpenSim models, we followed an established process to verify and validate both models [26, 27]. Verification addresses whether a model produces theoretically correct results for the quantities of interest, whereas validation addresses whether it produces physically realistic results that closely reproduce experimental measurements for the quantities of interest. Although validation is always important, verification was especially important for these OpenSim models due to the complexities added by a large number of closed kinematic chains.

Computational verification of the robot and coupled arm-robot models was performed for all four experimental conditions (Fig. 4). The goal of model verification was to show that robot and/or arm joint torques calculated via inverse dynamics using the experimentally measured robot motion could drive a subsequent forward dynamic simulation to reproduce the original robot motion. The verification process was performed for each of the 24 experimental trials completed for each experimental condition.

For each of the four experimental conditions, we performed one to three mixed dynamic simulations followed by a final forward dynamic simulation without prescribed motions (Table 3) to verify correct theoretical function of the robot and combined arm-robot models for all 24 experimental trials. For each experimental condition, the first mixed dynamic simulation used the robot motor angles measured experimentally to define the time histories of the corresponding prescribed coordinates in the robot portion of the model. Conditions 1 and 2 used one mixed dynamic simulation to calculate robot motor torques from measured robot motor angles. Condition 3 used two mixed dynamics simulations: the first to calculate arm joint (i.e., shoulder and elbow) angles from measured robot motor angles (required since

Table 3 Overview of computational simulations used to perform model verification for each of the four experimental conditions. For each condition, the outputs from one simulation are always used as the inputs to the next simulation. For example, for Condition 1 (robot only model), robot angles are input to the model, and the resulting robot torques are output from the model as calculated via a mixed dynamic simulation. These robot torques are then input to the same model, and the resulting robot angles are output from the model as calculated via a forward dynamic simulation. The final output robot joint angles should closely match the initial input robot joint angles if all computational processes were performed correctly, realizing that small numerical errors are introduced by numerical differentiation (for mixed dynamics) and integration (for forward dynamics)

Condition	Inputs	Outputs	Simulation
1	Robot angles	Robot torques	Mixed dynamics
	Robot torques	Robot angles	Forward dynamics
2	Robot angles	Robot torques	Mixed dynamics
	Robot torques	Robot angles	Forward dynamics
3	Robot angles	Arm angles	Mixed dynamics
	Arm angles	Arm torques	Mixed dynamics
	Arm torques	Robot angles	Forward dynamics
4	Robot angles	Arm angles	Mixed dynamics
	Arm angles	Arm torques	Mixed dynamics
	Robot angles and 50% arm torques	Robot torques	Mixed dynamics
	Robot torques and 50% arm torques	Robot angles	Forward dynamics

arm joint angles are not identical to corresponding robot joint angles), and the second to calculate arm joint torques from calculated arm joint angles. Condition 4 required three mixed dynamic simulations: the first two were the same as for Condition 3, whereas the third calculated robot motor torques from measured robot motor angles while applying 50% of the calculated arm joint torques from Condition 3 to the shoulder and elbow of the arm model. Condition 4, which represents cooperative control between the arm and robot, is an approximation of the corresponding experimental situation since the amount of shoulder and elbow joint torque provided by the subject's arm could not be measured. For all four conditions, a final forward dynamic simulation was performed using as controls the robot and/or arm torques calculated via mixed dynamics, which were applied to the model using coordinate actuators. This final simulation predicted robot motor angles to verify consistency between inverse (via mixed) and forward dynamic simulations, which is challenging for models with multiple closed kinematic chains due to integration and other numerical issues.

To quantify errors in the model verification process, we calculated three robot joint angle error measures from the 24 experimental trials simulated for each of the four experimental conditions. Robot joint (i.e., elbow and shoulder) angles were calculated from robot motor (i.e., motor1 and motor2) angles using the analytical relationship

$$\begin{aligned}\theta_{elbow} &= \theta_{motor2}, \\ \theta_{shoulder} &= \theta_{motor1} + \theta_{motor2}.\end{aligned}\quad (3)$$

Each error measure quantified differences between experimentally measured robot joint angles input (as robot motor angles) into the first mixed dynamic simulation and computationally simulated robot joint angles output by the final forward dynamic simulation. The three error measures were the root-mean-square (RMS) error, the percent RMS error calculated

by normalizing the RMS error by the joint angle range of motion, and the correlation coefficient R . Each error measure was reported as the mean \pm standard deviation of the 24 errors calculated for the individual experimental trials.

3.4 Arm-robot model validation

Experimental validation of the robot and coupled arm-robot OpenSim models was performed for only the first two experimental conditions where either the arm was not present or the arm was passive. Validation could not be performed for the other two experimental conditions since no measurements were available for the joint torques generated by the subject's arm muscles. The goal of model validation was to show that the robot joint torques calculated via inverse dynamics using the experimentally measured robot motion closely matched the robot joint torques measured experimentally. The validation process required only a single mixed dynamic simulation (Table 3) to verify realistic physical prediction of robot joint torques for each of the 24 experimental trials completed for each experimental condition.

To quantify errors in the model validation process, we calculated three robot joint torque error measures from the 24 experimental trials simulated for the first two experimental conditions. Robot joint (i.e., elbow and shoulder) torques were calculated from robot motor (i.e., motor1 and motor2) torques using the analytical relationship

$$\begin{aligned}\tau_{elbow} &= \tau_{motor2}, \\ \tau_{shoulder} &= \tau_{motor1} + \tau_{motor2}.\end{aligned}\quad (4)$$

Each error measure quantified differences between experimentally measured robot joint torques and computationally predicted robot joint torques output by a mixed dynamic simulation. The experimentally measured robot motor torques used in Eq. (2) accounted for friction effects based on a joint friction model developed by Kinarm Corporation. The three error measures were again the RMS error, the percent RMS error, and the correlation coefficient R , with each being reported as the mean \pm standard deviation of the 24 individual errors.

4 Results

For the four verification conditions, the robot and arm-robot models accurately reproduced the experimentally measured robot shoulder and elbow joint angles via forward dynamic simulations driven by inverse dynamic robot and/or arm torques (Fig. 5). The mean RMS errors in robot elbow and shoulder joint angles were less than 0.3 deg, representing at most a 1% RMS error, with all R values equal to 1.00 (Table 4).

For the two validation conditions, the robot and arm-robot models reproduced the experimentally measured robot shoulder and elbow joint torques well via mixed dynamic simulations (Figs. 6–7). The mean RMS errors in robot joint torques were less than or equal to 0.5 Nm, representing at most a 14% RMS error, with all R values greater than or equal to 0.92 (Table 5).

5 Discussion

This study provided general guidelines for constructing coupled arm-robot closed-chain models in OpenSim and then developed, verified, and validated an OpenSim model of the

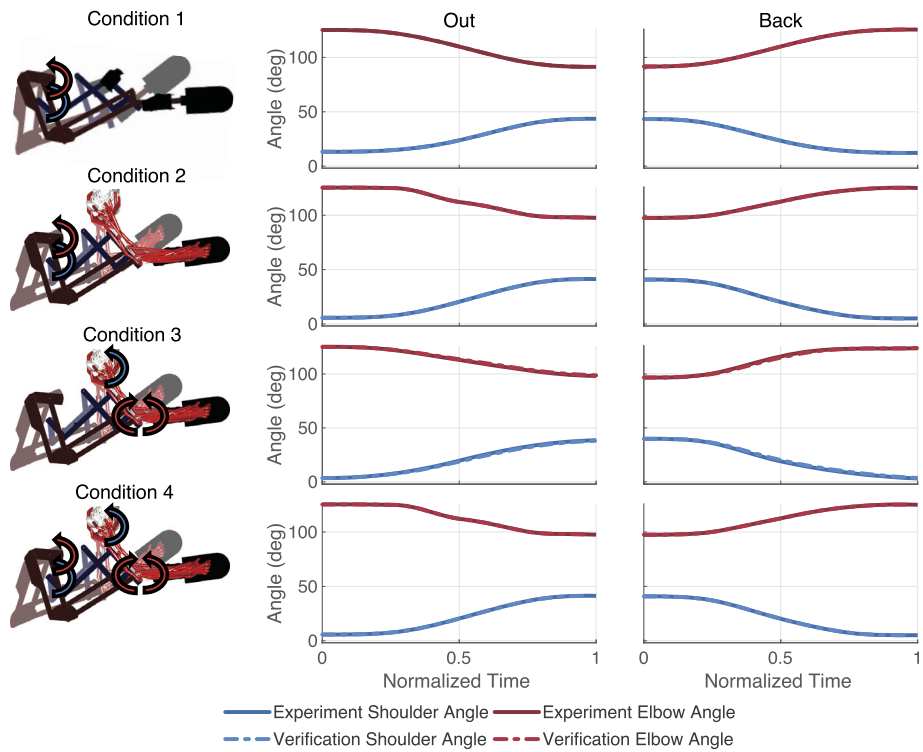


Fig. 5 One trial of each verification test for target 4 (Fig. 4) and Conditions 1 (top row: Active robot, no arm) through 4 (bottom row: Active robot, active arm) (Fig. 3). Plots were similar for the other 5 targets. The left column provides a visualization of the endpoints of the motion tested, with the shaded image representing the start/end position and the solid image representing the maximum reach position. The second and third columns show the experimental and simulated motion for each reach out (second column) and reach in (third column) motion. Blue (lighter) curved arrows indicate where robot and/or arm shoulder torques were generated, whereas red (darker) curved arrows represent where robot and/or arm elbow torques were generated. The experiment (solid) and verification (dashed) curves (blue/lighter for shoulder, red/darker for elbow) are almost indistinguishable due to extremely small verification errors

Kinarm exoskeleton rehabilitation robot coupled with a published upper extremity musculoskeletal model [25]. The combined model possessed multiple closed kinematic chains but only two DOFs, in agreement with the physical Kinarm robot. The model verification and validation processes were based on experimental robot motor angle and torque data collected from a healthy subject performing six different out-and-back reaching motions within the robot. An additional test condition involved the robot performing the same six out-and-back motions by itself without the arm present. The verification and validation simulations were performed for both the robot model alone and the combined arm-robot model. Verification of the combined model involved simulations controlled by the robot alone, the arm alone, and the arm and robot cooperatively, whereas validation of the combined model involved only simulations controlled by the robot alone, since no experimental measurements were available for the shoulder and elbow joint torques generated by the arm. The verification process revealed that the robot and combined arm-robot models generated theoretically correct simulation results within the expected errors due to numerical differentiation (for mixed dynamics) and integration (for forward dynamics). Similarly, the validation process

Table 4 Summary of differences in experimental and simulated robot joint angles for model verification. Percent RMS errors were calculated by dividing each RMS error by the maximum range of the trial. R values indicate the correlation coefficient. Values shown are the mean \pm standard deviation for all trails

Condition	Shoulder angle			Elbow angle		
	RMSE (deg)	% RMSE	R value	RMSE (deg)	% RMSE	R value
1. Active robot, no arm	0.06 \pm 0.08	0.34 \pm 0.62	1.00 \pm 5e ⁻⁵	0.06 \pm 0.06	0.19 \pm 0.17	1.00 \pm 5e ⁻⁵
2. Active robot, passive arm	0.05 \pm 0.03	0.25 \pm 0.26	1.00 \pm 4e ⁻⁵	0.09 \pm 0.10	0.36 \pm 0.42	1.00 \pm 5e ⁻⁵
3. Passive robot, active arm	0.10 \pm 0.10	0.58 \pm 0.70	1.00 \pm 1e ⁻⁴	0.27 \pm 0.35	1.12 \pm 2.00	1.00 \pm 5e ⁻⁵
4. Active robot, active arm	0.05 \pm 0.03	0.26 \pm 0.28	1.00 \pm 5e ⁻⁵	0.09 \pm 0.10	0.35 \pm 0.43	1.00 \pm 5e ⁻⁵

revealed that the models predict physically realistic robot joint torques with small RMS errors. Successful verification and validation processes confirmed the reliability and utility of the proposed general guidelines for modeling and simulating closed-chain arm-robot systems in OpenSim. With the addition of personalized neural control and arm muscle models, the combined arm-robot model may eventually provide a useful testbed for designing novel patient-specific robot control algorithms that target changing desired aspects of a stroke patient's coordination strategy.

Although the verification results provide high confidence in the combined model formulation, the validation results suggest that areas for improvement exist. Comparison of validation results for Condition 1 (active robot, no arm) and Condition 2 (active robot, passive arm) suggests that the largest area for improvement is in the arm model rather than the robot model. For Condition 1, the validation results are already excellent with mean RMS robot joint torque errors of at most 0.07 Nm. The most likely way to reduce those errors further is by improving the standard Kinarm friction model. For Condition 2, the validation results are still reasonable, but mean RMS robot joint torque errors of up to 0.49 Nm are an order of magnitude larger. Apart from simple scaling based on the measured lengths of the subject's arm segments, the generic upper extremity model was not calibrated to subject data, and no active or passive muscle forces were included in the model. Errors in uncalibrated shoulder and elbow functional axes of the upper extremity model may have had a small contribution to the torque validation errors for Condition 2. However, the largest contributor to the Condition 2 validation errors was most likely the omission of passive and active muscle force generation in the upper extremity model. Passive muscle force, which arises from the force-length properties of muscle [28], typically occurs near the ends of the ranges of joint motions [29], which are where torque matching errors tended to be the largest (Fig. 7). Furthermore, even though the subject was instructed to keep his arm passive while the robot moved it in Condition 2, it is possible that a small amount of muscle activation occurred, resulting in background active muscle forces that could have influenced the validation results. These possibilities will be explored in the future once a personalized EMG-driven musculoskeletal arm model of the subject can be successfully constructed and validated.

Only a few previous studies have coupled a robot model with some form of upper extremity musculoskeletal model to facilitate the design of assistive robotic devices and/or rehabilitation interventions [19, 20, 30]. Given the current paucity of such studies, it is worth asking whether development of combined arm-robot models provides any advantages over use of just an arm model with added mass properties to account for the presence of the

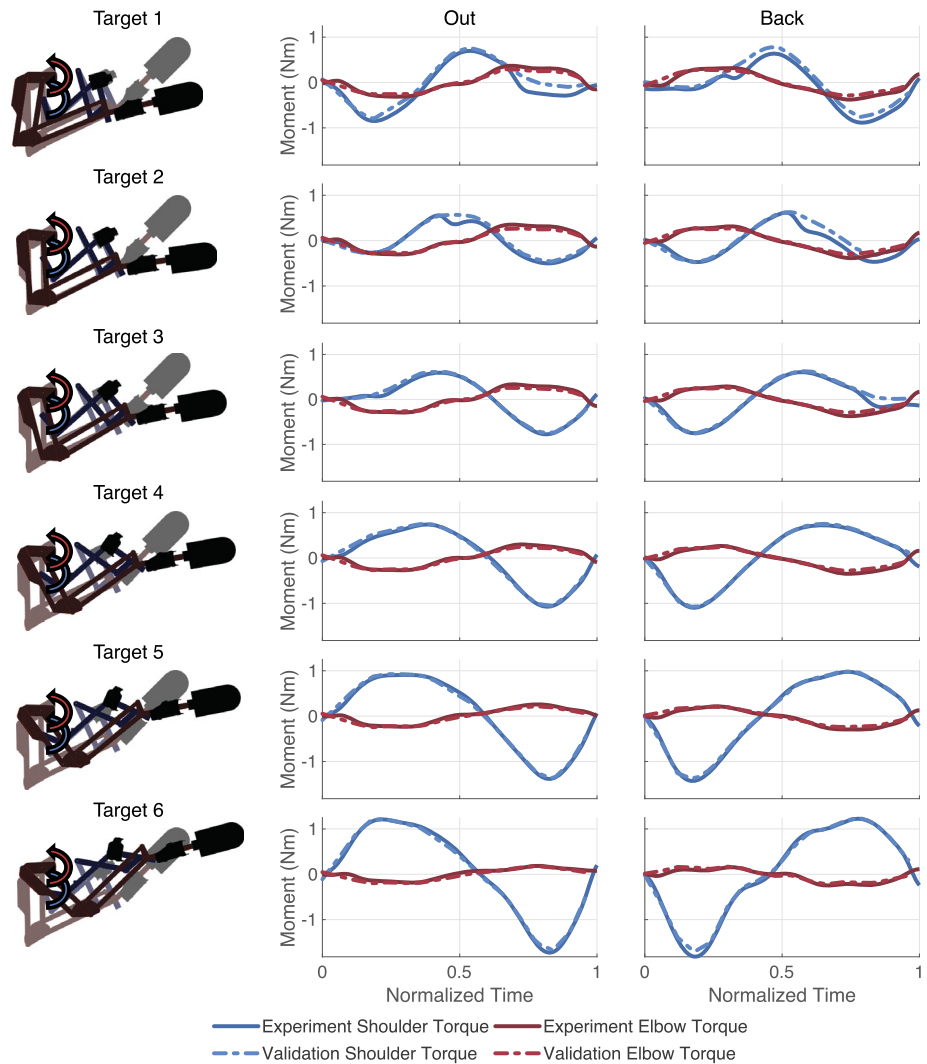


Fig. 6 One trial of each validation test for Condition 1: Active robot, no arm. The left column provides a visualization of the endpoints of the motion tested, with the shaded image representing the start/end position and the solid image representing the maximum reach position. The second and third columns show the experimental and simulated motion for each reach-out (second column) and reach-in (third column) motion. Blue (lighter) curved arrows indicate where robot shoulder torques were generated, whereas red (darker) curved arrows represent where robot elbow torques were generated. The experiment (solid) and validation (dashed) curves (blue/lighter for shoulder, red/darker for elbow) are extremely similar due to only small validation errors

robotic exoskeleton. Although creating combined arm-robot models is more complicated, we believe that this approach possesses at least two important advantages. First, creation of an explicit robot model allows for a more accurate representation of how robot dynamics and control interact with the wearer. Without an explicit robot model, it is harder to validate proper functioning of the robot portion of the model, validate different components

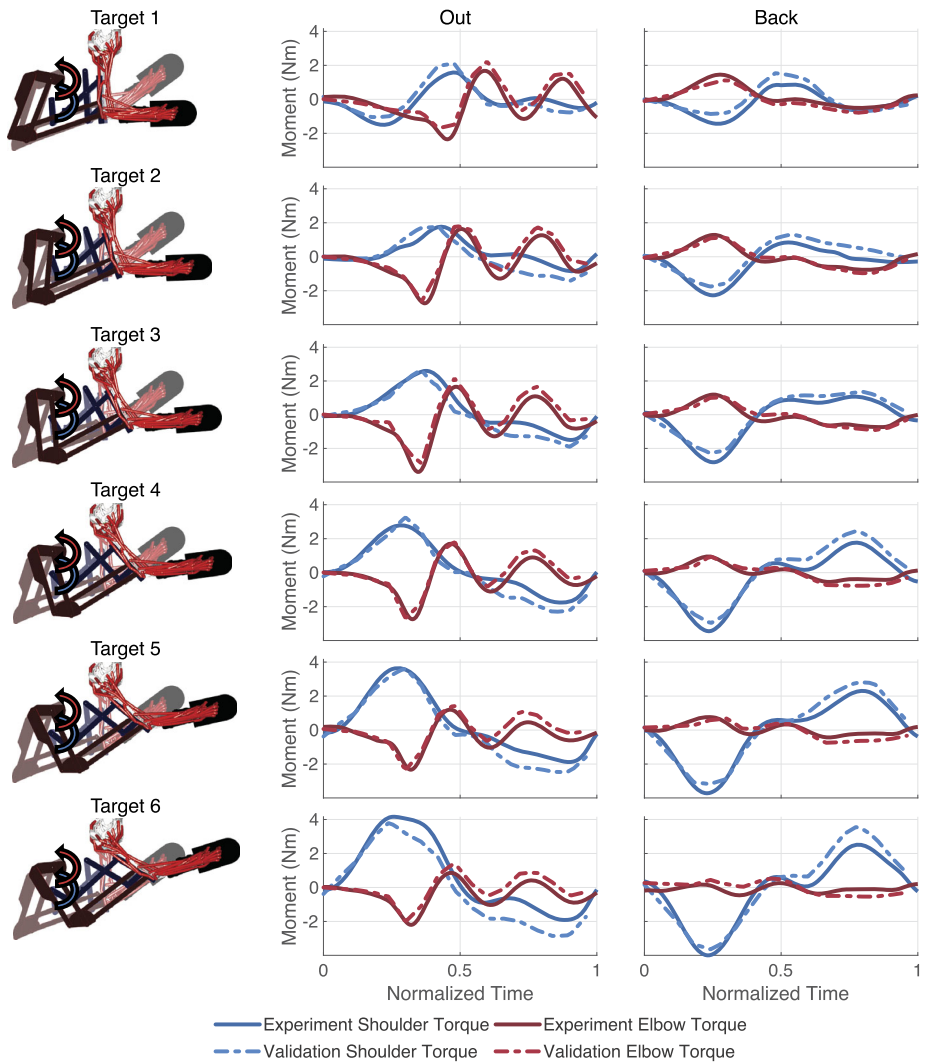


Fig. 7 One trial of each validation test for Condition 2: Active robot, passive arm. Columns and curved arrows are the same as in Fig. 6

of the robot portion of the model (e.g., joint friction model), and evaluate robot and controller design variations computationally. Lack of an explicit robot model also means that effective robot mass properties must be calculated for incorporation into the arm model segments, which could be challenging for robots like the Kinarm that possess multiple closed kinematic chains rather than a single serial structure. Second, combining an explicit robot model with an arm model accounts for misalignment between the robot joints and the human joints as well as interaction forces and moments between the exoskeleton segments and body segments [31]. Modeling misalignment and interaction forces/moments could be important for situations where wearer comfort or adverse joint motion/loading is a concern, or where

Table 5 Summary of differences between experimental and simulated robot joint moments for model validation. Percent RMS errors were calculated by dividing each RMS error by the maximum range of the trial. R values indicate the correlation coefficient. Values shown are the mean \pm standard deviation for all trails

Condition	Shoulder moment			Elbow moment		
	RMSE (Nm)	% RMSE	R value	RMSE (Nm)	% RMSE	R value
1. Active robot, no arm	0.07 \pm 0.04	4.42 \pm 3.29	0.99 \pm 0.01	0.05 \pm 0.01	7.90 \pm 1.34	0.98 \pm 0.01
2. Active robot, passive arm	0.50 \pm 0.13	12.8 \pm 4.80	0.97 \pm 0.03	0.30 \pm 0.08	14.3 \pm 9.74	0.92 \pm 0.12

misalignment might affect the assumption that the robot control torques are experienced unaltered by the human joints.

As noted earlier, the present study is only the first step toward the long-term goal of using personalized arm-robot models to design more effective robot control algorithms that account for a patient-specific neural control deficits and desired changes. To date, no such robot control algorithms have been designed using personalized computational models. Even when they can be, they will need to be evaluated in light of the complex and multifactorial nature of each patient's neural control deficits and potential recovery mechanisms.

Future implementation of a model-based approach for robotic neurorehabilitation would require an entirely new paradigm for the treatment design process [32]. The clinician would need to have access to a human movement lab possessing video motion capture and electromyographic (EMG) measurement systems, along with the upper extremity rehabilitation robot to be used for patient training. The patient would visit the movement lab for a pre-training test session during which arm motions would be performed outside and inside the robot. With current experimental technology, such movement testing would likely require about two hours of patient time. Collaborating engineers would use the patient's experimental data to create a personalized neuromusculoskeletal model of the patient's upper extremity and couple it with a previously developed and validated model of the upper extremity robot. The function of the combined arm-robot model would be validated using experimental motion and EMG data withheld for that purpose. Once formalized, this model creation and validation process would likely take about two weeks. Working with the clinician, the engineer would then run optimal control simulations using the validated combined model to predict the simplest changes to the patient's impaired neural control strategy, and how the robot controller could help induce them, which would produce the biggest gains in the patient's arm function. This predictive process would likely take an additional week or two of effort. Finally, the patient would return to the movement lab for their first robotic training session using the personalized robot control algorithm. Since the entire process would require nearly a month between initial patient testing and initial patient training, shortening this process would become the topic of future research if a personalized model-based approach to robotic neurorehabilitation proved to be highly effective.

The most significant challenges encountered in this study involved model construction and simulation. For model construction, three main challenges had to be addressed. The first was constructing the OpenSim model of the Kinarm robot from scratch, which required developing principles for how to model multiple closed kinematic chains. In OpenSim, joints are used to connect rigid bodies in open kinematic chains, whereas constraints (point, weld, or point-on-line) are used to close kinematic chains. Thus, to create a model with closed kinematic chains, one must first define joints to create open kinematic chains and then define constraints to close kinematic chains. For the planar Kinarm robot with multiple closed

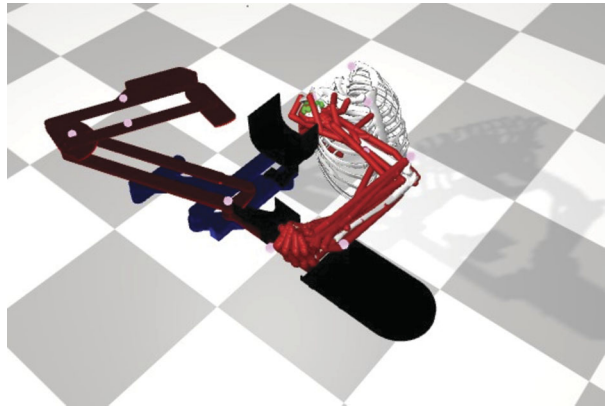
kinematic chains, we had flexibility in deciding which robot joints should be modeled using pin joints and which should be modeled using point constraints (Fig. 3). To achieve the desired model functionality, we identified two principles for deciding where to close kinematic chains using point constraints. The first principle was that two point constraints should never be used on a single rigid body, since having more than one point constraint on a rigid body increases the possibility of encountering model assembly errors. The second principle was that a point constraint should not be used to represent a pin joint where a joint torque needs to be measured, a coordinate actuator needs to be applied, or a joint motion needs to be prescribed, as OpenSim functionality related to coordinates does not work for point constraints. Thus, bodies F7 and U5 (Fig. 3) were connected to the Kinarm base using pin joints rather than point constraints, since we needed to measure the robot motor torques acting on those pin joints, apply robot motor torques to those pin joints using coordinate actuators, and prescribe the motion of those pin joints. These principles may provide a helpful starting point for other researchers seeking to develop models with closed kinematic chains in OpenSim.

The second model construction challenge was combining separate robot and arm OpenSim models into a single combined OpenSim model. To avoid performing error-prone manual model editing tasks when creating the combined model, we developed custom Matlab code (available on Simtk.org) to combine two OpenSim models automatically. Although OpenSim's current Matlab application programming interface (API) is extremely valuable, it can be challenging to use when modifying existing models. To address this issue, we developed Matlab code that allows OpenSim model files to be read into and written from a Matlab data structure. The code makes it easy to navigate through a model's structure, mirror a model to the opposite side (e.g., the Kinarm model), and combine two models into a single model.

The third model construction challenge involved the definition of the shoulder joint in the generic OpenSim upper extremity model [25]. A surprising issue surfaced when we calculated the robot motor torques using mixed dynamics for the active robot-passive arm condition. The calculated robot motor torque contributing to only the robot shoulder torque (motor 1 in Eq. (2)) was an order of magnitude larger than that measured experimentally. Furthermore, when an OpenSim inverse dynamics analysis was performed for the same motion using the arm model decoupled from the robot model, the calculated shoulder torque was again significantly larger than expected. Since the verification simulations worked as expected, these observations led us to hypothesize that a problem existed with the way the shoulder was modeled. After further investigation, we identified the nonphysical intermediate bodies and associated coordinate couplers in the shoulder joint model as the source of the problem. Replacing the shoulder model with a 3-DOF gimbal joint resolved the problem and yielded calculated motor torques that were close to those measured experimentally. Further investigation will be required to understand why nonphysical intermediate bodies and their associated coordinate couplers introduced this issue.

For model simulation, two main challenges had to be addressed, both of which were related to the presence of closed kinematic chains. The first involved assembly errors caused by small geometric inconsistencies, necessitating the introduction of a small amount of geometric "slop" into the model. By default, OpenSim uses a tight assembly tolerance when seeking to find a feasible model configuration that satisfies all constraints. When small geometric inconsistencies exist in a model with closed kinematic chains, this tight tolerance can prevent OpenSim from finding a feasible model configuration, causing the simulation to fail. After encountering this problem with the robot model, we identified two ways to resolve it. The first is to set the `assembly_accuracy` tolerance to a looser value in the OpenSim model file. A good way to select an appropriate tolerance is to determine the accuracy of the

Fig. 8 Example of an incorrect OpenSim assembled model configuration possibly due to entrapment in a local minimum. Note that the weld constraint was defined properly and the initial conditions were close to the desired final configuration



measurements used to construct the model geometry. For example, if the geometric measurements used to construct the model were accurate to within 0.1 mm, then the OpenSim `assembly_accuracy` tolerance should be set to 1×10^{-4} meters. The downside of this approach is that the assembled configuration of the model may become less accurate than desired. The second way is to introduce additional DOFs into the model that eliminate redundant constraints. We followed this approach in the present study by replacing some pin joints with cylindrical joints, which allowed extremely small out-of-plane motions in the robot model to achieve the original tight assembly accuracy for all simulations.

The second model simulation challenge involved entrapment in a local minimum when solving assembly constraints. We encountered this issue when seeking to enforce the weld constraint that connected the hand in the arm model to the handrest in the robot model. If OpenSim was given a poor initial guess for the configuration of the combined model, then its assembly algorithm would find a solution where the relative position of the two bodies in the weld constraint was correct but their relative orientation was incorrect, resulting in an unrealistic assembled configuration (Fig. 8). To help the assembly algorithm find the correct assembled configuration, we tried placing the two models in an initial configuration that was close to satisfying the weld constraint, and we also clamped the torso rotations in the arm model to be within realistic ranges. When that approach still did not produce the correct assembled configuration, we swapped the directionality of the weld constraint such that the OpenSim assembly analysis moved the robot model with respect to a fixed arm model to perform the assembly process. Although this final approach produced the correct assembled configuration of the model, it remains unclear why having OpenSim move the arm model with respect to the robot model when enforcing the weld constraint would not work.

This study possesses two types of limitations that should be considered when evaluating the current study and planning future studies with the arm-robot model. The first type is model limitations. Neural control and muscle–tendon models were not utilized in the existing arm model, though generic musculoskeletal geometry is already present in the arm model. To our knowledge, no study to date has developed and validated a personalized neuromusculoskeletal arm model that when driven by the subject's experimental EMG data can reproduce the subject's experimentally measured motion via a forward dynamic simulation, as the development of such a model remains a significant research challenge. Furthermore, the development of such a personalized model will require significant modifications to the MoBL-ARMS upper extremity model, the development of regression relationships for estimating peak isometric forces for upper extremity muscles, and the identification of a reliable method for estimating missing EMG signals. In addition, interactions between the

upper arm/forearm and their corresponding armrests on the robot were not modeled. Modeling arm interaction with the armrests will require eliminating the weld constraint between the hand and handrest and adding either point-on-line constraints or simple contact models between the upper arm/forearm and the corresponding robot armrests. The second type is experimental limitations. EMG, arm motion (via video motion capture), and arm torque (via inverse dynamics) data were not available to validate the two experimental conditions with active arm control. Collection of these additional data in a future study would allow the incorporation of subject-specific neural control and muscle–tendon models into the arm model and subsequent validation of combined model function in the two conditions with active arm control.

6 Conclusion

This study presented general guidelines for creating and simulating coupled arm-robot models in OpenSim, as coupling a skeletal arm model to a rehabilitation robot model introduces a closed kinematic chain that significantly complicates the OpenSim modeling and simulation process. The utility of these general guidelines was demonstrated using a specific example involving modeling and simulation of a three-dimensional arm moving within a two degree-of-freedom upper extremity rehabilitation robot, where the combined arm-robot OpenSim model was verified computationally and validated experimentally. With the future addition of patient specific muscle–tendon and neural control models, the coupled arm-robot OpenSim model may prove useful for designing patient-specific cooperative arm-robot control algorithms that improve the effectiveness of robotic training for stroke neurorehabilitation.

Author Contribution MG, YH, BF, and JR performed the experiment, MG developed the computational models and performed the computational simulations, MG and BF analyzed and interpreted the computational results, MG and BF wrote the manuscript and generated the figures, and all authors reviewed the manuscript before submission.

Funding This study was funded by the United States Air Force Academy (MG), Rice University (BF), and the University of Houston (YH and JR).

Data Availability The Matlab code and experimental data used to perform this study can be found on Simtk.org at <https://simtk.org/projects/armrobotmodel>.

Declarations

Ethics statement The human subject who participated in this study gave informed written consent. The study was approved by the institutional review boards of the University of Houston, where the data were collected, and Rice University, where the data were modeled and simulated.

Competing Interests The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Feigin, V.L., Nquyen, G., Cercy, K.: Global, regional, and country-specific lifetime risks of stroke, 1990 and 2016. *N. Engl. J. Med.* **379**, 2429–2437 (2018). <https://doi.org/10.1056/NEJMoa1804492>
2. Cramer, S.C., Nelles, G., Benson, R.R., Kaplan, J.D., Parker, R.A., Kwong, K.K., Kennedy, D.N., Finklestein, S.P., Rosen, B.R.: A functional MRI study of subjects recovered from hemiparetic stroke. *Stroke* **28**, 2518–2527 (1997). <https://doi.org/10.1161/01.str.28.12.2518>
3. Kantak, S.S., Zahedi, N., McGrath, R.: Complex skill training transfers to improved performance and control of simpler tasks after stroke. *Phys. Ther.* **97**, 718–728 (2017). <https://doi.org/10.1093/ptj/pzx042>
4. Kelly-Hayes, M., Beiser, A., Kase, C.S., Scaramucci, A., D'Agostino, R.B., Wolf, P.A.: The influence of gender and age on disability following ischemic stroke: the Framingham study. *J. Stroke Cerebrovasc. Dis.* **12**, 119–126 (2003). [https://doi.org/10.1016/S1052-3057\(03\)00042-9](https://doi.org/10.1016/S1052-3057(03)00042-9)
5. Go, A.S., Mozaffarian, D., Roger, V.L., Benjamin, E.J., Berry, J.D., Borden, W.B., Bravata, D.M., Dai, S., Ford, E.S., Fox, C.S., Franco, S., Fullerton, H.J., Gillespie, C., Hailpern, S.M., Heit, J.A., Howard, V.J., Huffman, M.D., Kissela, B.M., Kittner, S.J., Lackland, D.T., Lichtman, J.H., Lisabeth, L.D., Magid, D., Marcus, G.M., Marelli, A., Matchar, D.B., McGuire, D.K., Mohler, E.R., Moy, C.S., Mussolino, M.E., Nichol, G., Paynter, N.P., Schreiner, P.J., Sorlie, P.D., Stein, J., Turan, T.N., Virani, S.S., Wong, N.D., Woo, D., Turner, M.B.: Heart disease and stroke statistics—2013 update. *Circulation* **127**, e6–e245 (2013). <https://doi.org/10.1161/CIR.0b013e31828124ad>
6. Theis, K.A., Roblin, D.W., Helmick, C.G., Luo, R.: Prevalence and causes of work disability among working-age U.S. adults, 2011–2013, NHIS. *Disabil. Health J.* **11**, 108–115 (2018). <https://doi.org/10.1016/j.dhjo.2017.04.010>
7. Babaiasl, M., Mahdioun, S.H., Jaryani, P., Yazdani, M.: A review of technological and clinical aspects of robot-aided rehabilitation of upper-extremity after stroke. *Disabil. Rehabil., Assist. Technol.* **11**, 263–280 (2016). <https://doi.org/10.3109/17483107.2014.1002539>
8. Proietti, T., Crocher, V., Roby-Brami, A., Jarrassé, N.: Upper-limb robotic exoskeletons for neurorehabilitation: a review on control strategies. *IEEE Rev. Biomed. Eng.* **9**, 4–14 (2016). <https://doi.org/10.1109/RBME.2016.2552201>
9. Blank, A.A., French, J.A., Pehlivan, A.U., O'Malley, M.K.: Current trends in robot-assisted upper-limb stroke rehabilitation: promoting patient engagement in therapy. *Curr. Phys. Med. Rehabil. Rep.* **2**, 184–195 (2014). <https://doi.org/10.1007/s40141-014-0056-z>
10. Bertani, R., Melegari, C., De Cola, M.C., Bramanti, A., Bramanti, P., Calabrò, R.S.: Effects of robot-assisted upper limb rehabilitation in stroke patients: a systematic review with meta-analysis. *Neurol. Sci.* **38**, 1561–1569 (2017). <https://doi.org/10.1007/s10072-017-2995-5>
11. Chien, W.-T., Chong, Y.-Y., Tse, M.-K., Chien, C.-W., Cheng, H.-Y.: Robot-assisted therapy for upper-limb rehabilitation in subacute stroke patients: a systematic review and meta-analysis. *Brain Behav.* **10**, e01742 (2020). <https://doi.org/10.1002/brb3.1742>
12. Reinkensmeyer, D.J., Wolbrecht, E.T., Chan, V., Chou, C., Cramer, S.C., Bobrow, J.E.: Comparison of three-dimensional, assist-as-needed robotic arm/hand movement training provided with Pneu-WREX to conventional tabletop therapy after chronic stroke. *Am. J. Phys. Med. Rehabil.* **91**, S232–241 (2012). <https://doi.org/10.1097/PHM.0b013e31826bce79>
13. Frullo, J.M., Elinger, J., Pehlivan, A.U., Fitle, K., Nedley, K., Francisco, G.E., Sergi, F., O'Malley, M.K.: Effects of assist-as-needed upper extremity robotic therapy after incomplete spinal cord injury: a parallel-group controlled trial. *Front. Neurobot.* **11** (2017). <https://doi.org/10.3389/fnbot.2017.00026>
14. Molteni, F., Gasperini, G., Cannaviello, G., Guanziroli, E.: Exoskeleton and end-effector robots for upper and lower limbs rehabilitation: narrative review. *PM&R* **10**, S174–S188 (2018). <https://doi.org/10.1016/j.pmrj.2018.06.005>
15. Meyer, A.J., Eskinazi, I., Jackson, J.N., Rao, A.V., Patten, C., Fregly, B.J.: Muscle synergies facilitate computational prediction of subject-specific walking motions. *Front. Bioeng. Biotechnol.* **4** (2016). <https://doi.org/10.3389/fbioe.2016.00077>
16. Sauder, N.R., Meyer, A.J., Allen, J.L., Ting, L.H., Kesar, T.M., Fregly, B.J.: Computational design of FastFES treatment to improve propulsive force symmetry during post-stroke gait: a feasibility study. *Front. Neurobot.* **13**, 80 (2019). <https://doi.org/10.3389/fnbot.2019.00080>
17. Tröster, M., Wagner, D., Müller-Graf, F., Maufroy, C., Schneider, U., Bauernhansl, T.: Biomechanical model-based development of an active occupational upper-limb exoskeleton to support healthcare workers in the surgery waiting room. *Int. J. Environ. Res. Public Health* **17**, 5140 (2020). <https://doi.org/10.3390/ijerph17145140>
18. Nelson, A.J., Hall, P.T., Saul, K.R., Crouch, D.L.: Effect of mechanically passive, wearable shoulder exoskeletons on muscle output during dynamic upper extremity movements: a computational simulation study. *J. Appl. Biomech.* **36**, 59–67 (2020). <https://doi.org/10.1123/jab.2018-0369>

19. Blanco, A., Catalán, J.M., Díez, J.A., García, J.V., Lobato, E., García-Aracil, N.: Electromyography assessment of the assistance provided by an upper-limb exoskeleton in maintenance tasks. *Sensors* **19**, 3391 (2019). <https://doi.org/10.3390/s19153391>
20. Ghannadi, B., Sharif Razavian, R., McPhee, J.: Configuration-dependent optimal impedance control of an upper extremity stroke rehabilitation manipulandum. *Front. Robot. AI* **5**, 124 (2018). <https://doi.org/10.3389/frobt.2018.00124>
21. Buongiorno, D., Barsotti, M., Barone, F., Bevilacqua, V., Frisoli, A.: A linear approach to optimize an EMG-driven neuromusculoskeletal model for movement intention detection in myo-control: a case study on shoulder and elbow joints. *Front. Neurobot.* **12** (2018). <https://doi.org/10.3389/fnbot.2018.00074>
22. Delp, S.L., Anderson, F.C., Arnold, A.S., Loan, P., Habib, A., John, C.T., Guendelman, E., Thelen, D.G.: OpenSim: open-source software to create and analyze dynamic simulations of movement. *IEEE Trans. Biomed. Eng.* **54**, 1940–1950 (2007). <https://doi.org/10.1109/TBME.2007.901024>
23. Seth, A., Hicks, J.L., Uchida, T.K., Habib, A., Dembia, C.L., Dunne, J.J., Ong, C.F., DeMers, M.S., Rajagopal, A., Millard, M., Hamner, S.R., Arnold, E.M., Yong, J.R., Lakshminanth, S.K., Sherman, M.A., Ku, J.P., Delp, S.L.: OpenSim: simulating musculoskeletal dynamics and neuromuscular control to study human and animal movement. *PLoS Comput. Biol.* **14**, e1006223 (2018). <https://doi.org/10.1371/journal.pcbi.1006223>
24. Fregly, B.J., Fregly, C.D., Kim, B.T.: Computational prediction of muscle moments during ARED squat exercise on the international space station. *J. Biomech. Eng.* **137**, 121005 (2015). <https://doi.org/10.1115/1.4031795>
25. Saul, K.R., Hu, X., Goehler, C.M., Vidt, M.E., Daly, M., Velisar, A., Murray, W.M.: Benchmarking of dynamic simulation predictions in two software platforms using an upper limb musculoskeletal model. *Comput. Methods Biomech. Biomed. Eng.* **18**, 1445–1458 (2015). <https://doi.org/10.1080/10255842.2014.916698>
26. Hicks, J.L., Uchida, T.K., Seth, A., Rajagopal, A., Delp, S.L.: Is my model good enough? Best practices for verification and validation of musculoskeletal models and simulations of movement. *J. Biomech. Eng.* **137**, 020905 (2015). <https://doi.org/10.1115/1.4029304>
27. Hack, E., Burguete, R., Dvurecenska, K., Labeas, G., Patterson, E., Siebert, T., Szigeti, E.: Steps Towards Industrial Validation Experiments. *Proceedings* **2**, 391 (2018). <https://doi.org/10.3390/ICEM18-05216>
28. Zajac, F.E.: Muscle and tendon: properties, models, scaling, and application to biomechanics and motor control. *Crit. Rev. Biomed. Eng.* **17**, 359–411 (1989)
29. Silder, A., Whittington, B., Heiderscheit, B., Thelen, D.G.: Identification of passive elastic joint moment-angle relationships in the lower extremity. *J. Biomech.* **40**, 2628–2635 (2007). <https://doi.org/10.1016/j.jbiomech.2006.12.017>
30. Agarwal, P., Neptune, R.R., Deshpande, A.D.: A simulation framework for virtual prototyping of robotic exoskeletons. *J. Biomech. Eng.* **138**, 061004 (2016). <https://doi.org/10.1115/1.4033177>
31. Serranolf, G., Falisse, A., Dembia, C., Vantilt, J., Tanghe, K., Lefeber, D., Jonkers, I., De Schutter, J., De Groot, F.: Subject-exoskeleton contact model calibration leads to accurate interaction force predictions. *IEEE Trans. Neural Syst. Rehabil. Eng.* **27**, 1597–1605 (2019). <https://doi.org/10.1109/TNSRE.2019.2924536>
32. Hong, Y.N.G., Ballekere, A.N., Fregly, B.J., Roh, J.: Are muscle synergies useful for stroke rehabilitation? *Curr. Opin. Biomed. Eng.* **19**, 100315 (2021). <https://doi.org/10.1016/j.cobme.2021.100315>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.