Towards using Musculoskeletal Models for Intelligent Control of Physically Assistive Robots

Marc G. Carmichael, Dikai Liu

***Abstract*— With the increasing number of robots being developed to physically assist humans in tasks such as rehabilitation and assistive living, more intelligent and personalized control systems are desired. In this paper we propose the use of a musculoskeletal model to estimate the strength of the user, from which information can be utilized to improve control schemes in which robots physically assist humans.**

An optimization model is developed utilizing a musculoskeletal model to estimate human strength in a specified dynamic state. Results of this optimization as well as methods of using it to observe muscle-based weaknesses in task space are presented. Lastly potential methods and problems in incorporating this model into a robot control system are discussed.

I. INTRODUCTION

There has been much research and development in recent decades into robotics which physically assist humans in tasks such as rehabilitation [1], [2], [3], [4] and assisted living [5], [6], [7], [8]. Because these tasks are generally quite taxing and costly due to the need of intensive or lengthy sessions with health care professionals, and demand is expected to increase with an aging population, efficacious robotic solutions are desirable.

A means to improve physically interactive robotic systems is to personalize the control system to the capabilities of the user. For example robots which assist in reaching and lifting tasks could use basic kinematic and strength models of the user to determine poses in which they are weaker and require greater assistance. A problem is that such simplistic models of the user do not capture the complex characteristics of the human body. This problem is exacerbated in the field of rehabilitative and assistive robotics where the user typically has a disability, and hence their capabilities are unlikely to be well represented by a model of the average person.

A solution is to use a model which can adequately represent the capabilities of the user. Imagine a patient with motor-neuron disease affecting only a small group of muscles. This patient may be weak in certain areas of the task space, yet have almost full strength in others. Using a simplistic model of their capabilities based at the joint level may not adequately capture the complexity of their disability, whereas a more complex model based at the muscular level might.

In this paper we propose an optimization model which utilizes musculoskeletal models to approximate the capabilities of the user. We focus on modelling users with a disability represented at the muscular level, and how the model can be used to obtain information regarding the user's disability in task space which could then be incorporated into the control of a robot. Although musculoskeletal models have been used in robotics previously [9], [10], to our knowledge musculoskeletal models have not been used in this manner.

M.G. Carmichael and D. Liu are with the Centre for Autonomous Systems (CAS), Faculty of Engineering and IT, University of Technology Sydney (UTS), NSW 2007, Australia. Email: marc.g.carmichael@eng.uts.edu.au, dkliu@eng.uts.edu.au

II. MUSCULOSKELETAL MODELING

Musculoskeletal models combine models of the skeleton, muscles and tendons, allowing the highly complex relationships of these systems to be examined. The following section details elements of musculoskeletal models used in this paper, however many different modelling methods exist and will not be examined in detailed here. For further details readers are directed towards references [11], [12], [13], [14].

A. Rigid Body Kinematics and Dynamics

The kinematics and dynamics of the rigid body system is analyzed in a similar fashion to a robotic manipulator. The skeleton is modelled with bones represented as rigid links connected by mechanical joints. A vector of generalized coordinate positions $\mathbf{q} = [q_1, q_2, \ldots, q_k]^T$ is used to define the pose of the model containing k generalized coordinates where T is transpose.

With the kinematic system defined the Jacobian $\mathbf{J_v}$ (1) can be derived relating joint-space generalized coordinate velocities to cartesian linear velocity of a rigid body point (in this paper we do not consider task space angular velocities). The Jacobian for any body point can be calculated, but here we calculate it at the *end effector*, which typically is the hand for upper limb models or foot for lower limb.

$$\mathbf{J_v}(\mathbf{q}) = \begin{bmatrix} \frac{\partial x}{\partial q_1} & \frac{\partial x}{\partial q_2} & \cdots & \frac{\partial x}{\partial q_k} \\ \frac{\partial y}{\partial q_1} & \frac{\partial y}{\partial q_2} & \cdots & \frac{\partial y}{\partial q_k} \\ \frac{\partial z}{\partial q_1} & \frac{\partial z}{\partial q_2} & \cdots & \frac{\partial z}{\partial q_k} \end{bmatrix} \tag{1}$$

Each link is given mass and internal properties equivalent to that of its respective body segment. Expressed in joint space the dynamic equation of the system is (2) where $\mathbf{H}(\mathbf{q})$ is the mass matrix, $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ are centrifugal and Coriolis effects, and $\boldsymbol{\tau}_\mathbf{g}(\mathbf{q})$ are joint torques due to gravity. $\boldsymbol{\tau}$ is a vector of generalized joint torques resulting from muscle forces, and $\mathbf{f_E}$ represents an external force acting on the end effector.

$$\mathbf{H}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \boldsymbol{\tau}_\mathbf{g}(\mathbf{q}) = \boldsymbol{\tau} + \mathbf{J_v}^T \mathbf{f_E} \tag{2}$$

B. Musculotendon unit (MTU) model

Actuation is modelled by musculotendon units (MTUs) representing muscles and the tendons joining them to the skeleton. Several models for MTUs exits [11], [15], [16], most of which are derived from the Hill muscle model [17] which relates maximum muscle tension to contraction velocity. The models allow calculation of the MTU active and passive tensile force in a given state, with the total force being the sum of the two.

Active and passive muscle fibre forces are both functions of muscle fibre length and velocity. Functions $\tilde{f}^P$ and $\tilde{f}^A$ are used to relate normalized passive and active force to the normalized muscle fibre length and velocity. Since muscle fibre length and velocity are functions of $\mathbf{q}$ and $\dot{\mathbf{q}}$ then the MTU tensile force can be expressed as (3), where F^0 is the muscle's maximum isometric force and a is muscle activation ranging from 0 to 1.

$$f_i = \underbrace{\tilde{f}_i^P(\mathbf{q},\dot{\mathbf{q}})F_i^0}_{\text{passive} f_P} + \underbrace{\tilde{f}_i^A(\mathbf{q},\dot{\mathbf{q}},a_i)F_i^0 a_i}_{\text{active} f_A} \tag{3}$$

C. Musculotendon unit routes and moment arms

The routes MTUs make in the body can be defined by a set of via points fixed in reference to the bones. The MTU length l is then the total Euclidean distance of the route [18]. Routes can also be wrapped over virtual surfaces representing bones or other body parts [19]. MTU routes effect how the muscles produce joint torque since a muscle's force is a function of its length and velocity, both of which depend on the MTU route. A muscle Jacobian $\mathbf{L}$ can be constructed relating the velocities of m MTUs to the velocities of k generalized coordinates (4). The principal of virtual work shows that the elements of $\mathbf{L}$ are equal to the moment arms of the MTUs about the generalized coordinates.

$$\mathbf{L}(\mathbf{q}) = \begin{bmatrix} \frac{\partial l_1}{\partial q_1} & \frac{\partial l_1}{\partial q_2} & \cdots & \frac{\partial l_1}{\partial q_k} \\ \frac{\partial l_2}{\partial q_1} & \frac{\partial l_2}{\partial q_2} & \cdots & \frac{\partial l_2}{\partial q_k} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial l_m}{\partial q_1} & \frac{\partial l_m}{\partial q_2} & \cdots & \frac{\partial l_m}{\partial q_k} \end{bmatrix} \tag{4}$$

III. Optimization Model for Estimating Strength

Before we can estimate the strength we need to represent muscle forces as joint torques. Expressing muscle activations in column vector $\mathbf{a}$ along with MTU active and passive forces in column vectors $\mathbf{f}_A$ and $\mathbf{f}_P$, active and passive joint torque vectors $\boldsymbol{\tau}_A$ and $\boldsymbol{\tau}_P$ can be calculated respectively (5).

The negative sign in (5) is due to the convention of MTU shortening being positive. Matrix $\mathbf{K}_A$ is the active muscle force gain matrix used to define the MTU force per unit of muscle activation [20] defined as (6).

$$\boldsymbol{\tau} = \big[\underbrace{-\mathbf{L}^T\mathbf{K}_A\mathbf{a}}_{\boldsymbol{\tau}_A}\big] + \big[\underbrace{-\mathbf{L}^T\mathbf{f}_P}_{\boldsymbol{\tau}_P}\big] \tag{5}$$

$$\mathbf{K}_A = \begin{bmatrix} \tilde{f}_1^A F_1^0 & 0 & \cdots & 0 \\ 0 & \tilde{f}_2^A F_2^0 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & \tilde{f}_m^A F_m^0 \end{bmatrix} \tag{6}$$

Based on the musculoskeletal model we desire to calculate the maximum strength of a person opposing an external force at their end effector in a given direction. To achieve we create an optimization model with the joint coupling acting as constraints. We define the external force applied at the end effector by its magnitude F_0 and the direction it is applied in by the unit vector $\mathbf{u} = [u_x, u_y, u_z]^T$ in cartesian space. The moment arms of the external force about the generalized coordinates of the model are calculated as $\mathbf{r} = \mathbf{J_v}^T\mathbf{u}$. The optimization we are performing is for a given model state, that is with the generalized coordinates, velocities and accelerations defined. With this state defined the inertial, velocity and gravitational components of dynamic equation (2) can be calculated and combined into a single torque vector for convenience, which we call the bias torque vector $\boldsymbol{\tau}_b$. This results in the new dynamic equation (7).

$$\boldsymbol{\tau}_b = \boldsymbol{\tau}_A + \boldsymbol{\tau}_P + \mathbf{r}F_0 \tag{7}$$

To find the largest magnitude of external force we take one row of (7) and optimize it to find the maximum F_0. Taking the ith row of (7) and rearranging results in objective function (8) which is to be maximized. The objective function input is the vector of muscle activations $\mathbf{a}$. Row i is selected by the row with the largest absolute value of r_i, since small magnitudes cause the objective function to lose integrity.

$$F_0 = \left[\frac{\tau_{bi} - \tau_{Pi}}{r_i}\right] + \left[\frac{[\mathbf{K}_a\mathbf{L}_i]^T}{r_i}\right]\mathbf{a} \tag{8}$$

The objective function alone does not take into account intercoupling of the joints required to ensure that the system satisfies equation (7). It is convenient to formulate the required equality constraints into the form $\mathbf{Ax} = \mathbf{b}$ where $\mathbf{x}$ is the input vector, in this case $\mathbf{a}$. To rearrange (7) into this form we eliminate the scalar F_0 by creating the proportionality constraint (9). This proportionality can be returned to an equality by normalizing each side by dividing by elements corresponding to the ith joint, resulting in (10). Again we choose the ith joint corresponding to the largest moment arm r_i to avoid mathematical degradation.

$$\mathbf{r} \propto \boldsymbol{\tau}_b - \boldsymbol{\tau}_A - \boldsymbol{\tau}_P \tag{9}$$

$$\frac{\mathbf{r}}{r_i} = \frac{\boldsymbol{\tau}_b - \boldsymbol{\tau}_A - \boldsymbol{\tau}_P}{\tau_{bi} - \tau_{Ai} - \tau_{Pi}} \tag{10}$$

Rearranging (10) results in expressions for $\mathbf{A}$ (11) and $\mathbf{b}$ (12). For a system with k generalized coordinates, this implies k constraints making the system over constrained system since only $k-1$ constraints are required. Upon examination it is seen that the ith rows of $\mathbf{A}$ and $\mathbf{b}$ can be removed before optimization since these rows reduce to a constraint of $[0]\mathbf{a} = 0$. Furthermore this constraint can cause the optimization to fail if rounding errors cause either side to not be exactly zero.

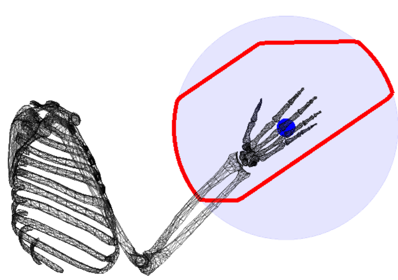
$$\mathbf{A} = \left[(\frac{\mathbf{r}}{r_i})\mathbf{L}_i^T - \mathbf{L}^T \right] \mathbf{K}_a \tag{11}$$

$$\mathbf{b} = \boldsymbol{\tau}_b - \boldsymbol{\tau}_P + \frac{\mathbf{r}}{r_i}(\tau_{Pi} - \tau_{bi}) \tag{12}$$

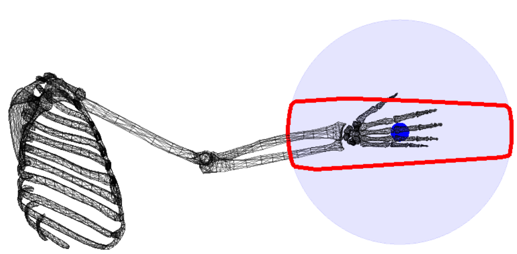
Lastly we consider instances when the optimization can produce unrealistically high strength values due to the moment arms $\mathbf{r}$ being small, resulting in little muscle forces being required to overcome the external force. In reality it is unrealistic to assume that the human body is capable of withstanding loads of vast magnitudes, so an upper limit F_0^{max} is specified. Appropriate values for this limit can be derived based on physiological or occupational health and safety limits, and then used to terminate the optimization process if required.

IV. Results

Figure 1 compares the results of the optimization model applied to the Upper Extremity model [13] for two different poses. Results are obtained by repeatedly calculating the strength of the model as the external force direction unit vector $\mathbf{u}$ is rotated 360 degrees in the sagittal plane. Strength is then plotted in a polar plot about the end effector with the plot radius indicating the calculated strength F_0. The polar plot is plotted in the opposite direction ($-\mathbf{u}$) as to visualize the human's reaction to the external force rather than the external force itself. The shaded circular region represents the strength limit F_0^{max} which was set to 300N.



(a)



(b)

Fig. 1: Strength comparison at different poses

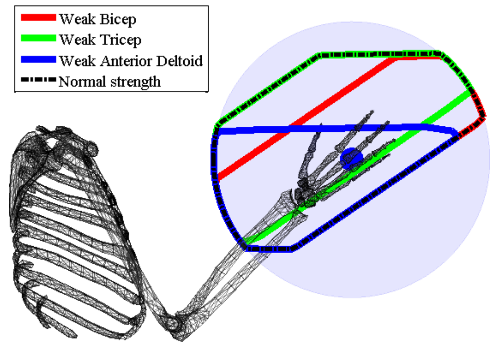


Fig. 2: Strength comparison for different weakened muscles

It can be seen how the strength profile changes as the arm is stretched forward. In doing so the strength in vertical directions reduces, while in the horizontal directions strength is retained, a behavior which can be described as realistic. This shows how the musculoskeletal model captures the complex relationships of strength against pose.

An interesting application of the optimization model is when we observe the effect introducing weaknesses into the muscle space has on task space strength. Figure 2 shows the strength profiles of the same model in the same state, but in three cases where the strengths of individual muscles (biceps, triceps, anterior deltoid) are reduced by 90%, compared to no weakness being introduced. In each weakened case it is seen that in some direction the strength remains equal to that of having no weakness. What is of more interest is that it is clearly seen which directions the muscle weakness does reduce the strength in task space, and how these directions are different for the different muscle groups.

This information provides a valuable insight into how muscle-based weakness is reflected in task space. Another method to observe this weakness is seen in Figure 3. In this figure for each case where the muscle is weakened, the reduction in strength is calculated as a percentage of the original strength of the model with no muscle weakness. This loss in strength is then plotted, with the radius of the shaded region corresponding to a 100% loss in strength.

V. Discussion on using model in robot control

With the users weakness expressed in the task space, the next step is to utilize this information in the robotic control system. This is an area of future research. One method might simply use the percentage of strength lost due to the modelled disability as an *assistance* gain in a lower level control system. Another method might precalculate the areas in the workspace in which the user is weakest, and use virtual force fields to repel the user from these areas. Alternatively a rehabilitation robot might deliberately seek these weak areas in an attempt to maximize the rehabilitation process.

Regardless of how the information is utilized, for the optimization model to be of any use in a robot control system, the ability of the model to map muscle weakness into the user's task space needs to be adequately accurate.

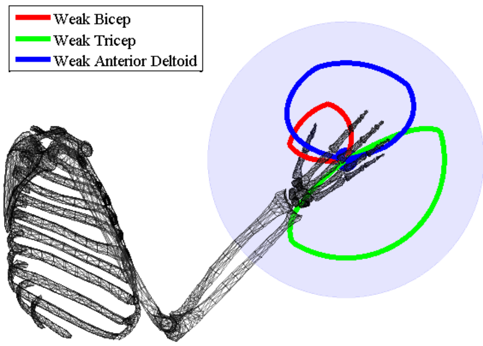


Fig. 3: Perentage of strength decrease for biceps, triceps, and anterior deltoid muscles

This relies heavily on the accuracy of the musculoskeletal model. The optimization model presented in this paper however is independent of the musculoskeletal model used. Future developments in musculoskeletal modelling will result in more accurate representation of the human body. One interesting area in musculoskeletal modelling research is the use of finite element models. This in the future may results in models of higher accuracy. Accuracy also relies on the ability of the individuals disability to be successfully represented in the model.

For use in a real time control scheme the computational time required must be considered. During this research the optimization process was timed to take approximately 15ms, quick enough to be considered for real time control. Calculations which did take a lot of time were the state-dependent musculoskeletal parameters required to be calculated before the optimization could be run. These parameters would take up to 800ms to compute, most of which was spent calculating the muscle Jacobian $\mathbf{L}$, highlighting the complexity of the model which consists of 15 generalized coordinates and 50 muscles. The system used had a Dual Core 2.53GHz processor running Windows 7 performing calculation in C++ via a Matlab mex function. In the future potential steps to improve calculation speed could involve alternate methods of moment-arm calculation or precalculation of the musculoskeletal parameters.

VI. Conclusion

In conclusion we presented an optimization model using a musculoskeletal model to estimate human task space strength. Using this optimization we showed how by defining weakness in the model's muscle space, the effect on strength in the task space can be observed. Such information could be used to improve the control of robots that physically assists the user like in rehabilitation and assistive living applications.

References

[1] H. Krebs, N. Hogan, M. Aisen, and B. Volpe, "Robot-aided neurorehabilitation," *Rehabilitation Engineering, IEEE Transactions on*, vol. 6, no. 1, pp. 75 –87, Mar 1998.

[2] G. Colombo, M. Joerg, R. Schreier, and V. Dietz, "Treadmill training of paraplegic patients using a robotic orthosis," *Journal of Rehabilitation Research and Development*, vol. 37, no. 6, pp. 693 – 700, 2000.

[3] P. S. Lum, C. G. Burgar, P. C. Shor, M. Majmundar, and M. V. der Loos, "Robot-assisted movement training compared with conventional therapy techniques for the rehabilitation of upper-limb motor function after stroke," *Arch. Phys. Med. Rehabil*, vol. 83, pp. 952 – 959, 2002.

[4] L. E. Kahn, M. L. Zygman, W. Z. Rymer, and D. J. Reinkensmeyer, "Robot-assisted reaching exercise promotes arm movement recovery in chronic hemiparetic stroke: a randomized controlled pilot study." *Journal of Neuroengineering and Rehabilitation*, vol. 3, no. 12, Jun 2006.

[5] H. H. Kwee, "Integrated control of manus manipulator and wheelchair enhanced by environmental docking," *Robotica*, vol. 16, no. 5, pp. 491–498, 1998.

[6] T. Rahman, W. Sample, and R. Seliktar, "16 design and testing of wrex," in *Advances in Rehabilitation Robotics*, ser. Lecture Notes in Control and Information Sciences, Z. Bien and D. Stefanov, Eds. Springer Berlin / Heidelberg, 2004, vol. 306, pp. 243–250.

[7] J. Pratt, B. Krupp, C. Morse, and S. Collins, "The roboknee: an exoskeleton for enhancing strength and endurance during walking," *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*, vol. 3, pp. 2430–2435, 26 April - 1 May 2004.

[8] K. Kiguchi, M. H. Rahman, M. Sasaki, and K. Teramoto, "Development of a 3dof mobile exoskeleton robot for human upper-limb motion assist," *Robotics and Autonomous Systems*, vol. 56, no. 56, pp. 678–691, 2008.

[9] O. Khatib, J. Warren, V. D. Sapio, and L. Sentis, "Human-like motion from physiologically-based potential energies," in *On Advances in Robot Kinematics*, J. Lenarčič and C. Galletti, Eds. Kluwer Academic Publishers, 2004, pp. 149 – 163.

[10] C. Fleischer and G. Hommel, "A human–exoskeleton interface utilizing electromyography," *Robotics, IEEE Transactions on*, vol. 24, no. 4, pp. 1552–3098, Aug 2008.

[11] F. E. Zajac, "Muscle and tendon: properties, models, scaling, and application to biomechanics and motor control," *Critical reviews in biomedical engineering*, vol. 17, no. 4, pp. 359–411, 1989.

[12] B. A. Garner and M. G. Pandy, "Musculoskeletal model of the upper limb based on the visible human male dataset," *Computer Methods in Biomechanics and Biomedical Engineering*, vol. 4, no. 2, pp. 93–126, 2001.

[13] K. R. S. Holzbaur, W. M. Murray, and S. L. Delp, "A model of the upper extremity for simulating musculoskeletal surgery and analyzing neuromuscular control," *Annals of Biomedical Engineering*, vol. 33, no. 6, pp. 829–840, June 2005.

[14] S. Delp, F. Anderson, A. Arnold, P. Loan, A. Habib, C. John, E. Guendelman, and D. Thelen, "Opensim: Open-source software to create and analyze dynamic simulations of movement," *Biomedical Engineering, IEEE Transactions on*, vol. 54, no. 11, pp. 1940–1950, Nov 2007.

[15] L. Schutte, M. Rodgers, F. Zajac, and R. Glaser, "Improving the efficacy of electrical stimulation-induced leg cycle ergometry: an analysis based on a dynamic musculoskeletal model," *Rehabilitation Engineering, IEEE Transactions on*, vol. 1, no. 2, pp. 109–125, Jun 1993.

[16] D. G. Thelen, F. C. Anderson, and S. L. Delp, "Generating dynamic simulations of movement using computed muscle control," *Journal of Biomechanics*, vol. 36, no. 3, pp. 321–328, 2003.

[17] A. V. Hill, "The heat of shortening and the dynamic constants of muscle," in *Proceedings of the Royal Society of London. Series B, Biological Sciences*, vol. 126, no. 843, 1938, pp. 136–195.

[18] S. L. Delp and J. P. Loan, "A graphics-based software system to develop and analyze models of musculoskeletal structures," *Computers in Biology and Medicine*, vol. 25, no. 1, pp. 21–34, 1995.

[19] B. A. Garner and M. G. Pandy, "The obstacle-set method for representing muscle paths in musculoskeletal models," *Computer Methods in Biomechanics and Biomedical Engineering*, vol. 3, pp. 1–30, 2000.

[20] V. D. Sapio, J. Warren, O. Khatib, and S. Delp, "Simulating the task-level control of human motion: a methodology and framework for implementation," *The Visual Computer*, vol. 21, no. 5, pp. 289–302, 2005.