1

A Task-Level Biomechanical Framework for Motion Analysis and Control Synthesis

Vincent De Sapio and Richard Chen
Sandia National Laboratories *
USA

1. Introduction

The behavioral richness exhibited in natural human motion results from the complex interplay of biomechanical and neurological factors. The biomechanical factors involve the kinematics and dynamics of the musculoskeletal system while the neurological factors involve the sensorimotor integration performed by the central nervous system (CNS). An adequate understanding of these factors is a prerequisite to understanding the overall effect on human motion as well as providing a means for synthesizing human motion.

The fields of neuroscience, biomechanics, robotics, and computer graphics provide motivation, as well as tools, for understanding human motion. In neuroscience, fundamental scientific understanding drives the motivation to understand human motion, whereas, in biomechanics, clinical applications often form the driving motivation. These clinical applications involve the use of movement analysis and simulation tools to help direct patient rehabilitation as well as predict the effects of surgery on movement.

In addition to the clinical desire to analyze movement there has been an emerging desire in recent years to synthetically generate human-like motion in both simulated and physical settings. In computer graphics this desire is directed toward autonomously generating realistic motion for virtual actors. The intent is to direct these virtual actors using high-level goal directed commands for which low-level motion control is automatically generated.

Motivated by similar desires, the robotics community seeks a high-level control framework for robotic systems. With the recent advent of complex humanoid robots this challenge has grown more demanding. Consistent with their anthropomorphic design, humanoid robots are intended to operate in a human-like manner within man-made environments and to promote interaction with their biological counterparts. To achieve this, common control strategies have involved generating joint space trajectories or learning specific motions, but these approaches require extensive motion planning computations and do not generalize well to related tasks.

1.1 Human motion control

The basic constituents of the human motor system include the biomechanical plant and the CNS. A high-level block diagram, sufficient for our present purposes, is depicted in Fig. 1. Based on some specified task the CNS performs motor planning which culminates in low-level control issued as a motor command to the biomechanical plant. This motor planning

*Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

and control occurs based on the integration of sensory information from proprioceptors distributed throughout the musculoskeletal system. Some knowledge of the biomechanical plant is also assumed to be encoded in the CNS.

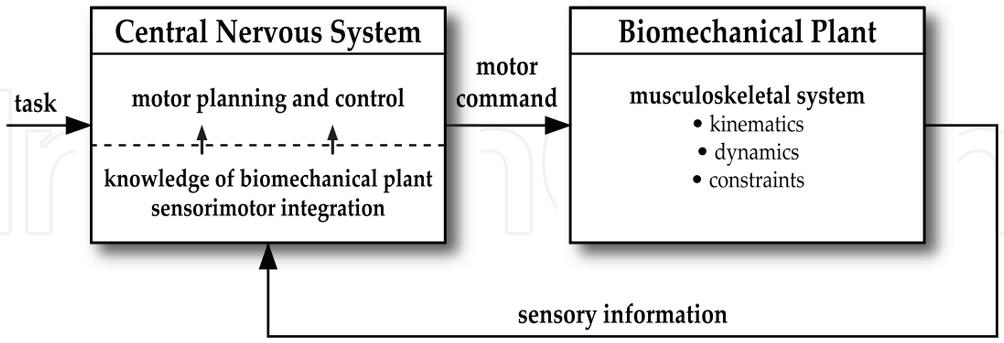


Fig. 1. Motor control involves the task-driven action of the central nervous system (CNS) on the biomechanical plant. Given proprioceptive information the CNS performs motor planning which results in the issuance of motor commands.

While the biomechanical plant can be decomposed and understood in reasonable detail the processes of the CNS are understood more vaguely. As a consequence, while Fig. 1 provides a conceptual framework, with regard to the CNS it lacks enough precision to be useful as a functional model. For this reason it is appropriate to consider some more basic analogs. To this end we will consider the most basic analog which is still useful, that of a joint space model, followed by task/posture analogs.

1.2 Joint space motion control

Joint space control is the earliest and still most common form of feedback control in robotic systems. In this scheme a task is specified in some natural coordinate system associated with the robot and environment. Based on a knowledge of the robot kinematics, the robot controller performs inverse kinematic computations to arrive at a posture or trajectory in terms of joint angles. The joint command is issued to the servo motors which execute the motion.
While this method of controlling a robot is effective it requires the computation of inverse kinematics. Additionally, it does not make use of any knowledge of the robot's dynamics. The method of computed torque is an enhancement of the basic joint space control approach in which the controller does make use of the robot's dynamics. However, the control is still encoded in joint space rather than a more natural task space description. As an analog to the human motor system this joint space encoding may constitute a deficiency since a number of studies, Buneo et al. (2002); Sabes (2000); Scholz & Schöner (1999); Shenoy et al. (2003), suggest a task-oriented spatial encoding of planning and control. Rather than using inverse kinematic transformations this task-oriented encoding is accomplished through visumotor transformations from retinal coordinates to hand- or body-centered spatial coordinates.

1.3 Task/posture motion control

Motivated by evidence for a task-oriented spatial encoding of motion by the CNS we now consider a task/posture control model. This is depicted in Fig. 2 and represents a generalization beyond a strict joint space motor control model. In this case the control is encoded in the same native space in which the task is expressed. This obviates the need for inverse kinematics to *convert* the task description into a joint space description. The dynamics of the robot plant are expressed in task space with a complementary description of the posture

(see Fig. 3). The controller exploits this decomposed structure to yield separate task and posture control terms. As such, the posture term can be chosen to minimize some criterion, consistent with the execution of the task. The motor command can then be issued in the appropriate actuator space (e.g., motor torques).

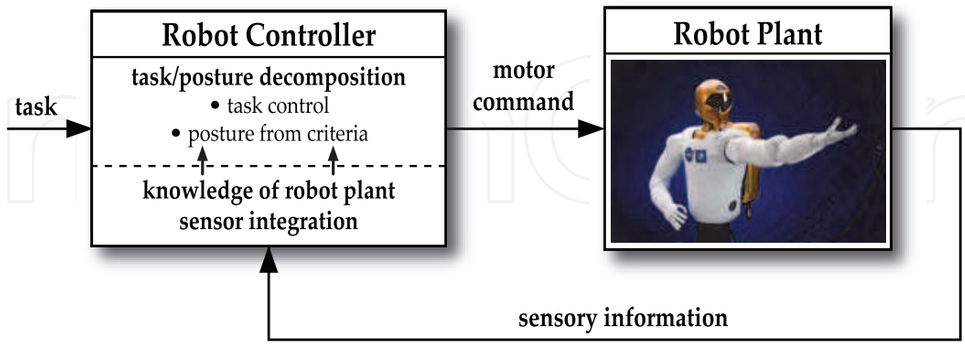


Fig. 2. Task space motion control model where the dynamics are decomposed into complementary task and posture spaces. The posture control can be chosen using an auxiliary criterion which can be optimized consistent with the execution of the task (image courtesy of NASA).

As alluded to earlier this task/posture model represents a more general abstraction than the joint space model, and may be more suitable for purposes of modeling and understanding human motor control. The notions of task and posture are directly applicable to human motion control and, as we shall see, can be specifically interpreted in terms of physiological criteria.

1.4 Task/posture approach for biomechanical systems

Up to this point we have considered robotic control models as a means of addressing human motor control. In a more general sense the challenge of synthesizing low-level human motion control from high-level commands can be addressed by integrating approaches from the

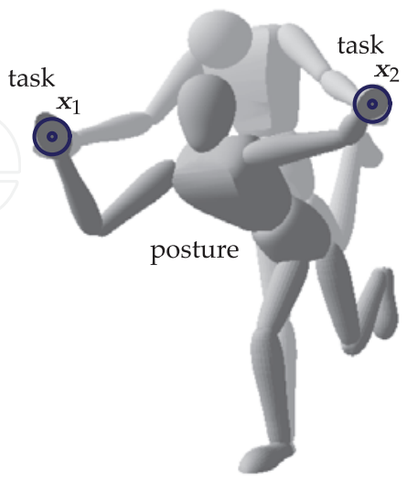


Fig. 3. A task description with complementary task consistent postures. Redundancy with respect to task introduces task dynamics as well as posture dynamics.

biomechanics and robotics communities. The biomechanics community has investigated the phenomenon of neuromuscular dynamics and control through the use of computational muscle models. This characterization allows for the description of muscle strength limitations, activation delays, and overall muscle contraction dynamics. Properly accounting for these characteristics is critical to authentically simulating human motion. In a complementary manner, the robotics community has investigated the task-level feedback control of robots using the operational space approach. This approach recasts the dynamics of the robotic system into a relevant task space description. This provides a natural mechanism for specifying high-level motion commands that can be executed using feedback control.

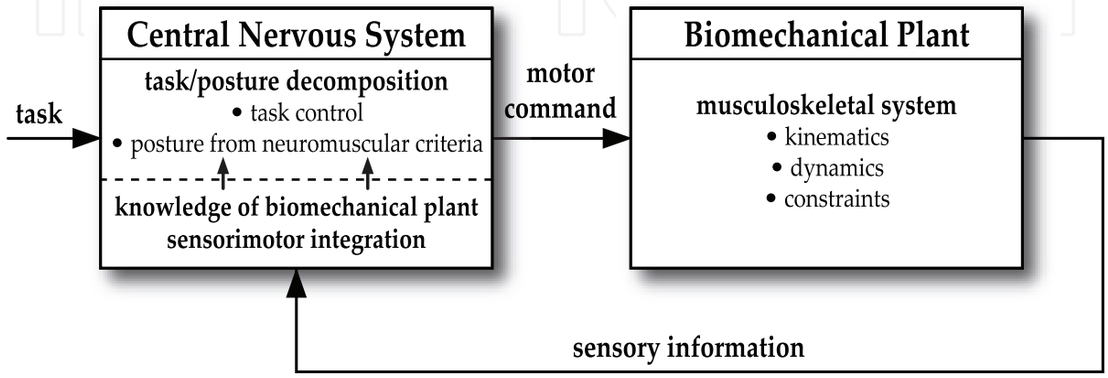


Fig. 4. Task/posture motion control model for biomechanical systems. In addition to task control, neuromuscular criteria are used to control the posture by minimizing neuromuscular cost, consistent with the execution of the task.

Fig. 4 depicts a task/posture model of motor control which integrates robotic and biomechanical approaches. In this model the CNS is seen to affect motor control using task/posture decomposition. While the control is task-driven the task consistent postures are driven by neuromuscular criteria. In other words, while the CNS issues motor commands to achieve some task it is assumed that this is being done in a way that minimizes some neuromuscular cost (subject to the task requirements). While the precise nature of what, if anything, is being minimized by the CNS is difficult to directly infer, computational muscle models can be used to evaluate particular hypothetical effort criteria. Predicted postural behavior associated with minimizing these criteria can then be compared with actual postures from subject trials to validate the applicability of the criteria.
Through the combined utilization of task-level constrained motion strategies and computational muscle models this chapter addresses motion control with application to human motion synthesis. A coherent framework is presented for the management of motion tasks, physical constraints, and neuromuscular criteria. The subsequent sections will address the constituent elements of this framework and will be divided into (i) task-based modeling and analysis and (ii) posture-based modeling and analysis.

2. Task-based modeling and analysis of biomechanical systems

In this section we present a task-based formulation for application to biomechanical systems. In the overall framework this addresses the highlighted element of Fig. 5. The focus is on task control in the presence of constraints.

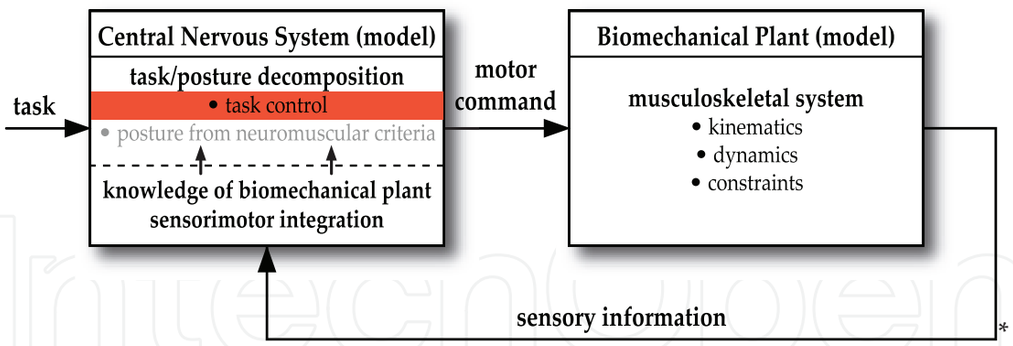


Fig. 5. Task/posture motion control model for biomechanical systems highlighting task control in the presence of constraints. In this chapter the human shoulder complex will be investigated using this approach.

2.1 Configuration space dynamics

The equations of motion for a multibody system that is unconstrained with respect to configuration space can be expressed by the Euler-Lagrange equations,

$$\frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{\boldsymbol{q}}} - \frac{\partial \mathcal{L}}{\partial \boldsymbol{q}} = \boldsymbol{\tau}, \tag{1}$$

where $\mathcal{L} = \mathcal{L}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ is the Lagrangian of the system, $\boldsymbol{q} \in \mathbb{R}^n$ is the vector of generalized coordinates, and $\boldsymbol{\tau} \in \mathbb{R}^n$ is the vector of generalized actuator forces (torques). In standard matrix form the equations of motion can be expressed as,

$$\boldsymbol{M}(\boldsymbol{q})\,\ddot{\boldsymbol{q}} + \boldsymbol{b}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{f}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \boldsymbol{B}(\boldsymbol{q}, \dot{\boldsymbol{q}})^T \boldsymbol{u}, \tag{2}$$

where $\boldsymbol{u} \in \mathbb{R}^k$ is the vector of control inputs, $\boldsymbol{B}(\boldsymbol{q}, \dot{\boldsymbol{q}})^T \in \mathbb{R}^{n \times k}$ is the matrix mapping control inputs to generalized actuator forces, $\boldsymbol{M}(\boldsymbol{q}) \in \mathbb{R}^{n \times n}$ is the configuration space mass matrix, $\boldsymbol{b}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \in \mathbb{R}^n$ is the vector of centrifugal and Coriolis terms, and $\boldsymbol{f}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \in \mathbb{R}^n$ is the vector of generalized applied forces. We will often use a modified and more specialized form of (2) common in robotics,

$$\boldsymbol{M}(\boldsymbol{q})\,\ddot{\boldsymbol{q}} + \boldsymbol{b}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\tau}, \tag{3}$$

where $\boldsymbol{g}(\boldsymbol{q}) \in \mathbb{R}^n$ is the vector of gravity terms. The form of (3) assumes that the generalized actuator forces can be directly interpreted as control inputs; that is, $\boldsymbol{\tau} = \boldsymbol{B}^T \boldsymbol{u} = \boldsymbol{u}$, i.e., $\boldsymbol{B}^T = \mathbf{1}$. Additionally, the generalized applied forces are assumed to be restricted to gravity terms; that is, $\boldsymbol{f}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \boldsymbol{g}(\boldsymbol{q})$.

We now introduce a set of m_C holonomic and scleronomic constraint equations, $\boldsymbol{\phi}(\boldsymbol{q}) = \mathbf{0} \in \mathbb{R}^{m_C}$, that are satisfied on a $p = n - m_C$ dimensional manifold, Q^p, in configuration space, $Q = \mathbb{R}^n$. The gradient of $\boldsymbol{\phi}$ yields the constraint matrix,

$$\boldsymbol{\Phi}(\boldsymbol{q}) = \frac{\partial \boldsymbol{\phi}}{\partial \boldsymbol{q}} \in \mathbb{R}^{m_C \times n}. \tag{4}$$

Adjoining the constraints to (3) by introducing a set of constraint forces yields the dynamic equation in the familiar multiplier form,

$$\boldsymbol{M}\ddot{\boldsymbol{q}} + \boldsymbol{b} + \boldsymbol{g} - \boldsymbol{\Phi}^T \boldsymbol{\lambda} = \boldsymbol{\tau}, \tag{5}$$

subject to,

$$\boldsymbol{\Phi}\ddot{\boldsymbol{q}} + \dot{\boldsymbol{\Phi}}\dot{\boldsymbol{q}} = \boldsymbol{0} \qquad (\boldsymbol{\phi} = \boldsymbol{0}, \quad \boldsymbol{\Phi}\dot{\boldsymbol{q}} = \boldsymbol{0}), \tag{6}$$

where $\boldsymbol{\lambda}$ is a vector of unknown Lagrange multipliers.

2.2 Task space dynamics and control

In the previous section we considered configuration space descriptions of the dynamics of constrained multibody systems. Our objective is to reformulate these descriptions in the context of task space, Khatib (1987); Khatib (1995). This will provide the foundation for constrained task-level control to be discussed in the next section.

Given a branching chain system and defining a set of m task, or operational space, coordinates, $\boldsymbol{x} \in \mathbb{R}^m$ we define the task Jacobian as,

$$\boldsymbol{J}(\boldsymbol{q}) = \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{q}} \in \mathbb{R}^{m \times n}. \tag{7}$$

The generalized force (or control torque) in (3) can then composed as $\boldsymbol{J}^T\boldsymbol{f}$, where $\boldsymbol{f} \in \mathbb{R}^m$ is the task, or operational space, force. In the redundant case an additional term needs to complement the task term in order to realize any arbitrary generalized force. We will refer to this term as the null space term and it can be composed as $\boldsymbol{N}^T\boldsymbol{\tau}_o$, where $\boldsymbol{\tau}_o$ is an arbitrary generalized force and $\boldsymbol{N}(\boldsymbol{q})^T \in \mathbb{R}^{n \times n}$ is the null space projection matrix. We then have the following set of unconstrained task, or operational space, equations of motion, Khatib (1987),

$$\boldsymbol{\Lambda}(\boldsymbol{q})\,\ddot{\boldsymbol{x}} + \boldsymbol{\mu}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{p}(\boldsymbol{q}) = \boldsymbol{f}, \tag{8}$$

where $\boldsymbol{\Lambda}(\boldsymbol{q}) \in \mathbb{R}^{m \times m}$ is the operational space mass matrix, $\boldsymbol{\mu}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \in \mathbb{R}^m$ is the operational space centrifugal and Coriolis force vector, and $\boldsymbol{p}(\boldsymbol{q}) \in \mathbb{R}^m$ is the operational space gravity vector. These terms are given by,

$$\boldsymbol{\Lambda}(\boldsymbol{q}) = (\boldsymbol{J}\boldsymbol{M}^{-1}\boldsymbol{J}^T)^{-1}, \tag{9}$$

$$\boldsymbol{\mu}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \boldsymbol{\Lambda}\boldsymbol{J}\boldsymbol{M}^{-1}\boldsymbol{b} - \boldsymbol{\Lambda}\dot{\boldsymbol{J}}\dot{\boldsymbol{q}}, \tag{10}$$

$$\boldsymbol{p}(\boldsymbol{q}) = \boldsymbol{\Lambda}\boldsymbol{J}\boldsymbol{M}^{-1}\boldsymbol{g}, \tag{11}$$

$$\boldsymbol{N}(\boldsymbol{q})^T = \boldsymbol{1} - \boldsymbol{J}^T\boldsymbol{\Lambda}\boldsymbol{J}\boldsymbol{M}^{-1}. \tag{12}$$

Thus, the overall dynamics of our multibody system, given by,

$$\boldsymbol{M}\ddot{\boldsymbol{q}} + \boldsymbol{b} + \boldsymbol{g} = \boldsymbol{J}^T\boldsymbol{f} + \boldsymbol{N}^T\boldsymbol{\tau}_o = \boldsymbol{\tau}, \tag{13}$$

can be mapped into task space,

$$\boldsymbol{M}\ddot{\boldsymbol{q}} + \boldsymbol{b} + \boldsymbol{g} = \boldsymbol{\tau} \quad \overset{\bar{\boldsymbol{J}}^T}{\rightarrow} \quad \boldsymbol{f} = \boldsymbol{\Lambda}\ddot{\boldsymbol{x}} + \boldsymbol{\mu} + \boldsymbol{p}, \tag{14}$$

where $\bar{\boldsymbol{J}}$ is the dynamically consistent inverse of the task Jacobian,

$$\bar{\boldsymbol{J}} = \boldsymbol{M}^{-1}\boldsymbol{J}^T\boldsymbol{\Lambda}. \tag{15}$$

In a complementary manner the overall dynamics can be mapped into the task consistent null space (or self-motion space) using $\boldsymbol{N}^T$.

The overall dynamics can be expressed as,

$$\boldsymbol{J}^T(\boldsymbol{\Lambda}\ddot{\boldsymbol{x}} + \boldsymbol{\mu} + \boldsymbol{p}) + \boldsymbol{N}^T\boldsymbol{\tau}_o = \boldsymbol{\tau}. \tag{16}$$

A controller employing (8) would be assumed to have imperfect knowledge of the system. Therefore, (8) should reflect estimates for the inertial and gravitational terms. Additionally, a control law needs to be incorporated. To this end we replace $\ddot{\boldsymbol{x}}$ in (8) with the input of the decoupled system, Khatib (1995) $\boldsymbol{f}^\star$, to yield the dynamic compensation equation,

$$\boldsymbol{f} = \widehat{\boldsymbol{\Lambda}}\boldsymbol{f}^\star + \widehat{\boldsymbol{\mu}} + \widehat{\boldsymbol{p}}, \tag{17}$$

where the $\widehat{\cdot}$ represents estimates of the dynamic properties. Thus our control torque is,

$$\boldsymbol{\tau} = \boldsymbol{J}^T(\widehat{\boldsymbol{\Lambda}}\boldsymbol{f}^\star + \widehat{\boldsymbol{\mu}} + \widehat{\boldsymbol{p}}) + \widehat{\boldsymbol{N}}^T\boldsymbol{\tau}_o. \tag{18}$$

Any suitable control law can be chosen to serve as input of the decoupled system. In particular, we can choose a linear proportional-derivative (PD) control law of the form,

$$\boldsymbol{f}^\star = \boldsymbol{K}_p(\boldsymbol{x}_d - \boldsymbol{x}) + \boldsymbol{K}_v(\dot{\boldsymbol{x}}_d - \dot{\boldsymbol{x}}) + \ddot{\boldsymbol{x}}_d, \tag{19}$$

where $\boldsymbol{x}_d$ are reference values for the task coordinates and $\boldsymbol{K}_p$ and $\boldsymbol{K}_v$ are gain matrices.
As we introduced a set of m_C holonomic and scleronomic constraint equations to the configuration space dynamics we can do the same for the task space dynamics. Mapping (5) and (6) into an appropriate task/constraint space yields, De Sapio et al. (2006),

$$\boldsymbol{\Lambda}\ddot{\boldsymbol{x}} + \boldsymbol{\mu} + \boldsymbol{p} - \bar{\boldsymbol{J}}^T\boldsymbol{\Phi}^T(\boldsymbol{\alpha} + \boldsymbol{\rho}) = \bar{\boldsymbol{J}}^T\boldsymbol{\Theta}^T\boldsymbol{\tau}. \tag{20}$$

The term $\boldsymbol{\alpha}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \in \mathbb{R}^{m_C}$ is the vector of centrifugal and Coriolis forces projected at the constraint, and $\boldsymbol{\rho}(\boldsymbol{q}) \in \mathbb{R}^{m_C}$ is the vector of gravity forces projected at the constraint. These terms are given by,

$$\boldsymbol{\alpha}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \boldsymbol{H}\boldsymbol{\Phi}\boldsymbol{M}^{-1}\boldsymbol{b} - \boldsymbol{H}\dot{\boldsymbol{\Phi}}\dot{\boldsymbol{q}}, \tag{21}$$

$$\boldsymbol{\rho}(\boldsymbol{q}) = \boldsymbol{H}\boldsymbol{\Phi}\boldsymbol{M}^{-1}\boldsymbol{g}, \tag{22}$$

where $\boldsymbol{H}(\boldsymbol{q}) \in \mathbb{R}^{m_C \times m_C}$ is the constraint space mass matrix which reflects the system inertia projected at the constraint,

$$\boldsymbol{H}(\boldsymbol{q}) = (\boldsymbol{\Phi}\boldsymbol{M}^{-1}\boldsymbol{\Phi}^T)^{-1}. \tag{23}$$

The constraint null space projection matrix, $\boldsymbol{\Theta}(\boldsymbol{q})^T \in \mathbb{R}^{n \times n}$, is given by,

$$\boldsymbol{\Theta}(\boldsymbol{q})^T = \mathbf{1} - \boldsymbol{\Phi}^T\bar{\boldsymbol{\Phi}}^T, \tag{24}$$

where, $\bar{\boldsymbol{\Phi}}$, is the dynamically consistent inverse of $\boldsymbol{\Phi}$,

$$\bar{\boldsymbol{\Phi}} = \boldsymbol{M}^{-1}\boldsymbol{\Phi}^T\boldsymbol{H}. \tag{25}$$

The control equation can be expressed as,

$$\bar{\boldsymbol{J}}^T\widehat{\boldsymbol{\Theta}}^T\boldsymbol{\tau} = \widehat{\boldsymbol{\Lambda}}\boldsymbol{f}^\star + \widehat{\boldsymbol{\mu}} + \widehat{\boldsymbol{p}} - \bar{\boldsymbol{J}}^T\boldsymbol{\Phi}^T(\widehat{\boldsymbol{\alpha}} + \widehat{\boldsymbol{\rho}}), \tag{26}$$

where the linear control law of (19) can be used.

It is noted that (20) does not expose the constraint forces (Lagrange multipliers). An alternate form of the constrained task space dynamics is, De Sapio & Park (2010); De Sapio (2011),

$$\boldsymbol{\Theta}^T \boldsymbol{J}^T (\boldsymbol{\Lambda}_c \ddot{\boldsymbol{x}} + \boldsymbol{\mu}_c + \boldsymbol{p}_c) + \boldsymbol{\Phi}^T (\boldsymbol{\alpha} + \boldsymbol{\rho}) + \boldsymbol{N}_c^T \boldsymbol{\tau}_o - \boldsymbol{\Phi}^T \boldsymbol{\lambda} = \boldsymbol{\tau}. \tag{27}$$

The term $\boldsymbol{\Lambda}_c(\boldsymbol{q}) \in \mathbb{R}^{m \times m}$ is the task/constraint space mass matrix, $\boldsymbol{\mu}_c(\boldsymbol{q}, \dot{\boldsymbol{q}}) \in \mathbb{R}^m$ is the task/constraint space centrifugal and Coriolis force vector, $\boldsymbol{p}_c(\boldsymbol{q}) \in \mathbb{R}^m$ is the task/constraint space gravity vector, and $\boldsymbol{N}_c(\boldsymbol{q})^T \in \mathbb{R}^{n \times n}$ is the task/constraint null space projection matrix. These terms are given by,

$$\boldsymbol{\Lambda}_c(\boldsymbol{q}) = (\boldsymbol{J}\boldsymbol{M}^{-1}\boldsymbol{\Theta}^T\boldsymbol{J}^T)^{-1}, \tag{28}$$

$$\boldsymbol{\mu}_c(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \boldsymbol{\Lambda}_c \boldsymbol{J}\boldsymbol{M}^{-1}\boldsymbol{\Theta}^T\boldsymbol{b} - \boldsymbol{\Lambda}_c(\dot{\boldsymbol{J}} - \boldsymbol{J}\boldsymbol{M}^{-1}\boldsymbol{\Phi}^T\boldsymbol{H}\boldsymbol{\Phi})\dot{\boldsymbol{q}}, \tag{29}$$

$$\boldsymbol{p}_c(\boldsymbol{q}) = \boldsymbol{\Lambda}_c \boldsymbol{J}\boldsymbol{M}^{-1}\boldsymbol{\Theta}^T\boldsymbol{g}, \tag{30}$$

$$\boldsymbol{N}_c(\boldsymbol{q})^T = \boldsymbol{\Theta}^T(\boldsymbol{1} - \boldsymbol{J}^T\boldsymbol{\Lambda}_c\boldsymbol{J}\boldsymbol{\Theta}\boldsymbol{M}^{-1}). \tag{31}$$

Equation (27) expresses the control torque as a function of the task accelerations, $\ddot{\boldsymbol{x}}$, the kinematic and dynamic properties, and the constraint forces, $\boldsymbol{\lambda}$. The control equation can be expressed as,

$$\boldsymbol{\tau} + \boldsymbol{\Phi}^T\boldsymbol{\lambda} = \widehat{\boldsymbol{\Theta}}^T\boldsymbol{J}^T(\widehat{\boldsymbol{\Lambda}}_c\boldsymbol{f}^\star + \widehat{\boldsymbol{\mu}}_c + \widehat{\boldsymbol{p}}_c) + \boldsymbol{\Phi}^T(\widehat{\boldsymbol{\alpha}} + \widehat{\boldsymbol{\rho}}) + \widehat{\boldsymbol{N}}_c^T\boldsymbol{\tau}_o, \tag{32}$$

where the linear control law of (19) can be used. These equations need to be complemented by a passivity condition on any unactuated joints,

$$\boldsymbol{S}_p\boldsymbol{\tau} = \boldsymbol{0}, \tag{33}$$

where $\boldsymbol{S}_p \in \mathbb{R}^{(n-k)\times n}$ is a selection matrix that identifies the passive (unactuated) joints. We can express (32) as,

$$\boldsymbol{\tau} + \boldsymbol{\Phi}^T\boldsymbol{S}_u^T\boldsymbol{\lambda}_u = \widehat{\boldsymbol{\Theta}}^T\boldsymbol{J}^T(\widehat{\boldsymbol{\Lambda}}_c\boldsymbol{f}^\star + \widehat{\boldsymbol{\mu}}_c + \widehat{\boldsymbol{p}}_c) + \boldsymbol{\Phi}^T(\widehat{\boldsymbol{\alpha}} + \widehat{\boldsymbol{\rho}} - \boldsymbol{S}_c^T\boldsymbol{\lambda}_{c_d}) + \widehat{\boldsymbol{N}}_c^T\boldsymbol{\tau}_o, \tag{34}$$

where $\boldsymbol{S}_c \in \mathbb{R}^{(k-p)\times m_C}$ is a selection matrix used to select the controlled constraint forces and $\boldsymbol{S}_u \in \mathbb{R}^{(n-k)\times m_C}$ is a selection matrix used to select the uncontrolled constraint forces. The terms $\boldsymbol{\lambda}_{c_d}$ and $\boldsymbol{\lambda}_u$ are the vectors of controlled and uncontrolled constraint forces, respectively, selected out of the full vector of constraint forces. The term $\boldsymbol{\lambda}_{c_d}$ is specified as part of the control reference, along with $\boldsymbol{x}_d$, $\dot{\boldsymbol{x}}_d$, and $\ddot{\boldsymbol{x}}_d$. We can solve for the control torque,

$$\boldsymbol{\tau} = (\boldsymbol{1} - \boldsymbol{\Phi}^T\boldsymbol{S}_u^T(\boldsymbol{S}_p\boldsymbol{\Phi}^T\boldsymbol{S}_u^T)^{-1}\boldsymbol{S}_p)\boldsymbol{h}(\boldsymbol{q}, \dot{\boldsymbol{q}}), \tag{35}$$

where,

$$\boldsymbol{h}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \triangleq \widehat{\boldsymbol{\Theta}}^T\boldsymbol{J}^T(\widehat{\boldsymbol{\Lambda}}_c\boldsymbol{f}^\star + \widehat{\boldsymbol{\mu}}_c + \widehat{\boldsymbol{p}}_c) + \boldsymbol{\Phi}^T(\widehat{\boldsymbol{\alpha}} + \widehat{\boldsymbol{\rho}} - \boldsymbol{S}_c^T\boldsymbol{\lambda}_{c_d}) + \widehat{\boldsymbol{N}}_c^T\boldsymbol{\tau}_o. \tag{36}$$

2.3 Task-level control applied to biomechanical Systems

In this section we apply the task-level control formulation, presented in the previous section, to a biomechanical subsystem. This formulation possesses particular efficacy in addressing

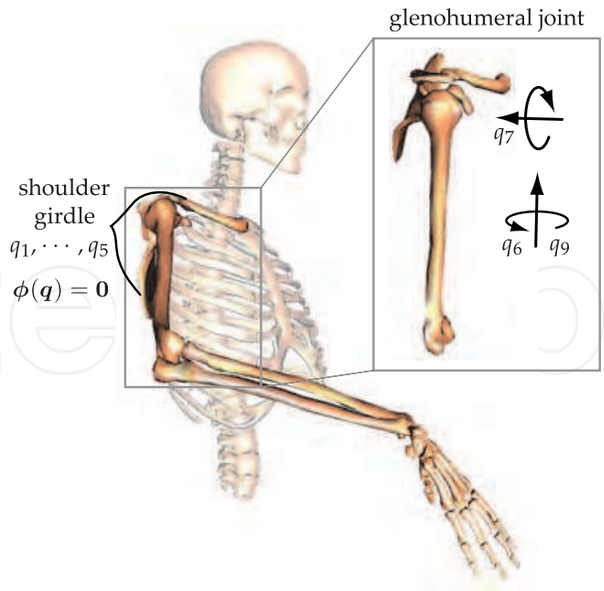


Fig. 6. Reparameterization of the model of Holzbaur et al. Five holonomic constraints couple the movement of the shoulder girdle with the glenohumeral rotations.

complicated systems that involve holonomic constraints. We will choose the human shoulder complex as an illustrative example of this.

Perhaps the most kinematically complicated subsystem in the human skeletal system is the shoulder complex. While the purpose of the shoulder complex is to produce spherical articulation of the humerus, the resultant motion does not exclusively involve motion of the glenohumeral joint. The shoulder girdle, which is comprised of the clavicle and scapula, connects the glenohumeral joint to the torso and produces some of the motion associated with the overall movement of the humerus. While this motion is small compared to the glenohumeral motion its impact on overall arm function is significant, Klopčar & Lenarčič (2001); Lenarčič et al. (2000). This impact is not only associated with the influence of the shoulder girdle on the skeletal kinematics of the shoulder complex, but also its influence on the routing and performance of muscles spanning the shoulder. As a consequence, shoulder kinematics is tightly coupled to the behavior of muscles spanning the shoulder. In turn, the action of these muscles (moments induced about the joints) influences the overall musculoskeletal dynamics of the shoulder. For the aforementioned reasons, when modeling the human shoulder it is important to model the kinematically coupled interactions between the shoulder girdle and the glenohumeral joint.

We can apply a constrained task-level approach to the control of a holonomically constrained shoulder model. This is based on work of De Sapio et al. (2006). The constrained task-level formulation has been updated to the one presented in the previous section. We reparameterized the model of Holzbaur et al. (2005) to include a total of $n = 13$ generalized coordinates (9 for the shoulder complex and 4 for the elbow and wrist) to describe the unconstrained configuration of the arm. As shown in Fig. 6, the coordinates q_6, q_7, and q_9 correspond to the independent coordinates for the shoulder complex used in Holzbaur et al. (2005); elevation plane, elevation angle, and shoulder rotation, respectively.

Five holonomic constraints need to be imposed to properly constrain the motion of the shoulder girdle. With an additional constraint at the glenohumeral joint we have a total of $m_C = 6$ constraints. This yields $p = n - m_C = 7$ degrees of kinematic freedom (3 for the

shoulder complex and 4 for the elbow and wrist). These constraint equations, $\boldsymbol{\phi}(\boldsymbol{q}) = \mathbf{0}$, are given by.

$$\boldsymbol{\phi}(\boldsymbol{q}) = \begin{pmatrix} q_1 - b_1 q_6 - c_1 q_7 \\ q_2 - b_2 q_6 - c_2 q_7 \\ q_3 - b_3 q_6 - c_3 q_7 \\ q_4 - b_4 q_6 - c_4 q_7 \\ q_5 - b_5 q_6 - c_5 q_7 \\ q_8 + q_6 \end{pmatrix} = \mathbf{0}, \tag{37}$$

where the constraint constants, $\boldsymbol{b}$ and $\boldsymbol{c}$, associated with the dependency on humerus elevation plane and elevation angle were obtained from the regression analysis of de Groot and Brand de Groot & Brand (2001).

2.4 Simulated control implementation

Defining a humeral orientation, or pointing, task we have,

$$\boldsymbol{x}(\boldsymbol{q}) = \begin{pmatrix} q_6 & q_7 & q_9 \end{pmatrix}^T. \tag{38}$$

We will not control any of the constraint forces so our control equations consist of,

$$\boldsymbol{\tau} + \boldsymbol{\Phi}^T \boldsymbol{\lambda} = \widehat{\boldsymbol{\Theta}}^T \boldsymbol{J}^T (\widehat{\boldsymbol{\Lambda}}_c \boldsymbol{f}^\star + \widehat{\boldsymbol{\mu}}_c + \widehat{\mathbf{p}}_c) + \boldsymbol{\Phi}^T (\widehat{\boldsymbol{\alpha}} + \widehat{\boldsymbol{\rho}}) + \widehat{\boldsymbol{N}}_c^T \boldsymbol{\tau}_o, \tag{39}$$

$$\boldsymbol{f}^\star = \boldsymbol{K}_p (\boldsymbol{x}_d - \boldsymbol{x}) + \boldsymbol{K}_v (\dot{\boldsymbol{x}}_d - \dot{\boldsymbol{x}}) + \ddot{\boldsymbol{x}}_d, \tag{40}$$

$$\boldsymbol{S}_p \, \boldsymbol{\tau} = \mathbf{0}, \tag{41}$$

where $\boldsymbol{S}_p$ accounts for the unactuated (passive) joints, $q_1, \cdots, q_5$, and q_8,

$$\boldsymbol{S}_p = \begin{pmatrix} 1\,0\,0\,0\,0\,0\,0\,0\,0 \\ 0\,1\,0\,0\,0\,0\,0\,0\,0 \\ 0\,0\,1\,0\,0\,0\,0\,0\,0 \\ 0\,0\,0\,1\,0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,1\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0\,0\,0\,1\,0 \end{pmatrix}. \tag{42}$$

Fig. 7 displays simulation plots for the shoulder complex under a goal position command. The controller was applied to both the constrained shoulder model and a simple model with only glenohumeral articulation (motion of the scapula and clavicle not coupled to glenohumeral motion). The glenohumeral joint control torques associated with the constrained and simple shoulder models, performing identical humeral pointing tasks, differ over their respective time histories. This is particularly true for shoulder elevation angle and elevation plane.

2.5 Muscle-based actuation

In the previous section the simulation of the shoulder complex was actuated with joint torque actuators. In reality biomechanical systems are actuated by a set of musculotendon actuators. Hill-type lumped parameter models for muscle-tendon pairs yield equations of state which

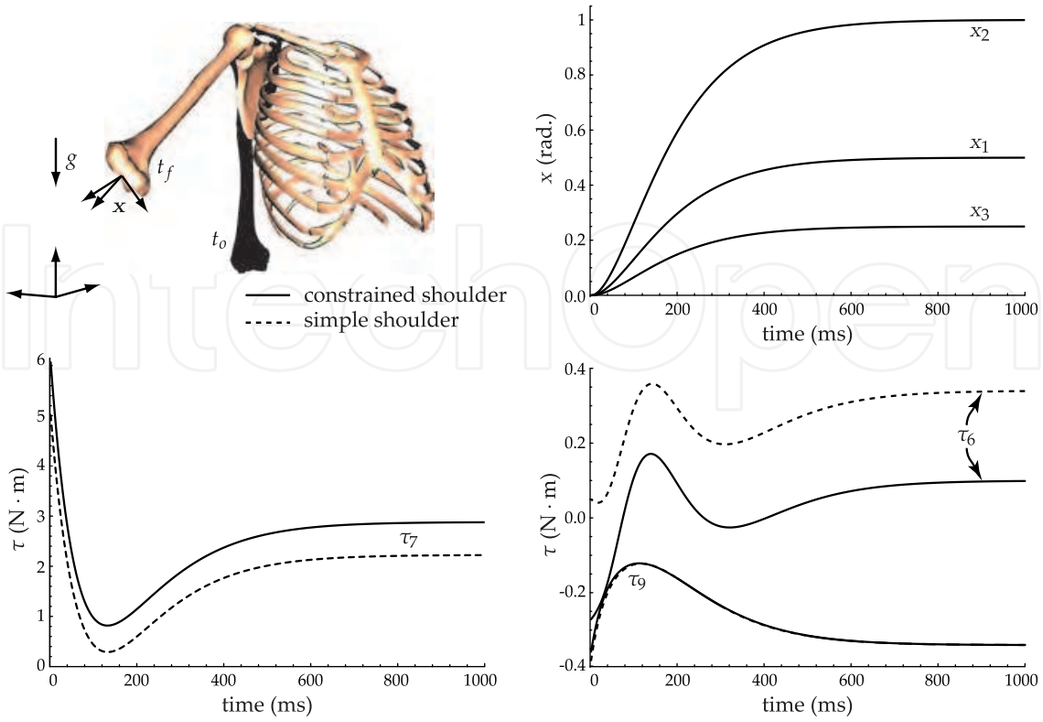


Fig. 7. (Top) Time response of humeral pointing during execution of a goal command for constrained and simple shoulder models. Appropriate dynamic compensation accounts for the control task, $\boldsymbol{x}$, and the shoulder girdle constraints, $\boldsymbol{\phi}$. The control gains are $k_p = 100$ and $k_v = 20$. (Bottom) Glenohumeral joint control torques as predicted by the constrained and simple shoulder models. The inclusion of shoulder girdle constraints influences the resulting torques, particularly for shoulder elevation plane, q_6, and elevation angle, q_7.

describe musculotendon behavior, Zajac (1993). Given a set of r musculotendon actuators we can express the vector of musculotendon forces as $\boldsymbol{f} = \boldsymbol{f}(\boldsymbol{l}, \dot{\boldsymbol{l}}, \boldsymbol{a}) \in \mathbb{R}^r$, where $\boldsymbol{l} \in \mathbb{R}^r$ are the muscle lengths whose behavior is described by a state equation and $\boldsymbol{a} \in \mathbb{R}^r$ are the muscle activations, which reflect the level of motor unit recruitment for a given muscle. Activation is a normalized quantity, that is $a_i \in [0, 1]$. By using either a stiff tendon model or a steady state evaluation of the musculotendon forces we can express $\boldsymbol{f} = \boldsymbol{f}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{a}) = \boldsymbol{F}(\boldsymbol{q}, \dot{\boldsymbol{q}})\boldsymbol{a}$, where $\boldsymbol{F}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \in \mathbb{R}^{r \times r}$ is a diagonal matrix mapping muscle activation, $\boldsymbol{a}$, to muscle force, $\boldsymbol{f}$. The joint moments induced by these musculotendon forces are,

$$\boldsymbol{\tau} = -\boldsymbol{L}(\boldsymbol{q})^T \boldsymbol{f} = -\boldsymbol{L}(\boldsymbol{q})^T \boldsymbol{F}(\boldsymbol{q}, \dot{\boldsymbol{q}})\boldsymbol{a} = \boldsymbol{B}(\boldsymbol{q}, \dot{\boldsymbol{q}})^T \boldsymbol{a}, \tag{43}$$

where $\boldsymbol{L}(\boldsymbol{q}) = \partial \boldsymbol{l} / \partial \boldsymbol{q} \in \mathbb{R}^{r \times n}$ is the musculotendon path Jacobian and $\boldsymbol{B}(\boldsymbol{q}, \dot{\boldsymbol{q}})^T \in \mathbb{R}^{n \times r}$ maps muscle activation, $\boldsymbol{a}$, to joint torque, $\boldsymbol{\tau}$. Equation (5) can thus be expressed in terms of muscle actuation,

$$\boldsymbol{M}\ddot{\boldsymbol{q}} + \boldsymbol{b} + \boldsymbol{g} - \boldsymbol{\Phi}^T \boldsymbol{\lambda} = \boldsymbol{B}^T \boldsymbol{a}. \tag{44}$$

We can then express the control equation as (26),

$$\bar{\boldsymbol{J}}^T \widehat{\boldsymbol{\Theta}}^T \boldsymbol{B}^T \boldsymbol{a} = \widehat{\boldsymbol{\Lambda}} \boldsymbol{f}^\star + \widehat{\boldsymbol{\mu}} + \widehat{\boldsymbol{p}} - \bar{\boldsymbol{J}}^T \boldsymbol{\Phi}^T (\widehat{\boldsymbol{\alpha}} + \widehat{\boldsymbol{\rho}}). \tag{45}$$

Fig. 8. Muscle paths spanning the shoulder complex. Muscle moment arms are determined from the muscle path data Holzbaur et al. (2005). The motion of the shoulder girdle influences the moment arms about the glenohumeral joint.

Due to both kinematic redundancy and actuator redundancy there will typically be many solutions for $\boldsymbol{a}$. Using a static optimization procedure, Thelen et al. (2003), this can be resolved by finding the solution which minimizes $\|\boldsymbol{a}\|^2$ given $a_i \in [0, 1]$. This corresponds to minimizing the instantaneous muscle effort. The use of $\|\boldsymbol{a}\|^2$ and similar cost measures have been suggested in a number of sources, Anderson & Pandy (2001); Crowninshield & Brand (1981).

In Section 2.4 we observed that the constrained shoulder model, which involves kinematic coupling between the humerus, scapula and clavicle, differs from the simple shoulder model with regard to the control torques that are required to achieve a desired motion control task. The constrained model also differs from the simple model in the degree to which the system of muscles are able to generate control forces to achieve a desired motion control task. This is due to the influence of the constrained motion between the humerus, scapula and clavicle on the muscle forces and muscle moment arms about the glenohumeral joint (see Fig. 8).

An example of this is shown in Fig. 9. Predicted muscle moment arms, muscle forces, and moment generating capacities for the deltoid muscles are compared for the simple and constrained shoulder models. The muscle path and force-length data were taken from the study of Holzbaur et al. (2005). In the constrained shoulder model the motions of the scapula and clavicle are highly coupled to humerus elevation angle (q_7 coordinate), whereas, in the simple shoulder model the motion of the scapula and clavicle are not coupled to glenohumeral motion. The paths of the deltoid muscles are affected by the constrained motion of the humerus, scapula, and clavicle. This results in significant differences in moment arms predicted by the two models, with the constrained model often generating moment arms of substantially larger magnitude than the simple model.

Additionally, the predicted isometric muscle forces (computed at full activation) generated by the two models differ. The resulting moment generating capacities of the constrained model are often substantially larger in magnitude than the simple model. This implies that the simple model, which excludes the constrained shoulder girdle motion, typically underestimates the moment generating capacities of muscles that span the shoulder, since Holzbaur et al. (2005) demonstrated correlation between predicted and experimental moment generating capacities

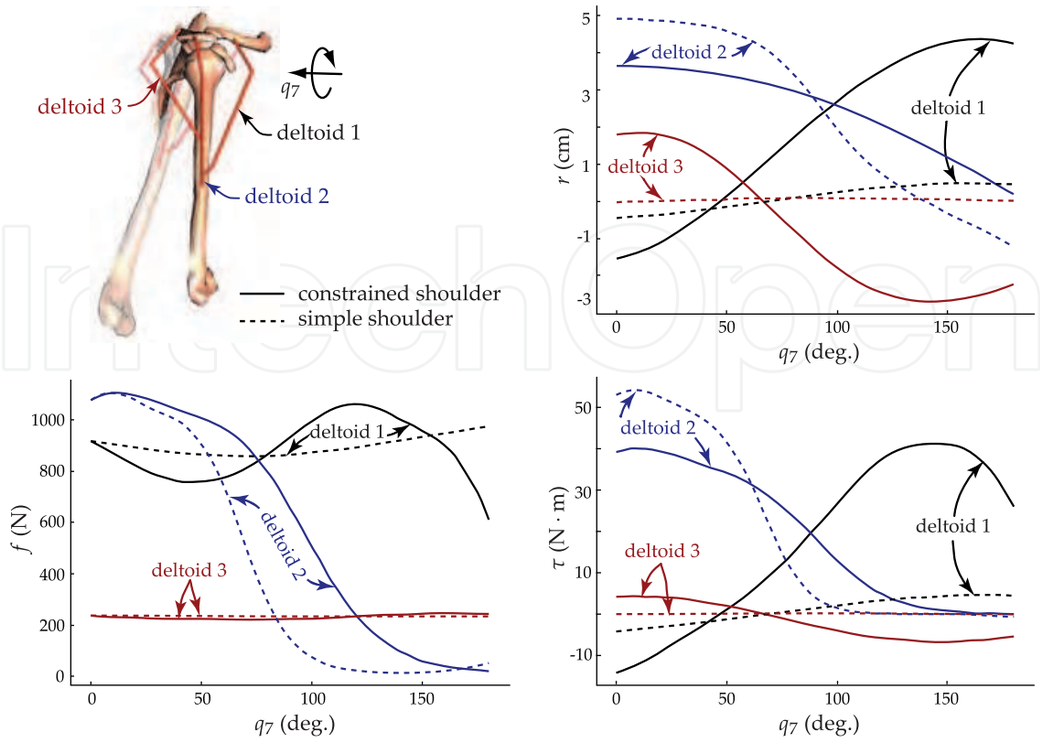


Fig. 9. (Top) Muscle moment arms for the deltoid muscles, as predicted by the constrained and simple shoulder models. The constrained model typically generates moment arms of substantially larger magnitude than those of the simple model. (Bottom) Muscle forces and moment generating capacities for the deltoid muscles. The resulting moment generating capacities associated with the constrained model are typically larger in magnitude than those associated with the simple model.

for the constrained model. This is critical in various applications involving the study and synthesis of human movement, Khatib et al. (2004).

3. Posture-based modeling and analysis of biomechanical systems

In this section we present a muscle effort criterion for the prediction of upper limb postures. In the overall framework this addresses the highlighted element of Fig. 10. The focus is on developing a neuromuscular criterion and a methodology for synthesizing posture in the presence of that criterion.

A particularly relevant class of human movements involves targeted reaching. Given a specific target the prediction of kinematically redundant upper limb motion is a problem of choosing one of a multitude of control solutions, all of which yield kinematically feasible configurations. It has been observed that humans resolve this redundancy problem in a relatively consistent manner, Kang et al. (2005); Lacquaniti & Soechting (1982). For this reason general mathematical models have proven to be valuable tools for motor control prediction across human subjects.

Approaches for predicting human arm movement have been categorized into posture-based and trajectory-based (or transport-based) models, Hermens & Gielen (2004); Vetter et al. (2002). Posture-based models are predicated upon the assumption of Donders' law. Specifically, Donders' law postulates that final arm configuration is dependent only on

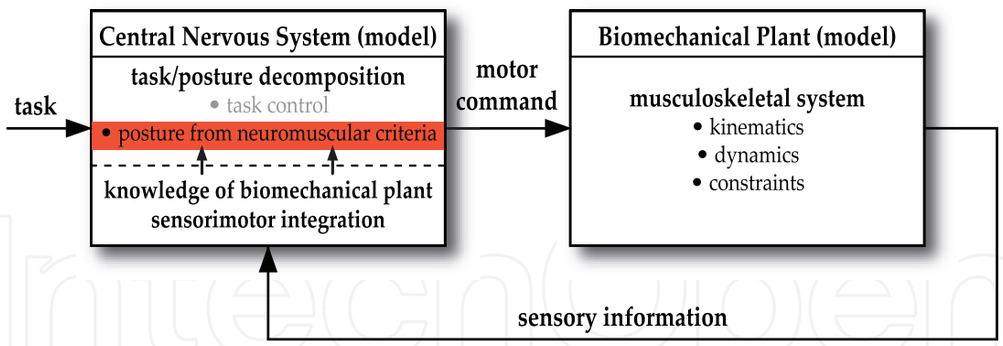


Fig. 10. Task/posture motion control model for biomechanical systems highlighting posture control from neuromuscular criteria.

final hand position and is independent of initial (or past) arm configurations. Thus, the fundamental characteristic of posture-based models is path independence in predicting equilibrium arm postures. In these models the postulated behavior of the central nervous system (CNS) can be said to execute movements based strictly on control variables (e.g., hand position). Conversely, trajectory-based models, which include the minimum work model, Soechting et al. (1995), and the minimum torque-change model, Uno et al. (1989), are characterized by dependence of final arm configuration on the final hand position, the starting configuration, and the choice of a specific optimal path parameterized over time (i.e., past arm configurations).

Many of the models for predicting human arm movement, including the minimum work model and the minimum torque-change model, do not involve any direct inclusion of muscular properties such as routing kinematics and strength properties. Even models described as employing biomechanical variables, Kang et al. (2005), typically employ only variables derivable purely from skeletal kinematics and not musculoskeletal physiology. It is felt that the utilization of a model-based characterization of muscle systems, which accounts for muscle kinematic and strength properties, is critical to authentically simulating human motion since all human motion is predicated upon physiological capabilities.

3.1 Biomechanical effort minimization

We begin with a general consideration of biomechanical effort measures. An instantaneous effort measure can be used in a trajectory-based model of movement by seeking a trajectory, consistent with task constraints, that minimizes the integral of that measure over the time interval of motion. Alternatively, the instantaneous effort measure can be used in a posture-based model by seeking a static posture, consistent with the target constraint, which minimizes the *static* form of the measure.

Proceeding from Section 2.5 we express the joint torques in terms of muscle activations,

$$\boldsymbol{\tau} = -\boldsymbol{L}(\boldsymbol{q})^T \boldsymbol{f} = -\boldsymbol{L}(\boldsymbol{q})^T \boldsymbol{F}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \boldsymbol{a} = \boldsymbol{B}(\boldsymbol{q}, \dot{\boldsymbol{q}})^T \boldsymbol{a}. \tag{46}$$

Due to the fact that there are typically more muscles spanning a set of joints than the number of generalized coordinates used to describe those joints this equation will have an infinite set of solutions for $\boldsymbol{a}$. Choosing the solution, $\boldsymbol{a}_o$, which has the smallest magnitude (least norm) yields,

$$\boldsymbol{a}_o = \boldsymbol{B}^{T+} \boldsymbol{\tau} = \boldsymbol{B} (\boldsymbol{B}^T \boldsymbol{B})^{-1} \boldsymbol{\tau}, \tag{47}$$

where $\boldsymbol{B}^{T+}$ is the pseudoinverse of $\boldsymbol{B}^T$. Our instantaneous muscle effort measure can then be expressed as,

$$U = \|\boldsymbol{a}_o\|^2 = \boldsymbol{\tau}^T(\boldsymbol{B}^T\boldsymbol{B})^{-1}\boldsymbol{\tau}. \tag{48}$$

Expressing this effort measure in constituent terms and dissecting the structure we have,

$$U = \boldsymbol{\tau}^T \overbrace{[\underbrace{\boldsymbol{L}^T}_{\text{kinematics}} \underbrace{(\boldsymbol{F}\boldsymbol{F}^T)}_{\text{kinetics}} \underbrace{\boldsymbol{L}}_{\text{kinematics}}]}^{\text{muscular capacity}}{}^{-1}\boldsymbol{\tau}. \tag{49}$$

This allows us to gain some physical insight into what is being measured. The terms inside the brackets represent a measure of the net capacity of the muscles. This is a combination of the force generating kinetics of the muscles as well as the mechanical advantage of the muscles, as determined by the muscle routing kinematics. The terms outside of the brackets represent the kinetic torque requirements of the task and posture.

It is noted that the solution of (46) expressed in (47) corresponds to a constrained minimization of $\|\boldsymbol{a}\|^2$, however, this solution does not enforce the constraint that muscle activation must be positive (muscles can only produce tensile forces). Imposing inequality constraints, $0 \leq a_i \leq 1$, on the activations requires a quadratic programming (QP) approach for performing the constrained minimization. In this case the solution of (46) which minimizes $\|\boldsymbol{a}\|^2$ and satisfies $0 \leq a_i \leq 1$ can be represented in shorthand as,

$$\boldsymbol{a}_o = \text{QP}(\boldsymbol{B}^T, \boldsymbol{\tau}, \|\boldsymbol{a}\|^2, 0 \leq a_i \leq 1), \tag{50}$$

where QP($\square$) represents the output of a quadratic programming function (e.g., `quadprog()` in the Matlab optimization toolbox). Our muscle effort criterion is then $U = \|\boldsymbol{a}_o\|^2$, where $\boldsymbol{a}_o$ is given by (50). Despite the preferred use of quadratic programming for computational purposes, (49) provides valuable insights at a conceptual level.

3.2 Posture-based criteria

For posture-based analysis the static form of the instantaneous muscle effort measure can be constructed by noting that $\dot{\boldsymbol{q}} \rightarrow \boldsymbol{0}$, thus eliminating the dependency of U on $\dot{\boldsymbol{q}}$. This also implies that $\boldsymbol{\tau} \rightarrow \boldsymbol{g}$. Thus, the static form, $U(\boldsymbol{q})$, of (48) is,

$$U(\boldsymbol{q}) = \boldsymbol{g}(\boldsymbol{q})^T[\boldsymbol{B}(\boldsymbol{q})^T\boldsymbol{B}(\boldsymbol{q})^T]^{-1}\boldsymbol{g}(\boldsymbol{q}). \tag{51}$$

Alternatively, imposing the inequality constraints on the activations we have $U = \|\boldsymbol{a}_o\|^2$ where,

$$\boldsymbol{a}_o = \text{QP}(\boldsymbol{B}(\boldsymbol{q})^T, \boldsymbol{g}(\boldsymbol{q}), \|\boldsymbol{a}\|^2, 0 \leq a_i \leq 1). \tag{52}$$

To find a task consistent static configuration which minimizes $U(\boldsymbol{q})$, we first define the self-motion manifold associated with a fixed task point, $\boldsymbol{x}_o$. This is given by $M(\boldsymbol{x}_o) = \{\boldsymbol{q} \,|\, \boldsymbol{x}(\boldsymbol{q}) = \boldsymbol{x}_o\}$ where $\boldsymbol{x}(\boldsymbol{q})$ is the operational point of the kinematic chain (e.g., the position of the hand). For each $\boldsymbol{q}$ on $M(\boldsymbol{x}_o)$ we can compute $U(\boldsymbol{q}) = \|\boldsymbol{a}_o\|^2$ by solving the quadratic programming problem of (52). The minimal effort task consistent configuration is then the configuration, $\boldsymbol{q}$, for which $U(\boldsymbol{q})$ is minimized on $M(\boldsymbol{x}_o)$. Figure 11 illustrates changes in the predicted posture associated with minimal muscle effort as weight at the hand is varied.

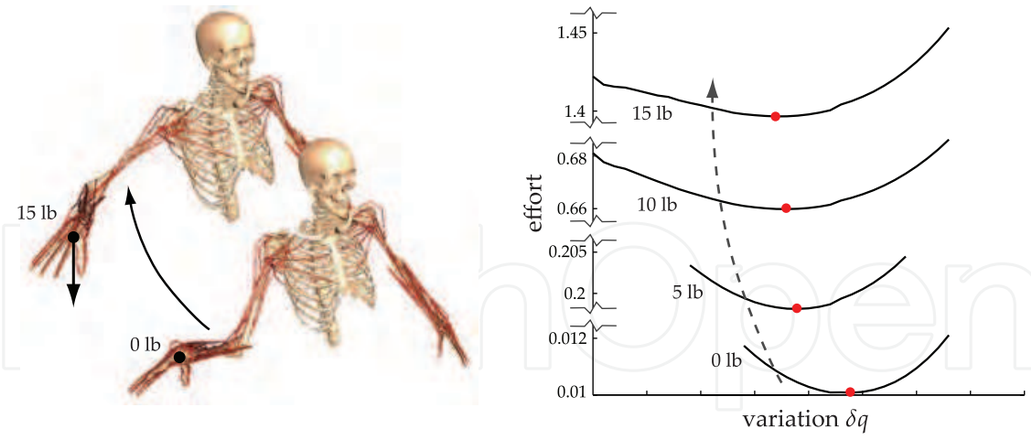


Fig. 11. Muscle effort variation and predicted minimal efforts associated with different weights in hand. The weight at the hand was projected into joint space and added to the gravity vector associated with the limb segments. The effect is that the predicted posture, associated with the minimum of the muscle effort curve, shifts as weight is added. Each point on each of the curves was computed by solving a quadratic programming problem.

3.3 Sphere methods for quadratic programming

Quadratic programming addresses the general minimization of a quadratic function subject to a combination of equality and inequality constraints. It can be formally stated as: Minimize the objective function, $z(\boldsymbol{x})$, with respect to $\boldsymbol{x}$, where,

$$z(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T\boldsymbol{D}\boldsymbol{x} + \boldsymbol{d}^T\boldsymbol{x}, \tag{53}$$

subject to,

$$\boldsymbol{A}\boldsymbol{x} \geq \boldsymbol{b}, \tag{54}$$

$$\boldsymbol{C}\boldsymbol{x} = \boldsymbol{y}. \tag{55}$$

We assume that $\boldsymbol{D}$ is symmetric positive definite and that the polytope defined by $\boldsymbol{A}\boldsymbol{x} \geq \boldsymbol{b}$ is convex. In the case of muscle effort minimization we have the specific form,

$$z(\boldsymbol{a}) = \frac{1}{2}\boldsymbol{a}^T\boldsymbol{a}, \tag{56}$$

subject to,

$$\begin{pmatrix} \mathbf{1}_{r\times r} \\ -\mathbf{1}_{r\times r} \end{pmatrix} \boldsymbol{a} \geq \begin{pmatrix} \mathbf{0}_{r\times 1} \\ -\mathbf{1}_{r\times 1} \end{pmatrix} \tag{57}$$

$$\boldsymbol{B}(\boldsymbol{q})^T\boldsymbol{a} = \boldsymbol{g}(\boldsymbol{q}), \tag{58}$$

where $\mathbf{1}_{r\times r}$ is the $r \times r$ identity matrix, $\mathbf{0}_{r\times 1}$ is a column vector of *zeros*, and $\mathbf{1}_{r\times 1}$ is a column vector of *ones*. Clearly, the quadratic form (56) is positive definite and the polytope (57) is convex. For the procedure of muscle effort minimization this QP problem is repeatedly solved for different values of $\boldsymbol{q}$ on $M(\boldsymbol{x}_o)$, generating the function $U(\boldsymbol{q})$. A line search over $M(\boldsymbol{x}_o)$ then yields $\boldsymbol{q}_o$ where $U(\boldsymbol{q}_o)$ represents the minimum of U on the self-motion manifold.

Since this QP problem needs to be solved repeatedly we would like an efficient method for solving it. There are a number of interior point method (IPM) solvers that addresses QP problems. We have implemented one based on the sphere method approach. This approach was initially developed for linear programming (LP) problems, Murty (2006); Murty (2010b), but has been extended for QP problems, Murty (2010a). Our implementation of the sphere method approach for QP will be described here and is based on the approach of Murty et al.
We begin with the general problem of minimizing (53) subject to (54) and (55). It is noted that the equality constraints, $\boldsymbol{Cx} = \boldsymbol{y}$, can be represented as the inequality constraints.

$$\boldsymbol{Cx} > \boldsymbol{y} - \boldsymbol{\epsilon}, \tag{59}$$

$$\boldsymbol{Cx} < \boldsymbol{y} + \boldsymbol{\epsilon}. \tag{60}$$

where ϵ is a vector of small positive tolerances. Consequently, we consider all constraints, both equality and inequality, as being represented by $\boldsymbol{Ax} \geq \boldsymbol{b}$. These constraints describe a polytope K. A simple check can be made to determine if the unconstrained minimum of the objective function is interior to the polytope. If this is the case then the solution to the QP problem is trivial. Assuming that this is not the case we proceed by noting that the facetal hyperplanes defined by, $\boldsymbol{Ax} = \boldsymbol{b}$, can be represented as,

$$\boldsymbol{v}_i^T \boldsymbol{x} = b_i \qquad \text{for } i = 1, \cdots, m, \tag{61}$$

where $\{\boldsymbol{v}_1, \cdots, \boldsymbol{v}_m\}$ are the inward normals of the facetal planes and,

$$\boldsymbol{A} = \begin{pmatrix} \boldsymbol{v}_1^T \\ \vdots \\ \boldsymbol{v}_m^T \end{pmatrix}. \tag{62}$$

We normalize (61) by dividing both sides by $\|\boldsymbol{v}_i\|$. Thus,

$$\hat{\boldsymbol{v}}_i = \frac{\boldsymbol{v}_i}{\|\boldsymbol{v}_i\|}, \qquad \hat{b}_i = \frac{b_i}{\|\boldsymbol{v}_i\|}, \qquad \hat{\boldsymbol{A}} = \begin{pmatrix} \hat{\boldsymbol{v}}_1^T \\ \vdots \\ \hat{\boldsymbol{v}}_m^T \end{pmatrix}. \tag{63}$$

Following these normalizations we perform centering steps from some initial point, $\boldsymbol{x}_i$, interior to the polytope. Two types of centering steps are performed. One is termed a line search from facetal normals (LSFN), the other is termed a line search from computed profitable directions (LSCPD). First, the touching set, $T(\boldsymbol{x})$, at the current point, $\boldsymbol{x}$ (initially $\boldsymbol{x}_i$) is computed. This is the set of facetal hyperplanes which are touched by the largest hypersphere that can be inscribed in the polytope, centered at the current point, $\boldsymbol{x}$.
For the LSFN step the facetal unit normals, $\{\hat{\boldsymbol{v}}_1, \cdots, \hat{\boldsymbol{v}}_m\}$, are iterated through until one is found, $\hat{\boldsymbol{y}}$, such that,

$$\hat{\boldsymbol{v}}_i^T \hat{\boldsymbol{y}} > 0 \qquad \text{for all } i \in T(\boldsymbol{x}), \tag{64}$$

and such that it reduces the objective function, that is,

$$-[\nabla z(\boldsymbol{x})]^T \hat{\boldsymbol{y}} > 0, \tag{65}$$

where $\nabla z(\boldsymbol{x}) = \boldsymbol{Dx} + \boldsymbol{d}$. Given a profitable direction, $\hat{\boldsymbol{y}}$, that meets these criteria a line search is performed to move along this profitable direction until a point is reached for which

the inscribed sphere at that point is a maximum. A backtracking line search has been implemented for this. The line search is terminated at any point where (65) is not satisfied (no longer descending). This LSFN step is repeated as long as profitable directions meeting the criteria are found.
For the LSCPD step the linear system,

$$\hat{\boldsymbol{v}}_i^T \boldsymbol{y}_1 = 1 \quad \text{and} \quad -[\nabla z(\boldsymbol{x})]^T \boldsymbol{y}_1 = 0 \qquad \text{for all } i \in T(\boldsymbol{x}), \tag{66}$$

is solved for a direction $\boldsymbol{y}_1$ and the linear system,

$$\hat{\boldsymbol{v}}_i^T \boldsymbol{y}_2 = 0 \quad \text{and} \quad -[\nabla z(\boldsymbol{x})]^T \boldsymbol{y}_2 = 1 \qquad \text{for all } i \in T(\boldsymbol{x}), \tag{67}$$

is solved for a direction $\boldsymbol{y}_2$. Backtracking line searches are performed sequentially in both of these unit directions, $\hat{\boldsymbol{y}}_1$ and $\hat{\boldsymbol{y}}_2$, until a point is reached for which the inscribed sphere at that point is a maximum. Again, the line search is terminated at any point where (65) is not satisfied. This LSCPD step is repeated until the incremental reduction in the objective function falls below some tolerance. The final output of the centering steps will be labeled $\boldsymbol{x}_r$.
Following the centering steps, descent steps are performed. For a given iteration, a single descent step is chosen based on the best performance of a set different descent steps, in reducing the objective function. All of these descent steps terminate at the boundary of the polytope. Given a unit descent direction $\hat{\boldsymbol{y}}$ the distance along this direction to the polytope boundary is given by,

$$\delta = \min \left(\frac{\hat{\boldsymbol{v}}_i^T \boldsymbol{x}_r - \hat{b}_i}{\hat{\boldsymbol{v}}_i^T \hat{\boldsymbol{y}}} \right) \quad \text{over } i, \quad \text{such that,} \quad \hat{\boldsymbol{v}}_i^T \hat{\boldsymbol{y}} < 0. \tag{68}$$

These candidate descent directions are as follows:

- D1: Choose $\boldsymbol{y} = -\nabla z(\boldsymbol{x}_r)$. Move from $\boldsymbol{x}_r$ along $\hat{\boldsymbol{y}}$ to the boundary of K.
- D2: Choose $\boldsymbol{y}$ to be the direction defined by the displacement vector between the previous two centering locations, $\boldsymbol{y} = \boldsymbol{x}_r - \boldsymbol{x}_{r-1}$. Move from $\boldsymbol{x}_r$ along $\hat{\boldsymbol{y}}$ to the boundary of K.
- D3: Define directions associated with projecting $-\nabla z(\boldsymbol{x}_r)$ on each of the facetal hyperplanes in the touching set. These directions are given by,

  $$\boldsymbol{y}_i = -(\mathbf{1} - \hat{\boldsymbol{v}}_i \hat{\boldsymbol{v}}_i^T) \nabla z(\boldsymbol{x}_r) \quad \forall i \in T(\boldsymbol{x}_r). \tag{69}$$

  Move from $\boldsymbol{x}_r$ along $\hat{\boldsymbol{y}}_i, \forall i \in T(\boldsymbol{x}_r)$, to the boundary of K. Of these $|T(\boldsymbol{x}_r)|$ descents retain the one that results in the greatest reduction in the objective function.
- D4: Choose $\boldsymbol{y}$ to be the average of the directions from D3. Move from $\boldsymbol{x}_r$ along $\hat{\boldsymbol{y}}$ to the boundary of K. The average of the directions from D3 is given by,

  $$\boldsymbol{y} = \sum_{i \in T(\boldsymbol{x}_r)} \frac{-(\mathbf{1} - \hat{\boldsymbol{v}}_i \hat{\boldsymbol{v}}_i^T) \nabla z(\boldsymbol{x}_r)}{|T(\boldsymbol{x}_r)|}. \tag{70}$$

- D5: Compute the touching point, $\boldsymbol{x}_r^i$ associated with $\boldsymbol{x}_r$. This is the point on each facetal hyperplane in the touching set where the maximum inscribed hypersphere, centered at $\boldsymbol{x}_r$, touches. These points are given by,

  $$\boldsymbol{x}_r^i = \boldsymbol{x}_r + \hat{\boldsymbol{v}}_i (b_i - \hat{\boldsymbol{v}}_i^T \boldsymbol{x}_r) \quad \forall i \in T(\boldsymbol{x}_r). \tag{71}$$

The near touching point is defined as a point on the line segment between $\boldsymbol{x}_r$ and $\boldsymbol{x}_r^i$.

$$\tilde{\boldsymbol{x}}_r^i = \epsilon \boldsymbol{x}_r + (1-\epsilon)\boldsymbol{x}_r^i \quad \forall i \in T(\boldsymbol{x}_r), \tag{72}$$

where *epsilon* is a small tolerance (e.g., ≈ 0.1). Projecting $-\nabla z(\tilde{\boldsymbol{x}}_r^i)$ on each of the facetal hyperplanes in the touching set yields,

$$\boldsymbol{y}_i = -(\mathbf{1} - \hat{\boldsymbol{v}}_i \hat{\boldsymbol{v}}_i^T)\nabla z(\tilde{\boldsymbol{x}}_r^i) \quad \forall i \in T(\boldsymbol{x}_r). \tag{73}$$

Move from $\tilde{\boldsymbol{x}}_r^i$ along $\hat{\boldsymbol{y}}_i, \forall i \in T(\boldsymbol{x}_r)$, to the boundary of K. Of these $|T(\boldsymbol{x}_r)|$ descents retain the one that results in the greatest reduction in the objective function.

The output of D1 through D5 that results in the greatest reduction in the objective function is used to yield the new point $\boldsymbol{x}$. The centering and descent steps are repeated until some solution tolerance is met. In subsequent iterations the feasible set K shrinks based on the objective tangent hyperplane passing thorough $\boldsymbol{x}$. That is, the constraints are appended to include the objective tangent hyperplane passing through the current $\boldsymbol{x}$,

$$\hat{\boldsymbol{A}} = \begin{pmatrix} -[\nabla z(\boldsymbol{x})]^T \\ \hat{\boldsymbol{v}}_1^T \\ \vdots \\ \hat{\boldsymbol{v}}_m^T \end{pmatrix} \quad \text{and} \quad \hat{\boldsymbol{b}} = \begin{pmatrix} -[\nabla z(\boldsymbol{x})]^T \boldsymbol{x} \\ \hat{b}_1 \\ \vdots \\ \hat{b}_m \end{pmatrix}. \tag{74}$$

Fig. 12 illustrates some of the general steps for centering and descent in this algorithm. The algorithm has been implemented in Matlab and in C++ on problems involving thousands of variables and constraints. It performs favorably in terms of accuracy and speed as compared with Matlab's `quadprog()` IPM routine. Quantitative benchmarking is planned for the future.

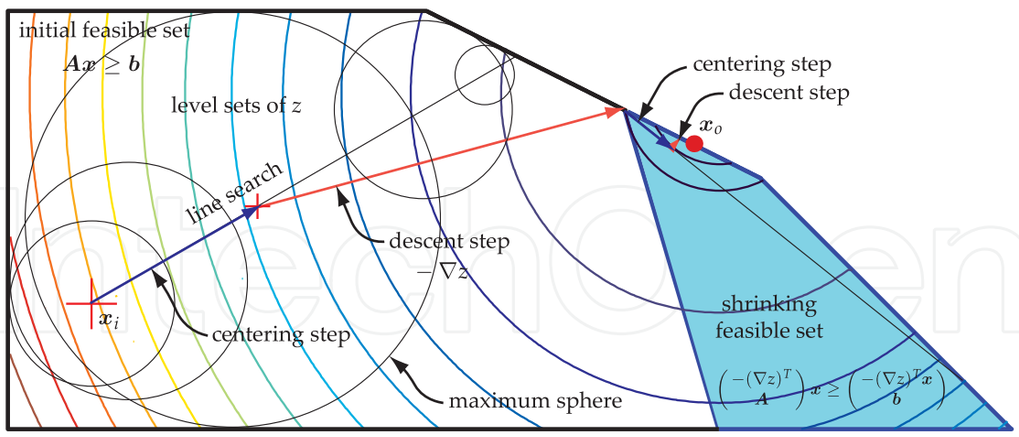


Fig. 12. An illustration of the centering and descent steps associated with the sphere method implemented for QP problems.

3.4 Least action of cost criteria

We now pose the problem of minimizing a cost criterion subject to a motion control task. This is detailed in De Sapio et al. (2008). We can perform this for an instantaneous potential-based

criterion, $U(\boldsymbol{q})$, by using a gradient descent method in conjunction with the task/posture decomposition of (13). Given our overall control torque,

$$\boldsymbol{\tau} = \boldsymbol{J}^T \boldsymbol{f} + \boldsymbol{N}^T \boldsymbol{\tau}_p, \tag{75}$$

the posture term, $\boldsymbol{\tau}_p$, can be chosen to correspond to the gradient descent, $-\partial U/\partial \boldsymbol{q}$, of our cost criterion. In this case the equations of motion are,

$$\boldsymbol{M}\ddot{\boldsymbol{q}} + \boldsymbol{b} + \boldsymbol{g} = \boldsymbol{J}^T \boldsymbol{f} - \boldsymbol{N}^T \frac{\partial U}{\partial \boldsymbol{q}}, \tag{76}$$

subject to the task $\ddot{\boldsymbol{x}}(\boldsymbol{q}) = \ddot{\boldsymbol{x}}_d(t)$. We complement (76) with the task space control law given by (17) and (19).

Gradient descent seeks to reduce an instantaneous criterion rather than extremize a criterion over an integration interval. To address this latter case we define the action integral associated with a cost criterion, as in De Sapio et al. (2008),

$$I \triangleq \int_{t_o}^{t_f} U(\boldsymbol{q}, \dot{\boldsymbol{q}})\, dt. \tag{77}$$

If no task trajectory constraints are specified we have,

$$\begin{gathered} \delta I = 0, \\ \forall\, \delta \,|\, \delta \boldsymbol{q}(t_o) = \delta \boldsymbol{q}(t_f) = \boldsymbol{0}. \end{gathered} \tag{78}$$

Equations (77) and (78) result in the Euler-Lagrange equations,

$$\frac{d}{dt}\frac{\partial U}{\partial \dot{\boldsymbol{q}}} - \frac{\partial U}{\partial \boldsymbol{q}} = \boldsymbol{0}. \tag{79}$$

Imposing rheonomic task trajectory constraints, $\boldsymbol{x}(\boldsymbol{q}) = \boldsymbol{x}_d(t)$, implies,

$$\begin{gathered} \delta J = 0, \\ \forall\, \delta \,|\, \delta \boldsymbol{q}(t_o) = \delta \boldsymbol{q}(t_f) = \boldsymbol{0}, \quad \text{and } \boldsymbol{J}\,\delta \boldsymbol{q} = \boldsymbol{0}, \end{gathered} \tag{80}$$

which yields the system,

$$\frac{d}{dt}\frac{\partial U}{\partial \dot{\boldsymbol{q}}} - \frac{\partial U}{\partial \boldsymbol{q}} = \boldsymbol{J}^T \boldsymbol{\lambda}, \tag{81}$$

or,

$$\boldsymbol{M}_U \ddot{\boldsymbol{q}} + \boldsymbol{b}_U + \boldsymbol{g}_U = \boldsymbol{J}^T \boldsymbol{\lambda}, \tag{82}$$

subject to $\ddot{\boldsymbol{x}}(\boldsymbol{q}) = \ddot{\boldsymbol{x}}_d(t)$. Projecting (82) into task space yields the operational space equations for this system,

$$\boldsymbol{\Lambda}_U(\boldsymbol{q}, \dot{\boldsymbol{q}})\, \ddot{\boldsymbol{x}} + \boldsymbol{\mu}_U(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{p}_U(\boldsymbol{q}) = \boldsymbol{\lambda}, \tag{83}$$

where $\boldsymbol{\Lambda}_U$, $\boldsymbol{\mu}_U$, and $\boldsymbol{p}_U$ are analogous to $\boldsymbol{\Lambda}$, $\boldsymbol{\mu}$, and $\boldsymbol{p}$, but with $\boldsymbol{M}$, $\boldsymbol{b}$, and $\boldsymbol{g}$ replaced by $\boldsymbol{M}_U$, $\boldsymbol{b}_U$, and $\boldsymbol{g}_U$. Applying constraint stabilization, the trajectory constraints can be expressed as,

$$\ddot{\boldsymbol{x}} = \boldsymbol{\lambda}^\star = \ddot{\boldsymbol{x}}_d(t) + \beta[\dot{\boldsymbol{x}}_d(t) - \dot{\boldsymbol{x}}] + \alpha[\boldsymbol{x}_d(t) - \boldsymbol{x}], \tag{84}$$

and the constraint stabilized system is,

$$\boldsymbol{\lambda} = \boldsymbol{\Lambda}_U \boldsymbol{\lambda}^\star + \boldsymbol{\mu}_U + \boldsymbol{p}_U. \tag{85}$$

Two examples from De Sapio et al. (2008) can be used to illustrate the approaches described. First we consider a simplified $n = 3$ degree-of-freedom model of the human arm actuated by $r = 14$ muscles. The system is kinematically redundant with respect to the $m = 2$ degree-of-freedom task of positioning the hand. The muscle attachment and force-length data were taken from the study of Holzbaur et al. (2005). We wish to control the hand to move to a target location, $\boldsymbol{x}_f$, while minimizing an instantaneous muscle effort criterion defined as,

$$U(\boldsymbol{q}) \triangleq \boldsymbol{g}^T (\boldsymbol{B}^T \boldsymbol{B})^{-1} \boldsymbol{g}, \tag{86}$$

where $\boldsymbol{B}(\boldsymbol{q}) = -\boldsymbol{L}(\boldsymbol{q})^T \boldsymbol{F}(\boldsymbol{q})$ and the muscle forces are modeled as $\boldsymbol{f}(\boldsymbol{q}, \boldsymbol{a}) = \boldsymbol{F}(\boldsymbol{q})\boldsymbol{a}$, where,

$$\boldsymbol{F} = \operatorname{diag}\left(f_{o_i} e^{-5\left(\frac{l_i(\boldsymbol{q})}{l_{o_i}} - 1\right)^2} \right). \tag{87}$$

The term, f_{o_i}, represents the maximum isometric force for the ith muscle and l_{o_i} represents the optimal fiber length for the ith muscle. No task trajectory, $\boldsymbol{x}_d(t)$, will be specified, just the final target location, $\boldsymbol{x}_f$.
We have the following control equations,

$$\boldsymbol{f}^\star = k_p(\boldsymbol{x}_f - \boldsymbol{x}) - k_v \dot{\boldsymbol{x}}, \tag{88}$$

$$\boldsymbol{\tau} = \boldsymbol{J}^T \boldsymbol{f}^\star + \widehat{\boldsymbol{g}} - \widehat{\boldsymbol{N}}^T (k_e \frac{\partial U}{\partial \boldsymbol{q}} + k_d \dot{\boldsymbol{q}}). \tag{89}$$

In this case no model of the dynamic properties is included in (**??**). Also, the terms $\ddot{\boldsymbol{x}}_d(t)$ and $\dot{\boldsymbol{x}}_d(t)$ have been omitted in (88) and $\boldsymbol{x}_d(t)$ has been replaced by the final target location, $\boldsymbol{x}_f$, since the goal is to move to a target location without specifying a trajectory. To the posture space portion of (89) we have added a dissipative term, $k_d \dot{\boldsymbol{q}}$, and a gain, k_e, on the gradient descent term. Finally, the gravity vector, $\boldsymbol{g}$, is perfectly compensated for in the overall control. Fig. 13 displays time histories of joint motion, hand motion, and muscle effort for a simulation run. We can see that the controller achieves the final target objective while the null space control simultaneously seeks to reduce the instantaneous muscle effort (consistent with the task requirement). It is recalled that no compensation for the dynamics (except for gravity) was included in (89). Thus, there is no feedback linearization present in the control. Normally, perfect feedback linearization without explicit trajectory tracking would produce straight line motion to the goal. In the absence of feedback linearization non-straight line motion results.
We now seek a trajectory which moves the hand to a target location (see Fig. 14), while extremizing muscle action,

$$I \triangleq \int_{t_o}^{t_f} U(\boldsymbol{q}, \dot{\boldsymbol{q}})\, dt. \tag{90}$$

In this case we will define the instantaneous muscle effort criterion as,

$$U(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \sum_{i=1}^{r} \left(\frac{l_i - l_{o_i}}{l_{o_i}} \right)^2 + \sum_{i=1}^{r} \left(\frac{\dot{l}_i}{v_{o_i}} \right)^2 + \dot{q}_3^2, \tag{91}$$

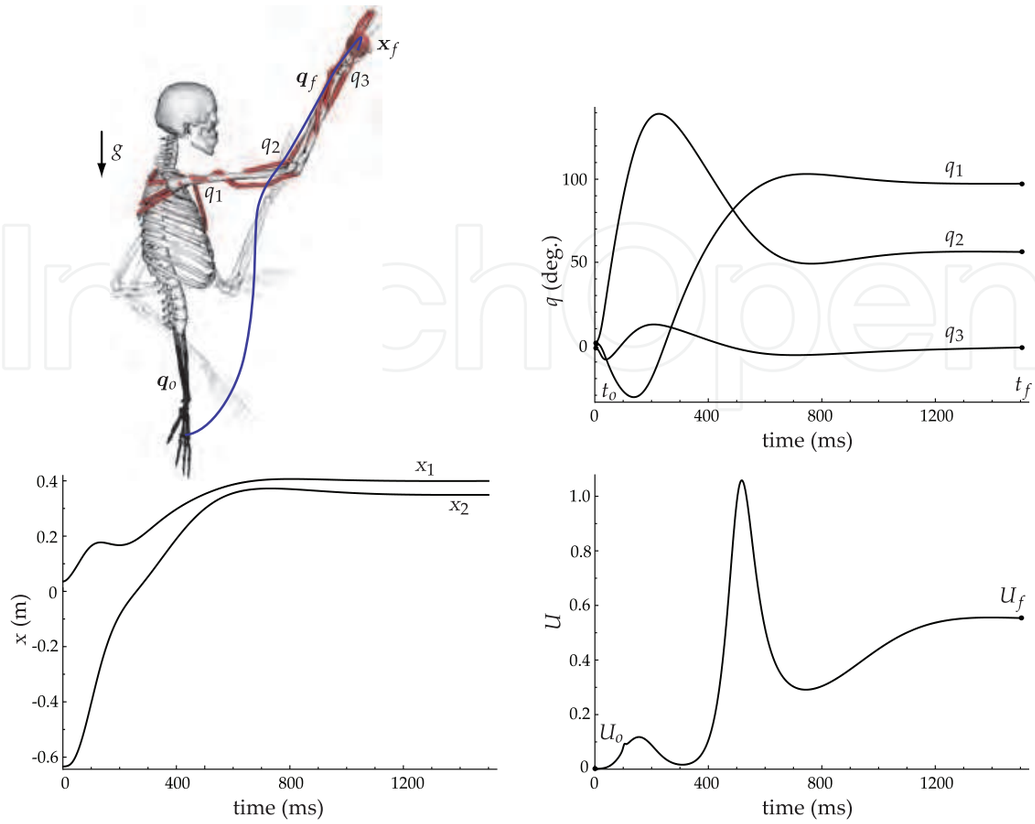


Fig. 13. A redundant muscle-actuated model of the human arm. Initial and final configurations, $\boldsymbol{q}(t_o)$ and $\boldsymbol{q}(t_f)$, associated with gradient descent movement to a target, $\boldsymbol{x}_f$, are shown. (Top) Time history of the arm motion to the target. Motion corresponds to gradient descent of the muscle effort, subject to the task requirement. (Bottom) Time history of hand trajectory and muscle effort criterion, $U(\boldsymbol{q}) = \boldsymbol{g}^T(\boldsymbol{B}^T\boldsymbol{B})^{-1}\boldsymbol{g}$, associated with gradient descent for human arm model. The null space control seeks to reduce the muscle effort but is also constrained by the task requirement.

where l_{o_i} represents the optimal fiber length for the ith muscle and v_{o_i} represents the maximum contraction velocity for the ith muscle.
Under task constraints the system which extremizes the muscle action is given by,

$$\boldsymbol{\lambda}^\star = \alpha(\boldsymbol{x}_f - \boldsymbol{x}) - \beta\dot{\boldsymbol{x}}, \tag{92}$$

$$\boldsymbol{\lambda} = \boldsymbol{\Lambda}_U\boldsymbol{\lambda}^\star + \boldsymbol{\mu}_U + \boldsymbol{p}_U. \tag{93}$$

and (82). The solution yields the muscle action extremizing path between configurations $\boldsymbol{q}(t_o)$ and $\boldsymbol{q}(t_f)$, given the hand target constraint. Fig. 14 displays time histories of joint motion, hand motion, and muscle effort for a simulation run. The straight line motion of the hand results from the feedback linearization employed.

4. Task/posture control for neural prosthetics

If we return to our initial description of the human motor system depicted in Fig. 1 we can add an outer loop associated with the high-level task reasoning and planning functions of

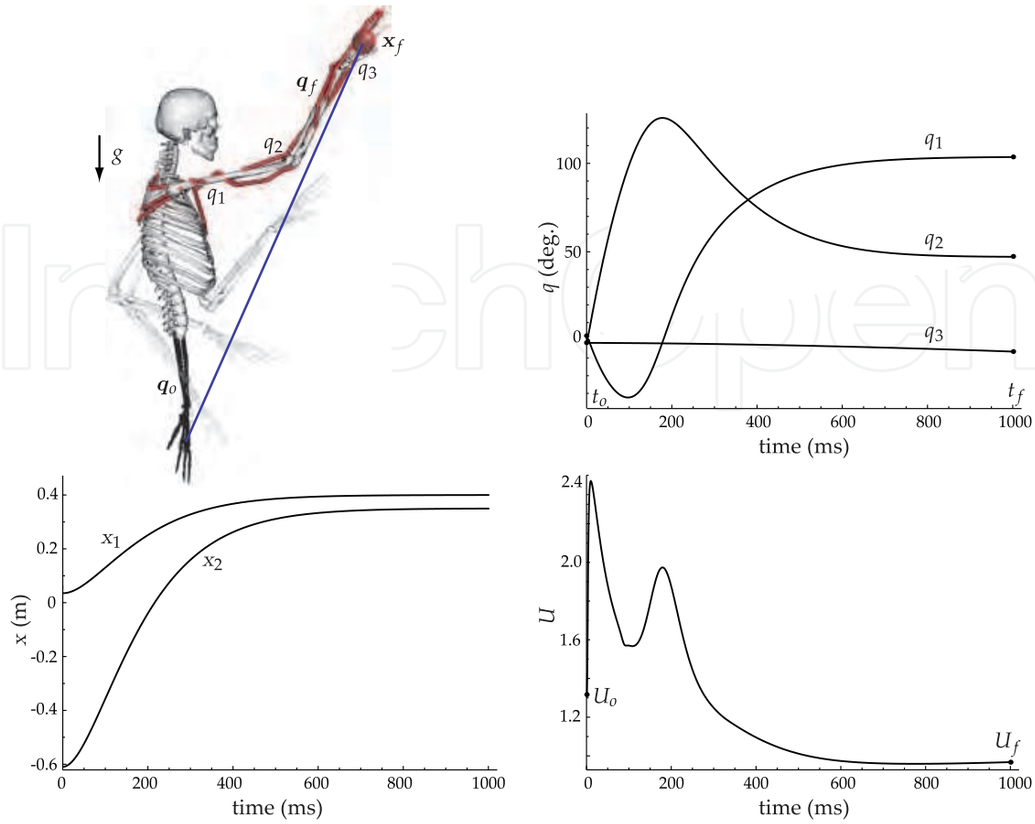


Fig. 14. (Top) Time history of the arm motion between configurations $\boldsymbol{q}(t_o)$ and $\boldsymbol{q}(t_f)$. Motion corresponds to extremization of the muscle effort action integral. (Bottom) Time history of hand trajectory for human arm model and time history of muscle effort criterion associated with extremizing the action integral of (91).

the brain. This is depicted in Fig. 15. In this abstraction motion control is divided into a task generative phase and a motor execution phase. The abstraction depicted in Fig. 15 has relevance not only to the basic understanding of the biomechanics and control of movement but also to the design of engineered systems that augment physiological systems.

Neural prosthetics and brain-computer interfaces have emerged as compelling technologies for the inference of cognitive motor intent using neuroimaging techniques. These techniques can be invasive, as in the case of a brain implant, or non-invasive, as in the case of electroencephalography (EEG). In either case the goal of these techniques is to restore or augment a degree of motor functionality to an individual. This is accomplished through the prediction of motor intent, based on inference from neuroimaging data, and subsequent realization of that intent through a robotic prosthesis. This inference involves decoding the neural encoding manifested in the neuroimaging data. As referenced earlier, current research suggests a task-oriented spatial encoding of motor intent. Based on this premise exciting work has been done to control robotic devices by decoding motor intent.

Current breakthroughs in motor-based brain-computer interfaces can be furthered by the implementation of more sophisticated control theoretic algorithms. Using existing invasive or non-invasive neuroimaging techniques it is believed that the performance of computer controlled robotic devices can be enhanced using a task/posture control framework where, in addition to the inference of task-oriented objectives, postural control objectives can also

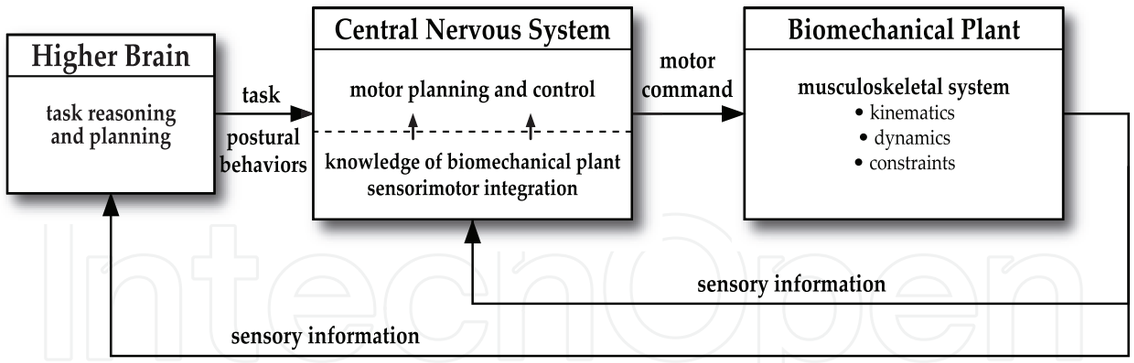


Fig. 15. An outer loop represents the high-level task reasoning and planning functions of the brain. This feeds into the lower-level motor control functions involving the task-driven action of the central nervous system (CNS) on the biomechanical plant.

be inferred from the neuroimgaing data and used as the control reference for the robotic prosthesis. Some of the approaches presented in the previous sections are relevant to the realization of such a neural-based task/posture control framework, as depicted in Fig. 16.

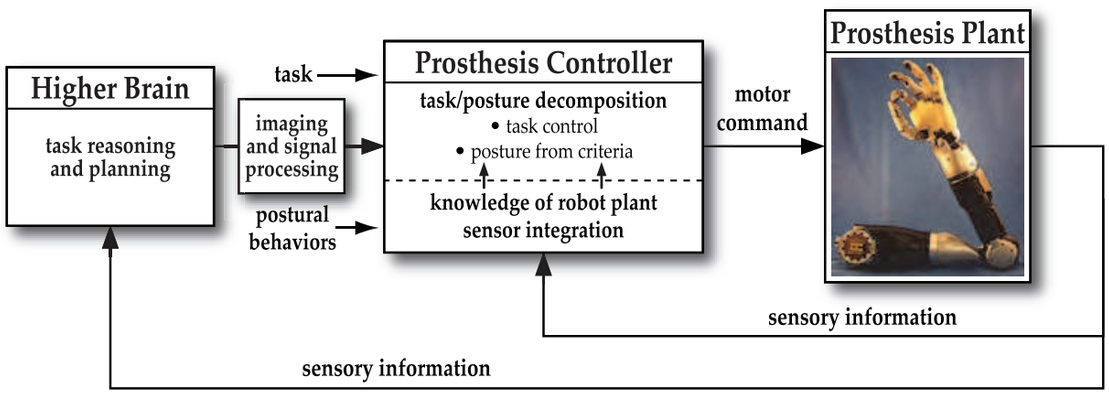


Fig. 16. Task and postural motion intent is inferred from the brain using neuroimaging technologies. The prosthesis controller realizes this intent using a task/posture decomposition. Ultimately, the motor commands are used to control a robotic prosthesis (robot prosthesis image courtesy of DARPA).

Such a framework would involve two principal components: (1) the application of existing signal processing and machine learning methods to the inference of both task-level motor intent as well as postural intent/behavior from neuroimaging data, and (2) control system design and implementation to realize the inferred motor intent on a robotic prosthesis. To complement the neuroimaging data both neuromuscular data in the form of electromyography (EMG) measurements, as well as computational neuromuscular models can be employed in such a framework. This would allow inference and synthesis of control laws based on neuromuscular criteria such as the minimization of neuromuscular effort, etc.

5. Conclusion

A framework has been presented for the analysis and synthesis of human motion through the management of motion tasks, physical constraints, and neuromuscular criteria. The

constituents of this framework include a task-level control methodology for constrained systems as well as a muscle effort criterion for the prediction of postures. The constrained task-level control methodology presented exploits the symmetry between task-level control and constrained dynamics. This approach can be applied to the motion control of systems with persistent holonomic constraints as well as to the motion control of systems which undergo intermittent contact with the environment, as in locomotive biomechanical and robotic systems which make intermittent ground contact.

With regard to posture synthesis a posture-based muscle effort criterion for predicting upper limb motion has been implemented. This criterion characterizes effort expenditure in terms of musculoskeletal parameters, rather than just skeletal parameters as with many previous criteria. As with any posture-based model this one is based upon the assumption of Donders' Law. In other words, the final arm configuration is assumed to be independent of initial or prior arm configurations and is only dependent on hand position (the control variable) and the instantaneous physiological criterion. Good correlation between natural reaching postures and those predicted by the proposed posture-based muscle effort criterion have been shown De Sapio, Warren & Khatib (2006); Khatib et al. (2009). Additionally, an analytical procedure has been outlined for the analysis of trajectory-based effort minimization using gradient descent and least action methods. We have also outlined how our task/posture approach might be employed in neural prosthetics and brain-computer interfaces.

6. References

Anderson, F. C. & Pandy, M. G. (2001). Static and dynamic optimization solutions for gait are practically equivalent, *Journal of Biomechanics* 34(2): 153–161.

Buneo, C. A., Jarvis, M. R., Batista, A. P. & Andersen, R. A. (2002). Direct visuomotor transformations for reaching, *Nature* 416: 632–636.

Crowninshield, R. & Brand, R. (1981). A physiologically based criterion of muscle force prediction in locomotion, *Journal of Biomechanics* 14: 793–801.

de Groot, J. H. & Brand, R. (2001). A three-dimensional regression model of the shoulder rhythm, *IEEE Transactions on Biomedical Engineering* 16: 735–743.

De Sapio, V. (2011). Task-level control of motion and constraint forces in holonomically constrained robotic systems, *Proceedings of the 18th World Congress of the International Federation of Automatic Control*. To appear.

De Sapio, V., Holzbaur, K. & Khatib, O. (2006). The control of kinematically constrained shoulder complexes: Physiological and humanoid examples, *Proceedings of the 2006 IEEE International Conference on Robotics and Automation*, Vol. 1, IEEE, pp. 2952–2959.

De Sapio, V., Khatib, O. & Delp, S. (2006). Task-level approaches for the control of constrained multibody systems, *Multibody System Dynamics* 16(1): 73–102.

De Sapio, V., Khatib, O. & Delp, S. (2008). Least action principles and their application to constrained and task-level problems in robotics and biomechanics, *Multibody System Dynamics* 19(3): 303–322.

De Sapio, V. & Park, J. (2010). Multitask constrained motion control using a mass-weighted orthogonal decomposition, *Journal of Applied Mechanics* 77(4): 041004 (10 pp.).

De Sapio, V., Warren, J. & Khatib, O. (2006). Predicting reaching postures using a kinematically constrained shoulder model, *in* J. Lenarčič & B. Roth (eds), *On Advances in Robot Kinematics*, first edn, Springer, pp. 209–218.

Hermens, F. & Gielen, S. (2004). Posture-based or trajectory-based movement planning: a comparison of direct and indirect pointing movements, *Experimental Brain Research* 159(3): 340–348.

Holzbaur, K. R. S., Murray, W. M. & Delp, S. L. (2005). A model of the upper extremity for simulating musculoskeletal surgery and analyzing neuromuscular control, *Annals of Biomedical Engineering* 33(6): 829–840.

Kang, T., He, J. & Tillery, S. I. H. (2005). Determining natural arm configuration along a reaching trajectory, *Experimental Brain Research* 167(3): 352–361.

Khatib, O. (1987). A unified approach to motion and force control of robot manipulators: The operational space formulation., *International Journal of Robotics Research* 3(1): 43–53.

Khatib, O. (1995). Inertial properties in robotic manipulation: An object level framework, *International Journal of Robotics Research* 14(1): 19–36.

Khatib, O., Demircan, E., De Sapio, V., Sentis, L., Besier, T. & Delp, S. (2009). Robotics-based synthesis of human motion, *Journal of Physiology - Paris* 103(3–5): 211–219.

Khatib, O., Warren, J., De Sapio, V., & Sentis, L. (2004). Human-like motion from physiologically-based potential energies, *in* J. Lenarčič & C. Galletti (eds), *On Advances in Robot Kinematics*, first edn, Kluwer, pp. 149–163.

Klopčar, N. & Lenarčič, J. (2001). Biomechanical considerations on the design of a humanoid shoulder girdle, *Proceedings of the 2001 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, Vol. 1, IEEE, pp. 255–259.

Lacquaniti, F. & Soechting, J. F. (1982). Coordination of arm and wrist motion during a reaching task, *Journal of Neuroscience* 2(4): 399–408.

Lenarčič, J., Stanišić, M. M. & Parenti-Castelli, V. (2000). Kinematic design of a humanoid robotic shoulder complex, *Proceedings of the 2000 IEEE International Conference on Robotics and Automation*, Vol. 1, IEEE, pp. 27–32.

Murty, K. G. (2006). A new practically efficient interior point method for lp, *Algorithmic Operations Research* 1: 3–19.

Murty, K. G. (2010a). Quadratic programming models, *Optimization for Decision Making*, Springer Verlag, pp. 445–476.

Murty, K. G. (2010b). Sphere methods for lp, *Algorithmic Operations Research* 5: 21–33.

Sabes, P. N. (2000). The planning and control of reaching movements, *Current Opinion in Neurobiology* 10: 740–746.

Scholz, J. P. & Schöner, G. (1999). The uncontrolled manifold concept: identifying control variables for a functional task, *Experimental Brain Research* 126: 289–306.

Shenoy, K. V., Meeker, D., Cao, S., Kureshi, S. A., Pesaran, B., Buneo, C. A., Batista, A. P., Mitra, P. P., Burdick, J. W. & Andersen, R. A. (2003). Neural prosthetic control signals from plan activity, *NeuroReport* 14: 591–596.

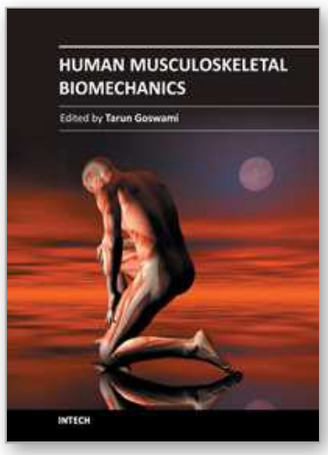
Soechting, J. F., Buneo, C. A., Herrmann, U. & Flanders, M. (1995). Moving effortlessly in three dimensions: Does donders law apply to arm movement?, *Journal of Neuroscience* 15(9): 6271–6280.

Thelen, D. G., Anderson, F. C. & Delp, S. L. (2003). Generating dynamic simulations of movement using computed muscle control, *Journal of Biomechanics* 36: 321–328.

Uno, Y., Kawato, M. & Suzuki, R. (1989). Formation and control of optimal trajectory in human multijoint arm movement, *Biological Cybernetics* 61: 89–101.

Vetter, P., Flash, T. & Wolpert, D. M. (2002). Planning movements in a simple redundant task, *Current Biology* 12(6): 488–491.

Zajac, F. E. (1993). Muscle coordination of movement: a perspective, *Journal of Biomechanics* 26: 109–124.



Human Musculoskeletal Biomechanics

Edited by Dr. Tarun Goswami

ISBN 978-953-307-638-6
Hard cover, 244 pages
Publisher InTech
Published online 05, January, 2012
Published in print edition January, 2012

This book covers many aspects of human musculoskeletal biomechanics. As the title represents, aspects of forces, motion, kinetics, kinematics, deformation, stress, and strain are examined for a range of topics such as human muscles, skeleton, and vascular biomechanics independently or in the presence of devices. Topics range from image processing to interpret range of motion and/or diseases, to subject specific temporomandibular joint, spinal units, braces to control scoliosis, hand functions, spine anthropometric analyses along with finite element analyses. Therefore, this book will be valuable to students at introductory level to researchers at MS and PhD level searching for science of specific muscle/vascular to skeletal biomechanics. This book will be an ideal text to keep for graduate students in biomedical engineering since it is available for free, students may want to make use of this opportunity. Those that are interested to participate in the future edition of this book, on the same topic, as a contributor please feel free to contact the author.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Vincent De Sapio and Richard Chen (2012). A Task-Level Biomechanical Framework for Motion Analysis and Control Synthesis, Human Musculoskeletal Biomechanics, Dr. Tarun Goswami (Ed.), ISBN: 978-953-307-638-6, InTech, Available from: http://www.intechopen.com/books/human-musculoskeletal-biomechanics/a-task-level-biomechanical-framework-for-motion-analysis-and-control-synthesis

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen