Robotic Repositioning of Human Limbs via Model Predictive Control

Kevin Chow and Charles C. Kemp

***Abstract*— Robots that effectively manipulate the human body could potentially be useful in a wide variety of applications, including assistive applications for people with disabilities. Toward this end, we present a method to enable robots to compliantly manipulate human limbs. Our approach uses model predictive control (MPC). Given an action by the robot, the model predicts how the human body will move and what forces the robot will apply to the human body. The robot uses this model to optimize its actions to achieve desired motions of the human body while controlling applied forces. This optimization is subject to various constraints, including constraints to avoid hyperextension of the human's joints and to avoid slipping of the robot's end effectors. In this paper, our controller uses a quasistatic model of the human limb in contact with the robot's end effectors, which have linear Cartesian stiffness with respect to Cartesian equilibrium positions. We evaluated our approach in simulation with the specific task of lifting the leg of a human body in a supine position (i.e., lying down). In our tests, we varied the goal configuration for the human leg, the stiffness of the robot's two end effectors, and the model error (i.e., the difference between the controller's model of the human body and the actual human body). Our evaluation demonstrates the feasibility of our approach, since our controller performed well in terms of the forces the robot applied to the human leg and the human leg's motions.**

I. Introduction

The human body is a complex object that merits special consideration by robots. Robots that effectively manipulate the human body could potentially be useful in a wide variety of applications. For our research, healthcare serves as an especially motivating application area. Many healthcare-related tasks could potentially benefit from robotic manipulation of the human body, including preparation for surgery, diagnostic procedures, rehabilitation, and assistance.

In this paper, we focus on the task of repositioning a person's limb (i.e., an arm or leg). This often serves as a subtask for other body manipulation tasks and sometimes serves as a complete task on its own. We propose a method by which a compliant robot could potentially use its end effectors to reposition a person's limb. We evaluated this method using a physics-based simulation of a robot with two end effectors repositioning the leg of a human in a supine position. This is a challenging task due to the substantial weight of the leg, the use of two small areas of nonprehensile contact, the importance of controlling the contact forces, and the articulated and constrained nature of the limb.

In addition to being an interesting problem for robotic manipulation, robots that assist with limb repositioning could potentially be beneficial to patients in hospitals or people with disabilities [1]. For example, the standard nursing practice is to reposition people in bed who cannot move themselves as often as once every 2 hours to prevent the formation of pressure ulcers [2]. Also, many people with motor impairments require assistance moving their legs, as evidenced by the wide variety of assistive devices that exist to help people with disabilities move their own legs [3]. A robot could potentially provide assistance with these tasks and others.

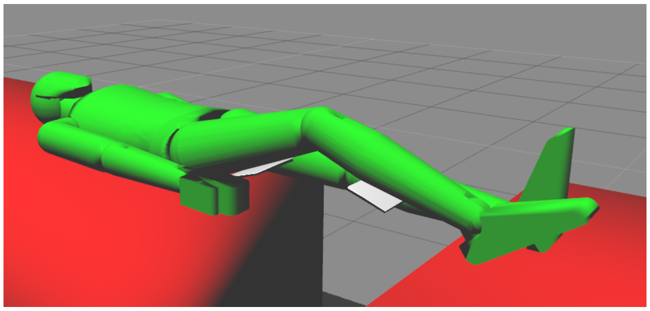

Fig. 1. An image from the physics simulation we used to evaluate our approach. The green figure is the simulated human body. The two white rectangles are the robot's two Cartesian impedance controlled end effectors. The red blocks are the surfaces upon which the body rests on during each trial.

Our approach uses model predictive control (MPC). MPC has the advantage of enabling us to explicitly define costs and constraints relevant to the task of manipulating the person's body. Given a proposed action by the robot, the model predicts the forces that will be applied by the robot and how the person's body will move. This allows the robot to optimize its actions to reduce the predicted forces, while achieving desirable predicted motions. It also allows the robot to select actions that the model predicts will satisfy constraints. These constraints can be used to avoid actions that might apply too much force to the person's body or hyperextend a joint. They can also help the end effectors stay in contact with the body and not slip.

In this paper, we use a quasistatic model of the robot manipulating a human limb. We consider this to be a good baseline model for this manipulation problem, since slow motions of the human limb would be desirable for many applications. We expected a quasistatic model to have suitable predictive power due to the limb's low joint velocities and joint accelerations. Our results support this notion. Investigating the use of dynamic models would be a natural step for future research, and could build on the framework we present in this paper.

To evaluate our approach, we created a specific controller in simulation based on our general formulation (see Fig. 1). The controller enabled a simulated robot to lift a leg with a knee joint to achieve goal poses for the thigh and shank.

We evaluated the performance of this controller in terms of its ability to smoothly move the simulated leg to desired configurations in reasonable lengths of time with appropriate applied forces. In our tests, we varied the goal configurations, the stiffness of the robot's end effectors, and the difference between the controller's model of the human body and the actual human body (i.e., model error). The results from our evaluation demonstrate the feasibility of our approach.

II. Related Work

Most research on robotic manipulation of the human body has focused on exoskeletons or specialized devices. Researchers have investigated rehabilitation devices such as the MIT-Manus and the Mirror Image Movement Enabler (MIME), which constrain arm movement and provide graded assistance to users in reaching tasks [4]. Researchers have also investigated lower-extremity rehabilitation devices to move the legs in conjunction with treadmills for gait-training [5] [6]. Likewise, researchers and practitioners have developed a variety of robotic exoskeletons for the lower extremities, such as the Berkeley Lower Extremity Exoskeleton (BLEEX) [7], EXPOS [8], and HAL [9]. Other related assistive devices include intelligent wheelchairs and wheelchairs with patient transferring capabilities, such as [10]. The "Intelligent Sweet Home" robotic residence developed at KAIST includes an intelligent bed robot system consisting of a manipulator and pressure sensors that can, among other features, provide support to a person in bed trying to reposition his or her body [11]. Most of these research efforts involve specialized robotic devices, such as exoskeletons strapped to the person's body, or actuated beds or chairs with large surface areas for contact with the person's body. In contrast, we focus on the possibility of a general-purpose robot using its end effectors to manipulate the human body, potentially with small contact areas and nonprehensile contact.

Research in nonprehensile manipulation spans topics ranging from orienting parts in factories to rolling, lifting, pushing, and throwing objects [12]. Researchers have previously investigated two-armed nonprehensile robotic manipulation using a geometric approach involving friction cones to help plan actions that maneuver parts [13]. Researchers have also worked with cooperative mobile manipulation [14] and cooperative lifting of objects between multiple robots [15][16]. Although the cooperative and nonprehensile control strategies in these studies have similarities to our approach, our focus on manipulating a human limb is distinct. In particular, we consider the problem of compliantly manipulating an articulated human limb with biomechanically-relevant constraints and costs.

The most similar research of which we are aware comes from a long-term effort to develop the RI-MAN [17] and the more recent RIBA (Robot for Interactive Body Assistance) robots that include capabilities for lifting a person. The RIBA robot has been used to investigate the possibility of mobile two-armed robots transferring people between beds and wheelchairs, lifting people, and estimating a person's comfort during lifting [18] [19] [20]. Additional studies have looked into developing a mechanical model to be used by controllers that lift and carry a person [21]. Similarly, we use a model-based approach to inform controllers that manipulate the human body. However, we focus on limb manipulation, provide a general formulation for our controller, and represent biomechanical considerations as costs and constraints. The details of our approach also differ, such as our use of model predictive control (MPC).

III. MPC for Repositioning a Human Limb

In this section, we present our general approach to robotic repositioning of a human limb. We start with a general model of the robot in contact with the human limb. Next, we find an approximate quasistatic model. We then take derivatives to find a linear model. Given a proposed change to the equilibrium positions of the robot's end effectors, this linear model predicts how the limb's joint angles and the applied forces will change. We formulate a quadratic programming problem based on this linear model that the robot solves at each time step in order to decide how to change the equilibrium positions of its end effectors. Throughout this paper, we use lowercase boldface symbols to represent vectors, uppercase boldface symbols to represent matrices, and lowercase symbols to represent scalars.

A. Model of the Human Limb and the Robot

We start by modeling the human limb as a serial chain of rigid bodies connected via m pin joints using the common joint-space Lagrangian formulation for a serial torque-controlled robot manipulator [22].

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\tau}_{human} + \boldsymbol{\tau}_{robot} \tag{1}$$

where $\boldsymbol{q}$ is the m-dimensional vector of the human limb's joint angles, $\dot{\boldsymbol{q}}$ and $\ddot{\boldsymbol{q}}$ are the limb's m-dimensional joint velocity and joint acceleration vectors, $\boldsymbol{M}(\boldsymbol{q})$ is the limb's $m \times m$ mass matrix, $\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})$ is the $m \times m$ Coriolis and centrifugal matrix for the limb, $\boldsymbol{g}(\boldsymbol{q})$ is the limb's m-dimensional configuration-dependent gravity vector, $\boldsymbol{\tau}_{human}$ is the m-dimensional joint torque vector due to moments generated by the limb itself, and $\boldsymbol{\tau}_{robot}$ is the m-dimensional joint torque vector generated by contact with the robot. In addition, we define joint angle constraints for the human limb to model acceptable ranges of motion, which we denote using element-wise inequalities as $\boldsymbol{q}_{min} \leq \boldsymbol{q} \leq \boldsymbol{q}_{max}$, where $\boldsymbol{q}_{min}$ and $\boldsymbol{q}_{max}$ are m-dimensional vectors representing the element-wise minimum angles and maximum angles for the human limb's joints.

We model the base of the human limb, the trunk, as being rigidly fixed to the world. We only model contact by the robot and assume all other contact with the limb is negligible. We model contact with the robot as consisting of a set of n point contacts, each of which only applies a d-dimensional force vector, $\boldsymbol{f}_{robot_i}$, to the human limb with no accompanying moments, such that

$$\boldsymbol{\tau}_{robot} = \sum_{i=1}^{n} \boldsymbol{J}_i^T \boldsymbol{f}_{robot_i} \tag{2}$$

where n is the total number of contacts, and $\boldsymbol{J}_i$ is the $d \times m$ contact Jacobian matrix for the i^{th} contact.

We model the robot as having a single end effector at each of the n contact locations. The robot controls the end effector, i, by commanding a d-dimensional Cartesian equilibrium position vector, $\boldsymbol{x}_{eq_i}$, that results in the force, $\boldsymbol{f}_{robot_i}$, applied at contact i with

$$\boldsymbol{f}_{robot_i} = \boldsymbol{K}_i(\boldsymbol{x}_{eq_i} - \boldsymbol{x}_i) \tag{3}$$

where $\boldsymbol{x}_i$ is the current contact location, and $\boldsymbol{K}_i$ is the end effector's $d \times d$ stiffness matrix. This robot model corresponds with a robot using a form of low-stiffness Cartesian impedance control with linear Cartesian stiffness at each of its end effectors [23][24] .

Furthermore, we model the contact between the robot and the human limb as not slipping or breaking. In other words, for the short periods of time over which the controller makes predictions, it models each contact position, $\boldsymbol{x}_i$, as being attached to a location on the human limb. In addition, our model assumes that no new significant contacts occur over this period of time. As we discuss later, our controller uses constraints on the predicted contact forces for each contact to avoid slipping or breaking contact.

Combining the components of this model results in

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\tau}_{human} + \sum_{i=1}^{n} \boldsymbol{J}_i^T \boldsymbol{K}_i(\boldsymbol{x}_{eq_i} - \boldsymbol{x}_i). \tag{4}$$

B. Quasistatic Model

For this paper, we intend for the control system to be used by robots that move the human body smoothly and slowly with negligible inertial effects, such that $\dot{\boldsymbol{q}} \approx 0$ and $\ddot{\boldsymbol{q}} \approx 0$. As such, we created a quasistatic model that neglects torques from the mass matrix and Coriolis and centrifugal matrix. Furthermore, we modeled the torques generated by the human body as being a function of the joint angles $\boldsymbol{\tau}_{human} = -\boldsymbol{h}(\boldsymbol{q})$, which is a form that has previously been used to model joint torques in the hip [25]. These modeling decisions result in

$$\boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{h}(\boldsymbol{q}) = \sum_{i=1}^{n} \boldsymbol{J}_i^T \boldsymbol{K}_i(\boldsymbol{x}_{eq_i} - \boldsymbol{x}_i). \tag{5}$$

C. Linear Model

We use our quasistatic model to predict changes to the human limb's joint angles, $\boldsymbol{q}$, and the forces applied to the human limb, $\boldsymbol{f}_{robot_i}$, given changes to the equilibrium positions of the contact points, $\boldsymbol{x}_{eq_i}$. We estimate these changes using the $m \times nd$ matrix $\frac{\partial \boldsymbol{q}}{\partial \boldsymbol{x}_{eq}}$ and the $nd \times nd$ matrix $\frac{\partial \boldsymbol{f}_{robot}}{\partial \boldsymbol{x}_{eq}}$, where $\boldsymbol{x}_{eq}$ is an nd-dimensional vector that is the concatenation of the n distinct d-dimensional equilibrium contact position vectors, $\boldsymbol{x}_{eq_i}$, and $\boldsymbol{f}_{robot}$ is an nd-dimensional vector that is the concatenation of the n distinct d-dimensional contact force vectors, $\boldsymbol{f}_{robot_i}$.

We first rewrite our quasistatic model in matrix form as

$$\boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{h}(\boldsymbol{q}) = \boldsymbol{J^T K}(\boldsymbol{x}_{eq} - \boldsymbol{x}) \tag{6}$$

where $\boldsymbol{K}$ is an $nd \times nd$ block diagonal matrix with $\boldsymbol{K}_{i,i} = \boldsymbol{K}_i$, $\boldsymbol{J^T} = [\boldsymbol{J}_1^T ... \boldsymbol{J}_n^T]$ is an $m \times nd$ block matrix representing the n contact Jacobians, and $\boldsymbol{x}$ is an nd-dimensional vector that is the concatenation of the n distinct d-dimensional contact locations, $\boldsymbol{x}_i$.

If the human limb were not in static equilibrium, it might begin to accelerate due to gravity and a lack of sufficient support from the robot. Our model assumes that the equations of static equilibrium, represented by (Eq. 6), are satisfied at all times. $rank(\boldsymbol{J^T K}) = m$ for the $m \times nd$ matrix $\boldsymbol{J^T K}$ would ensure that this is possible. Although it is not generally necessary, this can be achieved by having a contact point on each link of the human limb, since this allows an independent force in an arbitrary direction to be applied to each link. However, once used by the controller, this may or may not be achievable due to constraints and model error.

1) Solving for $\frac{\partial \boldsymbol{q}}{\partial \boldsymbol{x}_{eq}}$: Since our model assumes that no contact slips or breaks, we can write the contact locations, $\boldsymbol{x}$, as a function, $\boldsymbol{x}(\boldsymbol{q})$, of the human limb's joint angles, $\boldsymbol{q}$. Notably, $\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{q}} = \boldsymbol{J}$. We also approximate the contact Jacobians, $\boldsymbol{J}$, as being constant between controller steps due to small changes in the joint angles, $\boldsymbol{q}$, which is an approximation that we have found to be useful and effective in our previous research, such as [26]. We then rewrite our quasistatic model as

$$\boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{h}(\boldsymbol{q}) + \boldsymbol{J^T K x}(\boldsymbol{q}) = \boldsymbol{J^T K x}_{eq} \tag{14}$$

and use implicit differentiation to find

$$\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{q}} \frac{\partial \boldsymbol{q}}{\partial \boldsymbol{x}_{eq}} + \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{q}} \frac{\partial \boldsymbol{q}}{\partial \boldsymbol{x}_{eq}} + \boldsymbol{J^T K} \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{q}} \frac{\partial \boldsymbol{q}}{\partial \boldsymbol{x}_{eq}} = \boldsymbol{J^T K}, \tag{15}$$

which we can write as

$$(\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{q}} + \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{q}} + \boldsymbol{J^T K J}) \frac{\partial \boldsymbol{q}}{\partial \boldsymbol{x}_{eq}} = \boldsymbol{J^T K}. \tag{16}$$

If the square $m \times m$ matrix $\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{q}} + \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{q}} + \boldsymbol{J^T K J}$ is invertible, we can solve for $\frac{\partial \boldsymbol{q}}{\partial \boldsymbol{x}_{eq}}$ with

$$\frac{\partial \boldsymbol{q}}{\partial \boldsymbol{x}_{eq}} = (\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{q}} + \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{q}} + \boldsymbol{J^T K J})^{-1} \boldsymbol{J^T K}. \tag{17}$$

For this paper, we only consider situations for which $\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{q}} + \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{q}} + \boldsymbol{J^T K J}$ is invertible. We computed the $m \times nd$ matrix $\frac{\partial \boldsymbol{q}}{\partial \boldsymbol{x}_{eq}}$ symbolically for the model that we used in our evaluation. We also note that for our model to predict that the robot can independently influence the m joint angles, it must be the case that $rank(\frac{\partial \boldsymbol{q}}{\partial \boldsymbol{x}_{eq}}) = m$, which will only be possible with a sufficient number of contacts, $n \geq \frac{m}{d}$.

$$\underset{\Delta \boldsymbol{x}_{eq}}{\textbf{minimize}}$$

$$\alpha \left\| \Delta \boldsymbol{q}_{des} - \frac{\partial \boldsymbol{q}}{\partial \boldsymbol{x}_{eq}} \Delta \boldsymbol{x}_{eq} \right\|^2 \tag{7}$$

$$+ \sum_{i=1}^{n} \beta_i \left\| \boldsymbol{f}_{robot_i} + \frac{\partial \boldsymbol{f}_{robot_i}}{\partial \boldsymbol{x}_{eq}} \Delta \boldsymbol{x}_{eq} \right\|^2 \tag{8}$$

$$+ \sum_{i=1}^{n} \sum_{j=i+1}^{n} \gamma_{i,j} \left\| \hat{\boldsymbol{n}}_i^T \left(\boldsymbol{f}_{robot_i} + \frac{\partial \boldsymbol{f}_{robot_i}}{\partial \boldsymbol{x}_{eq}} \Delta \boldsymbol{x}_{eq} \right) - \hat{\boldsymbol{n}}_j^T \left(\boldsymbol{f}_{robot_j} + \frac{\partial \boldsymbol{f}_{robot_j}}{\partial \boldsymbol{x}_{eq}} \Delta \boldsymbol{x}_{eq} \right) \right\|^2 \tag{9}$$

$$\textbf{subject to}:$$

$$\boldsymbol{q}_{min} \leq \quad \boldsymbol{q} + \frac{\partial \boldsymbol{q}}{\partial \boldsymbol{x}_{eq}} \Delta \boldsymbol{x}_{eq} \quad \leq \boldsymbol{q}_{max} \tag{10}$$

$$-\Delta x_{eq_{step}} \leq \quad \hat{\boldsymbol{s}}_j^T \Delta \boldsymbol{x}_{eq_i} \quad \leq \Delta x_{eq_{step}} \qquad \forall i, j \tag{11}$$

$$\mu \hat{\boldsymbol{n}}_i^T (\boldsymbol{f}_{robot_i} + \frac{\partial \boldsymbol{f}_{robot_i}}{\partial \boldsymbol{x}_{eq}} \Delta \boldsymbol{x}_{eq}) \leq \quad \hat{\boldsymbol{c}}_{i,j}^T \left(\boldsymbol{I} - \hat{\boldsymbol{n}}_i \hat{\boldsymbol{n}}_i^T \right) (\boldsymbol{f}_{robot_i} + \frac{\partial \boldsymbol{f}_{robot_i}}{\partial \boldsymbol{x}_{eq}} \Delta \boldsymbol{x}_{eq}) \quad \leq -\mu \hat{\boldsymbol{n}}_i^T (\boldsymbol{f}_{robot_i} + \frac{\partial \boldsymbol{f}_{robot_i}}{\partial \boldsymbol{x}_{eq}} \Delta \boldsymbol{x}_{eq}) \qquad \forall i, j \tag{12}$$

$$f_{break} < \quad -\hat{\boldsymbol{n}}_i^T (\boldsymbol{f}_{robot_i} + \frac{\partial \boldsymbol{f}_{robot_i}}{\partial \boldsymbol{x}_{eq}} \Delta \boldsymbol{x}_{eq}) \quad < f_{discomfort} \qquad \forall i \tag{13}$$

$\alpha, \beta_i, \gamma_{i,j}$	Scalar weights for the multi-objective cost function
n	The number of contact locations with one end effector per contact location
$\boldsymbol{q}$	Current joint angles for the human limb
$\Delta \boldsymbol{q}_{des}$	Desired change in the joint angles for the human limb
$\Delta \boldsymbol{x}_{eq}$	Change in the equilibrium positions of the robot's end effectors
$\Delta x_{eq_{step}}$	Maximum allowed change for the equilibrium positions of the robot's end effectors per time step
$\boldsymbol{q}_{min}$	Minimum allowed joint angles for the human limb
$\boldsymbol{q}_{max}$	Maximum allowed joint angles for the human limb
$\boldsymbol{f}_{robot_i}$	Current contact force applied by the robot to the human limb at contact i
μ	Coefficient of friction for contact between the robot's end effectors and the human limb
$\hat{\boldsymbol{n}}_i$	Unit vector normal to the surface of the human limb at contact i that points away from the interior of the limb
$\hat{\boldsymbol{c}}_{i,j}$	Unit vector tangent to the surface of the human limb at contact i and, if $d = 3$, the j^{th} sample from a circle around the contact location (i.e., sampled friction cone)
$\hat{\boldsymbol{s}}_j$	The j^{th} unit vector sampled from the unit sphere ($d = 3$) or unit circle ($d = 2$) corresponding to equal magnitude changes in the equilibrium positions of the end effectors
f_{break}	Minimum compressive force allowed between the robot's end effector and the human limb to avoid breaking contact (e.g., disallows adhesive contact forces)
$f_{discomfort}$	Maximum compressive force allowed between the robot's end effector and the human limb to avoid causing discomfort

Fig. 2. The quadratic programming problem for the general controller.

2) Solving for $\frac{\partial \boldsymbol{f}_{robot}}{\partial \boldsymbol{x}_{eq}}$: To find $\frac{\partial \boldsymbol{f}_{robot}}{\partial \boldsymbol{x}_{eq}}$, we directly take the derivative of $\boldsymbol{f}_{robot}$ written as

$$\boldsymbol{f}_{robot} = \boldsymbol{K}(\boldsymbol{x}_{eq} - \boldsymbol{x}(\boldsymbol{q})) \tag{18}$$

which for our contact model without sliding results in

$$\frac{\partial \boldsymbol{f}_{robot}}{\partial \boldsymbol{x}_{eq}} = \boldsymbol{K}(\boldsymbol{I} - \boldsymbol{J} \frac{\partial \boldsymbol{q}}{\partial \boldsymbol{x}_{eq}}). \tag{19}$$

D. Quadratic Programming Problem

Our approach uses model predictive control (MPC) with a time horizon of length one. Our model corresponds with the discrete-time state-space representation[1] $\mathtt{x}[k+1] = \mathtt{A}[k]\mathtt{x}[k] + \mathtt{B}[k]\mathtt{u}[k]$, where the state vector at the current time step $\mathtt{x}[\mathtt{k}] = \begin{bmatrix} \boldsymbol{q} \\ \boldsymbol{f} \end{bmatrix}$, the state matrix $\mathtt{A} = \boldsymbol{I}$ for all time steps, the input matrix at the current time step $\mathtt{B}[\mathtt{k}] = \begin{bmatrix} \frac{\partial \boldsymbol{q}}{\partial \boldsymbol{x}_{eq}} \\ \frac{\partial \boldsymbol{f}_{robot}}{\partial \boldsymbol{x}_{eq}} \end{bmatrix}$, and the input vector at the current time step $\mathtt{u}[\mathtt{k}] = \Delta \boldsymbol{x}_{eq}$.

For the task of moving the human limb, we want the robot to attain a desired change to the limb's joint angles, $\Delta \boldsymbol{q}_{des}$, while controlling the applied forces. At each time step, the robot creates a convex optimization problem. The

[1] To avoid confusion, we have written the conventional discrete-time state-space equation using distinct notation without boldface.

optimization finds changes to the commanded equilibrium positions for the robot's end effectors, $\Delta \boldsymbol{x}_{eq}$, that result in the lowest cost relative to an objective function subject to constraints. We base the objective function on the predicted forces the robot would apply to the human limb, $\boldsymbol{f}_{robot} + \frac{\partial \boldsymbol{f}_{robot}}{\partial \boldsymbol{x}_{eq}} \Delta \boldsymbol{x}_{eq}$, and the predicted joint angles for the human limb, $\boldsymbol{q} + \frac{\partial \boldsymbol{q}}{\partial \boldsymbol{x}_{eq}} \Delta \boldsymbol{x}_{eq}$. In general, the controller searches for actions that are predicted to achieve the desired changes to the joint angles of the human limb while keeping the applied forces low and distributed across the contacts, subject to constraints on the human limb's predicted joint angles, the robot's commanded equilibrium positions, and the predicted contact forces.

1) Quadratic Cost Function: Specifically, we formulate the quadratic programming problem shown in Fig. 2, which has a quadratic cost function and linear constraints. The α-weighted term of the cost function (Eq. 7) penalizes actions based on the difference between the predicted change in human joint angles and the desired change in the human joint angles, $\Delta \boldsymbol{q}_{des}$. The β-weighted terms of the cost function (Eq. 8) penalize actions based on the predicted magnitudes of the forces applied by the robot, thereby encouraging solutions that result in lower contact force magnitudes. The γ-weighted terms of the cost function (Eq. 9) penalize actions based on the differences between the normal forces applied by the end effectors in order to distribute the applied normal forces more evenly. For example, this can be advantageous when a robot has multiple contacts on a single link of the human limb.

2) Linear Constraints: We express the first set of linear constraints (Eq. 10) using element-wise inequalities to place limits on the human limb's joint angles in order to avoid hyperextension, or joint angles that could result in discomfort. The second set of constraints (Eq. 11) limit the size of the change in the equilibrium positions, $\Delta \boldsymbol{x}_{eq_i}$, by projecting proposed changes onto unit vectors sampled from a unit circle or unit sphere, depending on the dimensionality of the equilibrium positions (i.e., $d = 2$ or $d = 3$). The third and fourth sets of constraints (Eq. 12 and Eq. 13) seek to ensure that the robot's end effectors do not break contact or slip along the surface of the limb, and thereby match the contact model we used in our derivation. The third set of constraints (Eq. 12) use a Coulomb friction model to limit the predicted shear forces applied by the robot's end effectors in one direction for $d = 2$ or in directions evenly sampled around a circle for $d = 3$. The fourth set of constraints (Eq. 13) limit the predicted forces applied normal to the surface of the human limb in order to avoid situations that would cause discomfort or result in the robot's end effector breaking contact with the human limb (e.g., disallowing adhesive forces).

IV. Evaluation

We evaluated our method using a simulated robot with two end effectors and a simulated human body in a supine position (see Fig. I). The robot's task was to reposition the human body's right leg to a desired configuration and then return the leg to its starting configuration. The human leg had two rotary joints, one at the hip and one at the knee.

A. Controller for Lifting a Human Leg

Using the general methods we have presented in this paper, we implemented a controller to enable a robot with two end effectors to reposition the leg's thigh and shank by rotating them around the hip and knee joints. The leg controller used a planar model of the leg with two degrees of freedom, one at the hip and one at the knee, so that $m = 2$, $d = 2$, and $n = 2$. We modeled all the links as rectangles in 2D (i.e., cylinders in 3D). We set $\boldsymbol{h}(\boldsymbol{q}) = 0$, since the simulated human body consisted of rigid bodies connected via pin joints (i.e., a ragdoll model). We found $\boldsymbol{g}(\boldsymbol{q})$ (Eq. 20), $\boldsymbol{x}_{thigh}$ (Eq. 21), $\boldsymbol{x}_{shank}$ (Eq. 22), and $\boldsymbol{J}_i$ using conventional methods with results similar to textbook examples, such as the ”Planar Elbow Manipulator” in [22].

$$\boldsymbol{g}^T = \begin{bmatrix} \frac{1}{2}(L_t m_t \cos(q_t)) + (L_s \cos(q_s) + 2L_t \cos(q_t))m_s)g_a \\ \frac{1}{2} L_s g_a m_s \cos(q_s) \end{bmatrix} \tag{20}$$

$$\boldsymbol{x}_t^T = \begin{bmatrix} r_t \cos(-\frac{1}{2}\pi + q_t) + d_t \cos(q_t) \\ r_t \sin(-\frac{1}{2}\pi + q_t) + d_t \sin(q_t) \end{bmatrix} \tag{21}$$

$$\boldsymbol{x}_s^T = \begin{bmatrix} r_s \cos(-\frac{1}{2}\pi + q_s) + d_s \cos(q_s) + L_t \cos(q_t) \\ r_s \sin(-\frac{1}{2}\pi + q_s) + d_s \sin(q_s) + L_t \sin(q_t) \end{bmatrix} \tag{22}$$

In these equations, q_t and q_s are the angles of the thigh and shank with respect to horizontal; g_a is the gravitational acceleration; m_t, L_t, and r_t are the mass, length, and radius of the thigh cylinder; m_s, L_s, and r_s are the mass, length, and radius of the shank cylinder; and d_t and d_s are the lengths along the thigh and shank at which contact is made. Using $\boldsymbol{g}(\boldsymbol{q})$, $\boldsymbol{J}_i^T$, and $\boldsymbol{K}_i$, we solved for $\frac{\partial \boldsymbol{q}}{\partial \boldsymbol{x}_{eq}}$ and $\frac{\partial \boldsymbol{f}_{robot}}{\partial \boldsymbol{x}_{eq}}$ symbolically with SageMathCloud [27] and Mathematica [28]. We set $\alpha = 500000000$, $\beta_i = 500$, $\gamma_{i,j} = 1$, $\hat{\boldsymbol{s}}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$, $\hat{\boldsymbol{s}}_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$, $\Delta x_{eq_{step}} = 0.01$ m, $\mu = 0.15$, $f_{break} = 5$ N and $f_{discomfort} = 150$ N for all our trials.

We set the cost function weights (α, β_i, and $\gamma_{i,j}$) based on empirical performance. The angular term weight, α, is much larger than the force term weight, β_i. This is due to the angular term representing differences between small changes in joint angles, while the force term represents the predicted force, rather than just a change in the force. We set μ to be lower than the friction parameter used by the physics simulator to provide a safety margin. The maximum force limit, $f_{discomfort} = 150$ N, corresponds with less than 10,000 Pa of uniformly distributed pressure across the robot's end effector. This is small compared to the average female pain threshold of 290,000 Pa from [29] based on contact with the hand. We set $\boldsymbol{q}_{min}$ and $\boldsymbol{q}_{max}$ to reflect the typical range of motion of a human knee. During the trials, we provided the controllers with the current configuration of the leg, the current positions of the end effectors, and the current forces applied by the end effectors. Between trials, we changed $\boldsymbol{K}_i$, the masses of the links, and the dimensions for the links of the leg models. We implemented the controllers in Python and used the OpenOpt framework [30] to solve the quadratic programming problems at each time step.

B. Trajectory Generation and Trajectory Following

When given a goal configuration, $\boldsymbol{q}_{goal}$, the system first generates a trajectory, $\boldsymbol{T}$, consisting of a sequence of configuration waypoints, $\boldsymbol{T} = [\boldsymbol{q}_1, \boldsymbol{q}_2, ...\boldsymbol{q}_{goal}]$. The waypoints, $\boldsymbol{T}_i$, are evenly spaced with respect to each of the limb's joints. The largest angular step between waypoints is less than or equal to a maximum step size, q_{step}. The simulated robot attempts to follow the waypoint trajectory, $\boldsymbol{T}$, using the limb repositioning controller.

At any iteration of the system, a particular waypoint, $\boldsymbol{T}_i$, is active. While a waypoint $\boldsymbol{T}_i$ is active, the system solves the quadratic programming problem with $\Delta\boldsymbol{q}_{des} = \boldsymbol{T}_i - \boldsymbol{q}$, where $\boldsymbol{q}$ is the limb's current configuration. It then sets the Cartesian equilibrium positions for the robot to $\boldsymbol{x}_{eq} + \Delta\boldsymbol{x}_{eq}$, where $\Delta\boldsymbol{x}_{eq}$ results from the optimization . Except for the last waypoint of the trajectory, once each and every limb joint angle is within a threshold, q_{thresh}, of the current waypoint, $\boldsymbol{T}_i$, the next waypoint, $\boldsymbol{T}_{i+1}$, becomes active. For the last waypoint of the trajectory, $\boldsymbol{q}_{goal}$, the system uses a more stringent threshold, $q_{thresh_{goal}}$, and either starts to follow a new trajectory or completes the trial upon meeting this threshold.

For the trials in our evaluation, the robot must reposition the leg from a supine configuration, $\boldsymbol{q}_{supine}$, to a lifted configuration, $\boldsymbol{q}_{lifted}$, and then back to the supine configuration, $\boldsymbol{q}_{supine}$. For all the trials in our evaluation, we set $q_{step} = 0.01$ rad (0.573°), $q_{thresh} = 0.05$ rad (2.865°), and $q_{thresh_{goal}} = 0.01$ rad (0.573°). In general for this paper, we express all angles with respect to a world frame with 0 rad (0°) pointing distally and horizontally along the platform. We allowed the system to iterate at a maximum of 3 Hz with respect to simulated time.

C. The Simulated Robot

The simulated robot consists of two end effectors each with three actively controlled degrees of freedom. These degrees of freedom move the end effectors in a vertical plane aligned with the leg's range of motion. They enable an end effector to translate vertically and horizontally, and also rotate to maintain contact with the leg. Each end effector is a 1 kg rectangular box with a 0.15 m x 0.104 m contact surface, which is comparable to a human hand. Our model predictive controller for limb repositioning commands the equilibrium positions of the end effectors, $\boldsymbol{x}_{eq_i}$, at 3 Hz. A low-level Cartesian impedance controller that runs at 1 kHz uses these equilibrium positions to command the translational forces, $\boldsymbol{f}_{imp_i}$, applied to the centers of mass of the two end effectors.

$$\boldsymbol{f}_{imp_i} = \boldsymbol{K}_i(\boldsymbol{x}_{eq_i} - \boldsymbol{x}_i) + \boldsymbol{D}_i(\dot{\boldsymbol{x}}_i) + \hat{\boldsymbol{G}}, \tag{23}$$

where $\hat{\boldsymbol{G}}$ compensates for the constant gravitational force at the center of mass of each of the end effectors. The nominal values of the stiffness matrix, $\boldsymbol{K}_i$, and damping matrix, $\boldsymbol{D}_i$, are $1000\boldsymbol{I}$ N/m and $600\boldsymbol{I}$ Ns/m, respectively. The controller uses these nominal values in the goal variation simulation trials V-A. They are then varied in further tests V-B.

We use two independent derivative controllers to dampen the movement of the end effectors as they rotate with the leg links.

$$\tau_{ee} = d\dot{\theta} \tag{24}$$

where τ_{ee} is the commanded torque around the end effector's center of mass, $\dot{\theta}$ is the current angular velocity of the end effector, and d is the derivative gain with $d = 20$ Nms/rad.

D. The Simulated Task

We performed tests of our controller in simulation using two end effectors and a simulated human ragdoll. We placed one end effector on the thigh link and one on the shank link. The simulated ragdoll leg shapes resemble human leg shapes, however the controller's model uses link shapes with rectangular cross sections. Likewise, in the physical simulation, the geometry of the ragdoll's legs and the mass of the ragdoll's foot resulted in the centers of mass not being exactly halfway along the lengths of the thigh and the shank, and thus differed from the centers of mass in the controller's planar model. These differences between the actual leg and the controller's model of the leg resulted in model error.

We used Gazebo Simulator version 1.8.3 to perform our evaluation. We constructed the simulated human ragdoll based on the dimensions of a 50 percentile male from [31], [32], and [33]. The length and mass of the thigh were ≈ 0.43 m and 12.3 kg, and the length and mass of the shank were ≈ 0.41 m and 4.95 kg. The hip, knee, and ankle joints of the ragdoll had joint damping values of 1 Nms/rad The ragdoll rests on top of a platform in part to test the feasibility of our fixed-base assumption for the human limb. In our tests, the trunk did not move, which suggests that modeling the human limb as having a fixed base can be a useful approximation.

Prior to each trial, a modified version of our controller moved the leg to the starting goal configuration, $q_{t,goal} = 0.2$ rad (11.459° relative to horizontal) and $q_{s,goal} = -0.1$ rad (−5.730° relative to horizontal) and held it there. This resulted in the full weight of the leg resting on the end effectors, no constraints being violated, and the system being in static equilibrium. At the start of each trial, we positioned the center of the robot end effectors at 0.2 m distal from the hip joint and 0.198 m distal from the knee joint. In the physics simulator, we set the coefficient of friction between the robot end effectors and the human leg to 0.5 for all trials.

E. Evaluation Criteria

We defined task success as lifting the leg to within a threshold of 0.01 rad (0.573°) around the goal angles in less than the amount of time it would take a trajectory with constant angular velocity averaging 0.017 rad/s (1° 1/s) to reach the larger of the two goal angles. Task success also required lowering the leg to within a threshold of 0.01 rad (0.573°) around the initial angles within the same amount of time as the lifting task, and not allowing the end effectors to slip along the leg more than 1.5 cm of

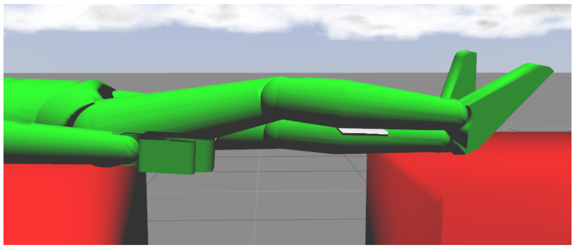

Fig. 3. Initial configuration for all trials. The robot must move the leg from this configuration to a goal configuration, hold the goal configuration, and then move the leg back to this configuration.

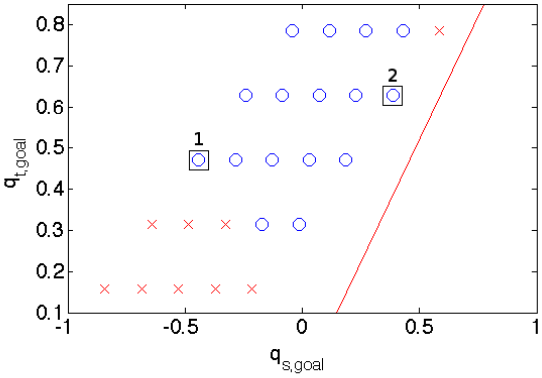

Fig. 4. Visualization of the results from varying the goal configuration. The y-axis is the goal angle for the thigh. The x-axis is the goal angle for the shank. The blue O's represent goals with which the system succeeded. The red X's represent goals with which the system failed. The squared in goals labeled 1 and 2 represent the two goal configurations we performed further tests on. Goals to the right of the red line hyperextend the knee.

distance away from their starting positions. For a controller to perform well, it should also move the leg smoothly to desired configurations in reasonable lengths of time while applying appropriate forces. To assess these properties, we calculated the mean amount of slipping, the mean linear fit, and the mean maximum force. Since we used joint angle trajectories that were linear in time, we used a least squares linear fit of the leg joint angles to quantify the smoothness of the controlled movement of the leg. We report the quality of a linear fit using the coefficient of determination (R^2) of each leg angle over time for each trajectory. A perfectly smooth trajectory would give $R^2 = 1$.

V. Results and Discussion

A. Varying the Goal Configuration

First, we evaluated the ability of the system to achieve various goal configurations for the simulated human leg (i.e., the thigh angle and shank angle). We conducted 25 trials, each with a distinct goal configuration (see Fig. 4). The blue circle indicates a success as defined by the criteria for task success described above and the red x indicates a failure. The red line to the right of the graph represents the joint limit constraint where $q_{s,rel,max} = -0.01$ rad ($-0.573°$) relative to the thigh angle. We only selected goal configurations that did not hyperextend the knee, which is why all of the trials are to the left of the red line.

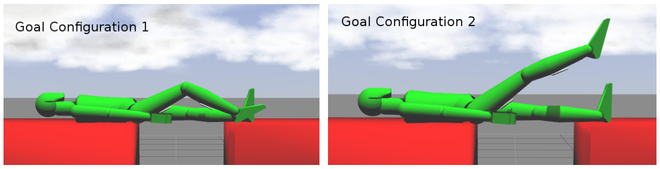

Fig. 5. The two goal configurations used for further testing (Left: Goal 1, Right: Goal 2).

The controller had difficulty primarily when the thigh goal angle was close to zero. With the thigh goal angle set at 0.157 rad (8.995°), the controller failed to complete the task because the red support platform prevented the goal shank angles from being reached. Similarly, the support platform resulted in the three failures for a thigh angle of 0.314 rad (17.991°). Our current system follows a simple straight line trajectory in joint space. Trajectory planning could potentially avoid collisions with the red platform and allow the controller to reach more goals.

B. Further Testing with Two Goal Configurations

We performed 9 trials for each of 2 goal configurations (a total of 18 trials) that we chose from the successful configurations. During the 9 trials, we varied the end effector stiffness and damping between three settings - low (k = 500 N/m, d = 424 Ns/m), medium (k = 1000 N/m, d = 600 Ns/m), and high (k = 2000 N/m, d = 849 Ns/m). We also varied the modeled leg mass relative to the actual leg mass by $\pm 15\%$ and the modeled leg length relative to the actual leg length by $\pm 15\%$. Figure 5 shows the two goal configurations. The first goal configuration was $q_{t,goal} = 0.628$ rad (35.982° relative to horizontal), $q_{s,goal} = 0.385$ rad (22.059° relative to horizontal) and the second goal configuration was $q_{t,goal} = 0.471$ rad (26.986° relative to horizontal), $q_{s,goal} = -0.443$ rad ($-25.382°$ relative to horizontal).

Table I provides the results for the two goals. For Goal 1, the controller succeeded in 8 out of the 9 trials and performed well in terms of low slipping, high smoothness, and low force relative to the task requirements. The controller failed only when the low stiffness setting was used. For Goal 2, which was closer to the initial configuration of the leg, the controller succeeded in 8 out of 9 trials, and also performed well by our metrics. The controller failed only when the length was varied to $+15\%$.

VI. Conclusion

General purpose robots capable of effectively manipulating the human body with their end effectors could potentially be beneficial for a variety of applications. We have presented a general approach for robotic repositioning of human limbs via model predictive control. While our results in simulation are promising, the extent to which our approach would work with real robots and real human bodies remains an open question. We have presented empirical evidence based on physics simulations to demonstrate the feasibility of our approach. Specifically, we evaluated the performance of a particular controller for the task of repositioning a human

TABLE I

RESULTS FROM FURTHER TESTING WITH TWO GOAL CONFIGURATIONS (M±SD)

	Success	Mean Slip (cm)	Mean Linear Fit (R^2)	Mean Max Force (N)	Max Max Force (N)
Goal 1	8/9 trials	0.51±0.098	0.993±0.005	91.996±10.891	105.025
Goal 2	8/9 trials	0.374±0.120	0.998±0.004	102.927±2.121	105.240

leg. The controller performed well at moving the simulated leg to a number of configurations and did so smoothly, at reasonable speeds, and with appropriate force. It also succeeded in the presence of model error, which bodes well for real-world implementations.

ACKNOWLEDGMENTS

We thank Connor Eaton for his simulation work, and Tapomayukh Bhattacharjee & Yash Chitalia for their feedback. This work was supported in part by the National Institute on Disability, Independent Living, and Rehabilitation Research (NIDILRR), grant 90RE5016-01-00 via RERC TechSAge, and by the Army Telemedicine & Advanced Technology Research Center (TATRC) via Hstar Technologies in support of contract # W81XWH-11-C-0007.

REFERENCES

[1] K. Krishnaswamy, R. Kuber, and T. Oates, "Developing a limb repositioning robotic interface for persons with severe physical disabilities," *Universal Access in the Information Society*, pp. 1–19, 2015.

[2] M. Reddy, S. S. Gill, and P. A. Rochon, "Preventing pressure ulcers: a systematic review," *Jama*, vol. 296, no. 8, pp. 974–984, 2006.

[3] "Patterson medical leg lifters." [Online]. Available: http://www.pattersonmedical.com/

[4] G. Kwakkel, B. J. Kollen, and H. I. Krebs, "Effects of robot-assisted therapy on upper limb recovery after stroke: a systematic review," *Neurorehabilitation and neural repair*, 2007.

[5] R. Riener, L. Lunenburger, S. Jezernik, M. Anderschitz, G. Colombo, and V. Dietz, "Patient-cooperative strategies for robot-aided treadmill training: first experimental results," *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, vol. 13, no. 3, pp. 380–394, 2005.

[6] S. Jezernik, G. Colombo, T. Keller, H. Frueh, and M. Morari, "Robotic orthosis lokomat: A rehabilitation and research tool," *Neuromodulation: Technology at the neural interface*, vol. 6, no. 2, pp. 108–115, 2003.

[7] H. Kazerooni, J.-L. Racine, L. Huang, and R. Steger, "On the control of the berkeley lower extremity exoskeleton (bleex)," in *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*. IEEE, 2005, pp. 4353–4360.

[8] K. Kong and D. Jeon, "Design and control of an exoskeleton for the elderly and patients," *Mechatronics, IEEE/ASME Transactions on*, vol. 11, no. 4, pp. 428–432, 2006.

[9] A. M. Dollar and H. Herr, "Lower extremity exoskeletons and active orthoses: challenges and state-of-the-art," *Robotics, IEEE Transactions on*, vol. 24, no. 1, pp. 144–158, 2008.

[10] R. H. Krishnan and S. Pugazhenthi, "Mobility assistive devices and self-transfer robotic systems for elderly, a review," *Intelligent Service Robotics*, vol. 7, no. 1, pp. 37–49, 2014.

[11] J.-W. Jung, J.-H. Do, Y.-M. Kim, K.-S. Suh, D.-J. Kim, and Z. Z. Bien, "Advanced robotic residence for the elderly/the handicapped: Realization and user evaluation," in *Rehabilitation Robotics, 2005. ICORR 2005. 9th International Conference on*. IEEE, 2005, pp. 492–495.

[12] M. T. Mason, "Progress in nonprehensile manipulation," *The International Journal of Robotics Research*, vol. 18, no. 11, pp. 1129–1141, 1999.

[13] M. Erdmann, "An exploration of nonprehensile two-palm manipulation," *The International Journal of Robotics Research*, vol. 17, no. 5, pp. 485–503, 1998.

[14] O. Khatib, K. Yokoi, K. Chang, D. Ruspini, R. Holmberg, A. Casal, and A. Baader, "Force strategies for cooperative tasks in multiple mobile manipulation systems," in *Robotics Research*. Springer, 1996, pp. 333–342.

[15] M. N. Ahmadabadi and E. Nakano, "A constrain and move approach to distributed object manipulation," *Robotics and Automation, IEEE Transactions on*, vol. 17, no. 2, pp. 157–172, 2001.

[16] A. Gupta and W. H. Huang, "Centralized and distributed algorithms for a cooperative carrying task."

[17] M. Onishi, Z. Luo, T. Odashima, S. Hirano, K. Tahara, and T. Mukai, "Generation of human care behaviors by human-interactive robot riman," in *Robotics and Automation, 2007 IEEE International Conference on*. IEEE, 2007, pp. 3128–3129.

[18] T. Mukai, S. Hirano, H. Nakashima, Y. Kato, Y. Sakaida, S. Guo, and S. Hosoe, "Development of a nursing-care assistant robot riba that can lift a human in its arms," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*. IEEE, 2010, pp. 5996–6001.

[19] T. Mukai, S. Hirano, M. Yoshida, H. Nakashima, S. Guo, and Y. Hayakawa, "Tactile-based motion adjustment for the nursing-care assistant robot riba," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 5435–5441.

[20] M. Ding, R. Ikeura, T. Mukai, H. Nagashima, S. Hirano, K. Matsuo, M. Sun, C. Jiang, and S. Hosoe, "Comfort estimation during liftup using nursing-care robotriba," in *Innovative Engineering Systems (ICIES), 2012 First International Conference on*. IEEE, 2012, pp. 225–230.

[21] O. Mehrez, Z. Zyada, H. S. Abbas, and A. Abo-Ismail, "Modeling and static analysis of a three-rigid-link object for nonprehensile manipulation planning," in *Mechatronics and Automation (ICMA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 1441–1446.

[22] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot modeling and control*. Wiley New York, 2006, vol. 3.

[23] N. Hogan and S. P. Buerger, "Impedance and interaction control," 2005.

[24] C. Ott, *Cartesian impedance control of redundant and flexible-joint robots*. Springer, 2008.

[25] R. Riener and T. Edrich, "Identification of passive elastic joint moments in the lower extremities," *Journal of biomechanics*, vol. 32, no. 5, pp. 539–544, 1999.

[26] A. Jain, M. D. Killpack, A. Edsinger, and C. C. Kemp, "Reaching in clutter with whole-arm tactile sensing," *The International Journal of Robotics Research*, p. 0278364912471865, 2013.

[27] W. Stein *et al.*, *Sage Mathematics Software*, The Sage Development Team, 2015, http://www.sagemath.org https://cloud.sagemath.com/.

[28] *Mathematica 10.0*, Wolfram Research, Inc., 2014.

[29] L. S. Chesterton, P. Barlas, N. E. Foster, G. D. Baxter, and C. C. Wright, "Gender differences in pressure pain threshold in healthy humans," *Pain*, vol. 101, no. 3, pp. 259–266, 2003.

[30] D. Kroshko, "OpenOpt: Free scientific-engineering software for mathematical modeling and optimization," 2007–. [Online]. Available: http://www.openopt.org/

[31] A. R. Tilley and S. B. Wilcox, *The measure of man and woman : human factors in design*. New York: Wiley, 2002. [Online]. Available: http://opac.inria.fr/record=b1129318

[32] C. L. Ogden, N. C. for Health Statistics (US) *et al.*, *Mean body weight, height, and body mass index: United States 1960-2002*. Department of Health and Human Services, Centers for Disease Control and Prevention, National Center for Health Statistics, 2004.

[33] P. De Leva, "Adjustments to zatsiorsky-seluyanov's segment inertia parameters," *Journal of biomechanics*, vol. 29, no. 9, pp. 1223–1230, 1996.