

Chapter 1

Better Science Through Predictive Modeling: Numerical Tools for Understanding Neuromechanical Interactions

Nathan E. Bunderson and Jeffrey Bingham

Abstract The animal kingdom is filled with amazing examples of coordinated locomotor and balance behavior. The intricate interaction of the neuromechanics of the combined skeletal, muscular, and neural systems that underlie these behaviors only adds to their impressiveness. To wit, the neuromechanics must deal with fantastically nonlinear dynamics, delayed and noisy sensory input, and multiple stability regimes in unpredictable environments. Because of these underlying complex interactions, an integrative systems approach is required to understand the performance of the locomotor and balance behavior that emerges. In this chapter, we propose the use of predictive modeling to facilitate the investigation of neuromechanics using our software platform, Neuromechanic. With this technique the dynamics of constituent neuromechanical systems are modeled and the resulting emergent behaviors studied; holistic behaviors are an output rather than an input for simulation. We describe three ways in which software can aid in a predictive approach to neuromechanical modeling: first, use of tools that emphasize control and optimization for predictive modeling; second, visualization and organization to aid in careful parameterization necessary to account for the variation found in biological specimens; third, building confidence in modeling results through the use of sensitivity analysis. We offer examples of these techniques using Neuromechanic, which is designed to simplify the prototyping of neural control strategies, formulate optimization criteria, visualize key parameters that effect model performance, and succinctly perform sensitivity analysis.

Keywords Optimization · Control · Neuromechanics · Stability · Simulation · Biomechanics · Sensitivity

N. E. Bunderson (✉)
School of Applied Physiology, Georgia Institute of Technology,
555 14th St NW, Atlanta, GA 30332-0356, USA
e-mail: nbunderson@gatech.edu

J. Bingham
School of Mechanical Engineering, Georgia Institute of Technology,
313 Ferst Drive, Atlanta, GA 30332, USA
e-mail: bingjeff@gmail.com

© Springer Science+Business Media New York 2016
B. I. Prilutsky, D. H. Edwards (eds.), *Neuromechanical Modeling of Posture and Locomotion*, Springer Series in Computational Neuroscience,
DOI 10.1007/978-1-4939-3267-2_1

1.1 The Need for and Value of Predictive Neuromechanical Models of Posture and Locomotion

The pirouette of a dancer, the leap of a receiver catching a football or a toddler clumsily taking her first steps are examples of the sophisticated interaction of the neural and musculoskeletal systems. We call these interactions neuromechanics and they are the bases for movement and balance. Our neuromechanics may evolve as we grow into adulthood, train for a sport, or suffer from injury or disease. Understanding human neuromechanics gives insight into how we are able to achieve grace and efficiency in our movements and offers a way to improve human health. Applying this knowledge to the fields of rehabilitation, robotics, and prosthetics will undoubtedly lead to better fitness training, new methods of injury prevention, improved treatment of neuromusculoskeletal disorders, and better engineered robotic systems.

A tremendous amount of knowledge about the neural, skeletal and muscular systems has come through reductionism and observation (Sherrington 1910; Liddell and Sherrington 1924; Fitts 1954; Huxley 1957; Gordon et al. 1966). However, significantly less advancement has been made in understanding the interplay of these systems during functional behavior. This is in part due to the complexity of each system, but also because the systems are strongly interdependent. Computational modeling is becoming an increasingly powerful tool to analyze these interdependencies and is commonly used to describe the behavior of constituent systems. Models have an advantage over experimentation because they allow complete control over the level of complexity of each system and over how the systems are combined. Also, models can be used to generate a considerably richer set of data for analysis that would likely be prohibitive with physical experimentation. However, *descriptive* modeling is not sufficient to advance science and does not take advantage of the opportunities of a virtual environment.

Compared with descriptive modeling, predictive modeling is less interested in deconstructing a particular behavior than in providing a prediction for *how* the behavior emerges from the interdependent neuromechanical systems. A predictive model can be used to provide constructive arguments, offering an additional logic tool for exploring a particular hypothesis. The constructive nature of predictive modeling often gives important insight about the nature of the problems faced and solved by neuromechanical systems even if the results do not directly explain how they are solved. Properly utilized a predictive model can be used to develop theories and inform the design of specific physical experimentation.

Making predictions about how postural and locomotor behaviors emerge requires greater emphasis on developing neural control theories. Excellent control strategies and optimization techniques have been developed for joint torque-based biomimetic robotic simulations (Brock and Khatib 2002; Jain et al. 2009; Coros et al. 2010; Erez et al. 2013). While these strategies have achieved a remarkable diversity of behaviors in diverse environments and contexts (as do neuromechanical systems) they make little or no attempt to provide an implementation framework for a

neuromechanical system. They are nevertheless important to neuromechanical predictive modeling since they directly address the question of how robust biomimetic behaviors are generated. Others have attempted to generate control strategies which incorporate muscle models and a variety of neural control structures including reflexive mechanisms (Welch and Ting 2008; Geyer and Herr 2010; Bingham et al. 2011; Geijtenbeek et al. 2013) 10/8/2015, central pattern generators (Ijspeert 2008; Markin et al. 2010), basal ganglia (Tomita and Yano 2007; Sarvestani et al. 2013), and the cerebellum (Jo and Massaquoi 2004, 2007). While these are generally not able to achieve the robust behavior of the more abstract biomimetic robotics simulations and are usually implemented with dramatically simplified mechanics they do generate emergent behaviors based on underlying neuromechanical principles.

1.2 What Tools Facilitate Predictive Neuromechanical Modeling?

Neuromechanical models have been implemented using a variety of software tools including Matlab (Mathworks, Natick Ma), OpenSim (Delp et al. 2007), MuJoCo (Todorov et al. 2012), AnimatLab (Cofer et al. 2010), DART (Bai et al. 2012), MSMS (Davoodi and Loeb 2012), and Neuromechanic (Bunderson et al. 2012). The goal of this paper is not to discuss the relative merits of each of these packages. Our purpose is to describe how Neuromechanic is being developed specifically to facilitate predictive neuromechanical modeling. There are three primary considerations in predictive neuromechanical modeling that we are incorporating into the design of Neuromechanic.

First, as discussed previously, predictive neuromechanical modeling requires an emphasis on control and optimization. Neuromechanic is designed to simplify the prototyping of neural control strategies and optimization. In the first section we will describe how an integrated control-oriented approach, which allows for emergent rather than pre-determined behavior, has greater potential for taking neuromechanical modeling from description to prediction.

Second, model parameterization for neuromechanical systems involves greater care and effort than for engineered systems due to the natural variance of parameters in biological systems and the difficulty of accurately measuring many of these parameters. Neuromechanic has visualization tools for several key parameters which we have determined have the greatest effect on model performance. In the second section we describe the parameter visualization tools we are implementing in Neuromechanic that allow the modeler to quickly compare, and preliminarily validate, neuromechanical models.

The third aspect of the predictive approach is the ability to assess the reliability and robustness of modeling results by performing sensitivity analyses. Neuromechanical systems are extremely variable and the behavior is amazingly robust. Any neuromechanical control paradigm that is highly sensitive to parameter variation should be viewed with a great deal of suspicion. Further, conclusions based on the

results of a single model, not to mention a single simulation, do not characterize robustness and are missing the great opportunity mathematical modeling provides to explore the parameter space. In the third section we discuss some sensitivity analysis tools which have been implemented in Neuromechanic.

1.3 Developing Control Strategies in Neuromechanic

In the descriptive framework, the model is generally used to answer “What ...” questions, as in, “What is the force contribution of muscle X to the vertical ground reaction force during vertical leaping?”. These types of questions can and often do generate predictions but they rely heavily on predetermined behaviors. On the other hand in the predictive approach behaviors are an output rather than input and research questions are framed as “Can ...” or “How ...”, as in, “How high can my model jump?”. Where the descriptive questions tend to be reductive, predictive questions tend to be holistic. There is no doubt that asking the reductive questions has been and will continue to be a very productive line of questioning with neuromechanical models. However, there is a strong case to be made for asking holistic questions of neuromechanical models that predict instead of describe behavior. Furthermore, even if a model is being used to answer reductive questions it is still useful and informative to know how it performs holistically (How high can it jump? How fast can it run? How stable is it against external perturbations?) before asking the important detailed reductive questions. The implication for a neuromechanical modeling package is that tools that assist in answering the holistic questions can be extremely useful for predictive modeling.

So what are the tools we need to be able to answer holistic questions such as “How high can a model jump?” Since the predictive approach requires the generation of new behaviors it requires an emphasis on optimization tools and control strategies to generate those behaviors. In the example of maximal vertical leaping we need a control strategy that produces leaping and a process to optimize the parameters of that control strategy to obtain maximum performance. This process may be enhanced by kinematic and kinetic data from experimentally measured behaviors but does not rely on it. We present one example of this approach in another section. Neuromechanic has several tools and control-oriented design features to facilitate predictive modeling.

1.3.1 *Python Application Programmer’s Interface and Tree-Like Data Structure*

With its focus on control and optimization, predictive simulation requires that the neuromechanical modeling package have a scripting environment that allows the modeler to quickly implement and test control and optimization strategies.

The scripting environment should be extensible, have a mathematical toolset, and give the modeler access to all the relevant parameters and dynamic variables of the model. Neuromechanic comes with Python as the scripting environment and includes the scientific computing package numpy. All parameters of the model are available from the Python scripting environment through a model tree that mirrors the input format for intuitive access (Fig. 1.1).

The scripting tools were used to construct a virtual prosthesis environment and prototype a new multi-functional prosthetic controller (Bunderson 2014). Able-bodied subjects and a shoulder disarticulation subject successfully used the system to complete a virtual box and blocks task. An early experimental setup, virtual environment, and the control structure for the virtual prosthesis are shown in Fig. 1.2. A command signal, generated from real-time processing of electromyograms (EMG)

Input file in XML format (e.g. 2392.nmcb)

```
<NeuromechanicFile>
...
<Bodies>
  <RigidBody Name="pelvis">
    <FrameLocation Parent="ground">.0 .0 .0</FrameLocation>
    <Mass>11.7</Mass>
    <Inertia>.1028 .0871 .0579 0. 0. 0.</Inertia>
    <CenterOfMass>-.0707 .0 .0</CenterOfMass>
  </RigidBody>
  <RigidBody Name="left_femur">
    ...
  </RigidBody>
...
</Bodies>
</NeuromechanicFile>
```

Accessing data with python API:

```
>>>import nmextension as nme
>>>nmtree = nme.Tree('model')
>>>print(nmtree.NeuromechanicFile.Bodies.RigidBody['Name=pelvis'].Mass.x_)
11.7
```

Fig. 1.1 Example input file and access commands to model parameters through the Python Application Programmer's Interface (API). Model parameters and dynamic variables can be accessed from the API with a tree class that matches the input XML file format

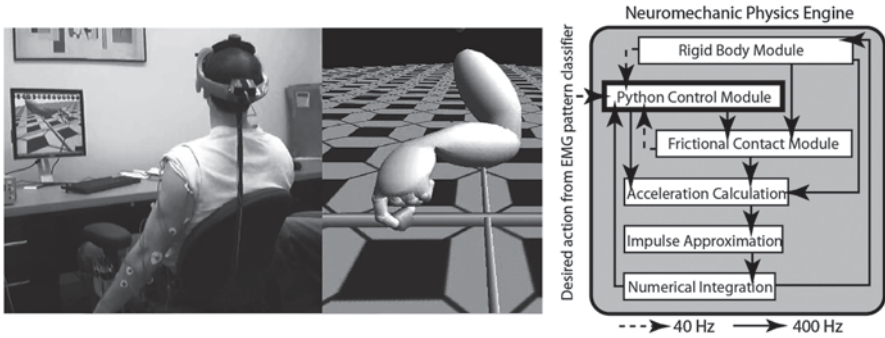


Fig. 1.2 A virtual prosthesis was developed in Neuromechanic and four able-bodied subjects and one shoulder disarticulation subject were able to complete a virtual box and blocks task using real-time EMG control signals

in a separate process, was retrieved in Neuromechanic through shared memory resources. The controller converts the command signal into the joint torques necessary to drive the virtual prosthesis in Neuromechanic in real time. The joint torque calculation relies on dynamic variables (such as frictional contact forces and rigid body dynamics) accessed from the model tree. The virtual prosthesis and box and blocks task implemented in Neuromechanic are extensible and modifiable and can serve as a platform and baseline to quickly and cheaply prototype prosthetic arms and control strategies.

1.3.2 Linearization and Eigenanalysis Tools

Stability is an essential feature of control and Neuromechanic has built in tools for assessing the local stability of a neuromechanical system. A model can be linearized about a state and neuromuscular parameters calculated to equilibrate dynamic variables in that state. In a previous study (Bunderson et al. 2008), we found that the classical muscle redundancy problem was reduced when a stability criterion (in addition to the endpoint force criteria) was used to choose patterns of muscle activation. We used the linearization and Eigenanalysis tools in Neuromechanic and found that the instabilities most likely to occur in the cat hindlimb are in the sagittal plane (Fig. 1.3). By comparing the local (joint-level) stiffness of muscles to the stability of the limb modes we were able to identify destabilizing muscles and construct an optimization cost function to enhance limb stability. This optimization cost function can be used in constructing controllers for locomotion and posture to predict behavior.



Fig. 1.3 Four dominant modes were found from the hindlimb stability analysis ranging from most unstable (*left*) to most stable (*right*)

1.4 Visualization Tools for Developing Model Parameter Intuition

The parameterization of neuromechanical models poses several difficulties. To begin with, the higher the fidelity at which the model represents the neuromechanical system the more parameters are introduced and the more difficult it is for the modeler to track parameter values. Moreover, because of the inherent variability in biological systems there may not be a single “correct value” for a particular parameter. In Neuromechanic we are developing tools to more completely visualize and “gain a feel for” the parameterization of the animal.

Most neuromechanical modeling packages have visualization tools for a limited set of the parameters of the system. For example most packages provide a visual representation of the global location of important points, vectors, and surfaces such as joint axes, muscle attachment points, and muscle wrapping surfaces. Most packages also render the muscles in colors that reflect the activation level of the muscle which can give a fast and intuitive impression of muscle coordination. Often polygon meshes representing bone surfaces or skin are also included (however, these usually do not represent any of the actual parameters that describe the model and are included for reference only). Neuromechanic also provides these visualizations but has additional tools to give the modeler an intuitive feel for other parameters which, in our experience, have the greatest impact on simulations. This includes the parameters directly impacting (1) the force generating capacity of the muscles, (2) the moment arm of the muscles and (3) the inertia of the body segments. To demonstrate these tools we have included figures generated from a Neuromechanic implementation of the 2392 human lower limb model that is included in the OpenSim (Delp et al. 2007) distribution. The name for the model derives from the fact that it models 23 kinematic degrees of freedom and 92 muscles from the lower body (legs and lumbar region). The head, arms, and trunk (HAT) are modeled as a single rigid body attached to the pelvis with a spherical (3 rotational degrees of freedom) joint.

1.4.1 Visualizing the Force Generating Capacity of Muscle

Hill-type muscle models, where the force produced by a muscle is a function of its length, velocity and activation and is scaled by a maximum isometric force (F_{max}), are commonly used in neuromechanical modeling. Zajac expanded a basic Hill-type model to incorporate first order activation dynamics and the effects of pennation (Zajac 1989). His model is the basis for the two primary models incorporated in Neuromechanic. In both of these models the maximum isometric force that can be generated by a muscle is given by $F_{max} a_{max} \theta$ where a_{max} represents the maximum activation level of the muscle (almost always 1) and θ represents the pennation angle. Since any neural control strategy depends critically on the ability of muscles to generate force, Neuromechanic provides a visual representation of this maximum force. Muscles are rendered in a fusiform shape (Fig. 1.4) with a maximum cross-sectional area (A_{max}) at the midpoint of the muscle lengthwise which is defined by

$$A_{max} = \frac{F_{max} a_{max} \cos(\theta)}{\sigma}$$

where the proportionality constant σ is a global parameter (i.e. the same value for all muscles in Neuromechanic) which represents the specific tension of muscle. This can be set by the user and is 22.5N/cm² by default.

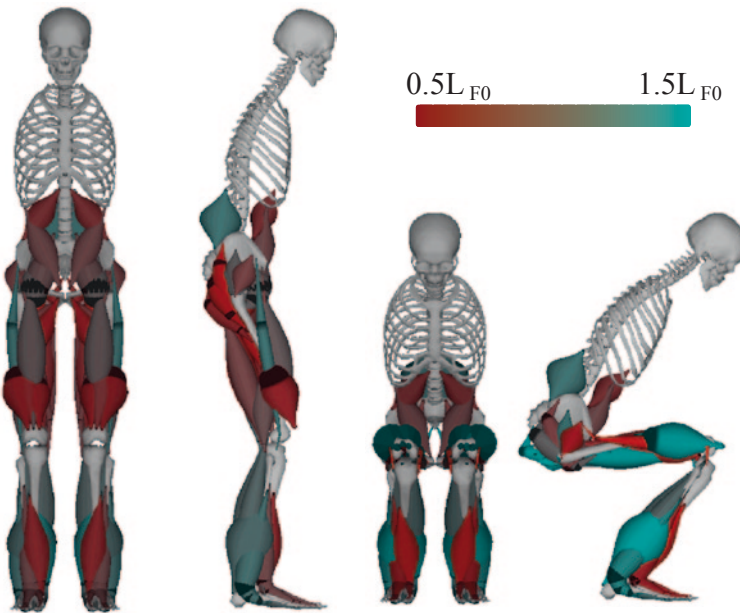


Fig. 1.4 The force generating capacity of muscle can be visualized in Neuromechanic by rendering the muscles in a fusiform shape. The maximum cross-sectional area is proportional to the maximum force that can be produced by the muscle. The color of the muscle can be chosen as a function of the operating length upon which the force generated by muscle depends critically

The force generated by a muscle also depends strongly upon the operating length of a muscle due primarily to variations in actin-myosin overlap of the sarcomeres. The operating length of muscle is not a parameter but a variable that depends on several other time dependent variables (the configuration or joint positions of the animal and fiber lengths), several parameters (the location of the muscle attachment and via points, muscle wrapping surfaces, the number of sarcomeres, pennation angle, and tendon slack lengths), and the particular muscle model in use. In Neuro-mechanic muscles can be rendered with colors that reflect the current normalized operating length of the muscles giving the modeler immediate insight into whether the muscles will be able to generate appropriate levels of force. In Fig. 1.4 the color of the muscles indicate their respective normalized operating lengths in an inelastic tendon model. The lengths vary from 50% shorter (red) to 50% longer (cyan) than the optimal operating length (L_{FO}). At the nominal upright posture on the left of Fig. 1.4 Vastus lateralis of the 2392 model operates at 52% of optimal fiber length while at the squatting position shown on the right it is operating at 132% of optimal fiber length.

1.4.2 Visualizing the Moment Arms of Muscle

Muscle moment arms also strongly impact the performance of neuromechanical models since muscle forces affect the dynamics of the body only after being scaled by the moment arms of the muscles at each joint. Like operating length, moment arms are state dependent variables rather than parameters. They depend on the muscle attachment point parameters, muscle wrapping surface parameters, and joint definition parameters. In particular when attachment points or wrapping surfaces are in close proximity to joint axes small changes in their location can have a large impact on the moment arm. We have implemented a moment arm visualization tool which scales and renders the moment arms at the joints as points along the joint axis. The moment arm of the Rectus femoris muscle in a standing and squatting posture is shown in Fig. 1.5a. The cyan line goes through the hip flexion/extension axis and the extent of the line to the right and left correspond to a 10x scaling of the maximum flexion and extension moment arm of *all* muscles at that joint. The cyan circle represents a 10x scaling of the Rectus femoris moment arm at that joint. The tool indicates that while Rectus femoris is a primary flexor of the hip at the upright posture its moment arm is relatively decreased in the squatting posture. The red line and circle reflect the scaled moment arm of Rectus femoris at the knee flexion/extension joint. Other packages such as OpenSim have excellent moment arm plotting tools which we are also incorporating into Neuromechanic to complement this visualization tool.

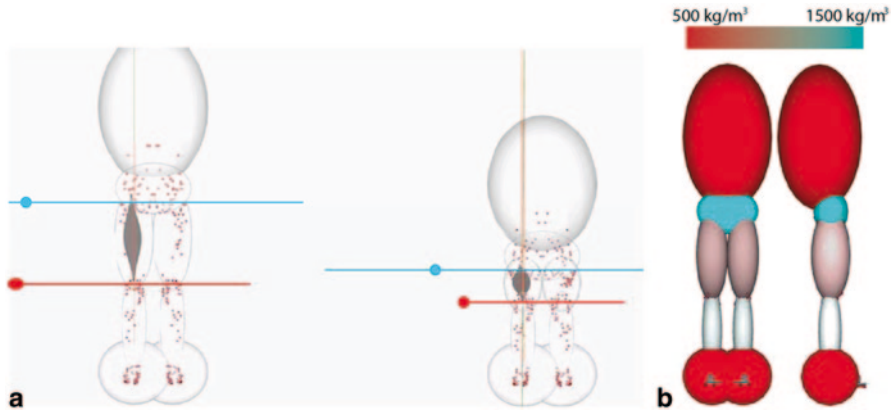


Fig. 1.5 **a** The moment arms of the Rectus femoris muscle of the 2392 OpenSim model (Delp et al. 2007) are represented visually as points along the joint axes. The extent of the cyan line represents the maximum moment arms from all muscles for hip flexion and extension in the shown posture. The cyan dot represents the moment arm of the selected muscle (Rectus femoris) in that posture. The red is for knee flexion/extension. **b** The seven inertial parameters of the rigid body segments in the 2392 OpenSim model are represented in Neuromechanic with shaded ellipsoids

1.4.3 Visualizing the Inertia of Body Segments

The dynamics of movement also depend fundamentally on the inertia of the body. Neuromechanic, like most neuromechanical modeling packages uses rigid body dynamics where the inertia of each segment is completely described by seven parameters including mass (translational inertia) and six components of rotational inertia. These seven parameters can be uniquely transformed into an equivalent inertial ellipsoid representation which can be easily visualized. In the equivalent inertial ellipsoid representation the seven parameters can be visualized as the size (3 parameters), orientation (3 parameters), and density (1 parameter) of an ellipsoid. If the average density of the animal is known then the modeler can quickly get a feel for the appropriateness of the inertial parameters by rendering the ellipsoid with color corresponding to density (Fig. 1.5b). In Fig. 1.5 the bones and most muscles of the 2392 model are not rendered so as not to obfuscate the information conveyed by the rendered inertial ellipsoids. Most segments of the 2392 model are between 50% below and 50% above the average density of human body (1062 kg/m^3) with the HAT (head, arms, trunk) and pelvis lying at the extremes of this range. Two segments, the talus and toes, saturate the scale with densities of less than 10 kg/m^3 for the talus (large spherical red ellipsoid near the ankle) and 6700 kg/m^3 for the toes (small very thin cyan ellipsoid).

We emphasize that none of these tools are meant to give final validation to the choice of parameters. We believe however that they are very useful for a quick appraisal of general appropriateness of the parameters that will have the greatest effect on performance of the model. In the case of the 2392 model this visualization suggests that some inertial parameters need revisiting.

1.5 Strategies and Tools for Determining Model Sensitivity

Ultimately the modeler must assess the reliability of model results. Modeling results are used to answer questions like: Is it safe to use a particular physical design of a prosthesis or orthosis? Is it safe to promote a gait modification to alleviate joint stress? Is it safe to perform a tendon transfer surgery? Due to the variety of anatomical structure and variation in observed behavior it is necessary to quantify how model parameters affect simulated behavior to gain confidence in model predictions. In short, the model prediction is only as good as the model itself, and the model is dependent on the accuracy of its parameters and the verisimilitude of its dynamics. Therefore, to vet a particular model often requires determining how accurate a parameter must be measured or predicted to achieve a reasonable variation in simulated behavior. Furthermore, it is also necessary to discern whether variations in observed behavior are the result of differences in parameters or actual differences in the behavior itself. Quantifying the sensitivity of behavior to changes in parameters also helps to identify parameters that are most important for a particular behavior and may give insight into the underlying neuromechanical interactions.

Quantifying sensitivity of neuromechanical systems is challenging due to their non-linear nature and multitude of parameters. However, using dynamical systems theory the stability of the neuromechanical system for a particular behavior and parameter set can be used to identify sensitivity of behavior to parameter changes. Therefore, Neuromechanic implements methods to determine equilibrium behavior, linearization of the full neuromechanical system and eigenvalue analysis. These tools have been discussed here in a previous section.

In Neuromechanic we have also implemented the stability radius technique as a tool for identifying different sets of parameter values that produce similar behavior, or quantifying the sensitivity of a behavior to parameter changes. Briefly, stability radius is a technique from robust control theory that produces a single number that measures the smallest change to any system parameter that would result in unstable behavior. This technique has advantages over other methods in that it is possible to compare dynamical behavior as parameters change for one system against itself, as well as against entirely different systems.

As an example, it is possible to use stability radius to compare the stability of different equilibrium postures during a station-keeping task. Here we use a model of a cat hind-limb having seven degrees-of-freedom and 31 Hill-type muscles (Burkholder and Nichols 2004) starting in a nominal configuration. The pelvis was held fixed and the leg was allowed to hang against gravity and minimum muscle activation was solved to hold the limb in the desired configuration. The hip angle was varied -30 to $+10^\circ$ from the nominal posture.

Stability radius was computed for each posture by linearizing this system about the equilibrium state, \tilde{x} , to form the linearized system with a system matrix, \mathbf{A} :

$$\dot{\tilde{x}} = \mathbf{A}\tilde{x}$$

The resulting characteristic equation of the system can be written as:

$$C.E. = zI - A$$

Finally, the stability radius, r , was calculated using the following relation where, $z \in \mathbb{C}$, $\Re(z) = 0$ implies z is purely imaginary, and σ_{min} was the operator that returned the smallest singular value:

$$r = \min_{\Re(z)=0} \sigma_{min}(zI - A)$$

Stability radius was calculated for different biomechanical postures across hip angles. Across the range of postures the maximum stability radius ($r=0.79$) was at a hip angle of -4° (Fig. 1.6b). This posture was least sensitive to changes in system parameters and implies that this posture produced the most stable behavior in the presence of modeling errors, state estimate error, and external disturbances. A similar level of stability could also be found between two postures one more flexed and the other more extended. This similar level of stability can be used to predict similar dynamic responses across different postures (Bingham and Ting 2013).

1.6 Predictive Modeling in Neuromechanic: Vertical Leap Example

Finally, we provide a brief example of the predictive approach with an analysis of maximal vertical leaping. The analysis is not intended to be comprehensive but provides an approximation of the maximal leap behavior and predicts major

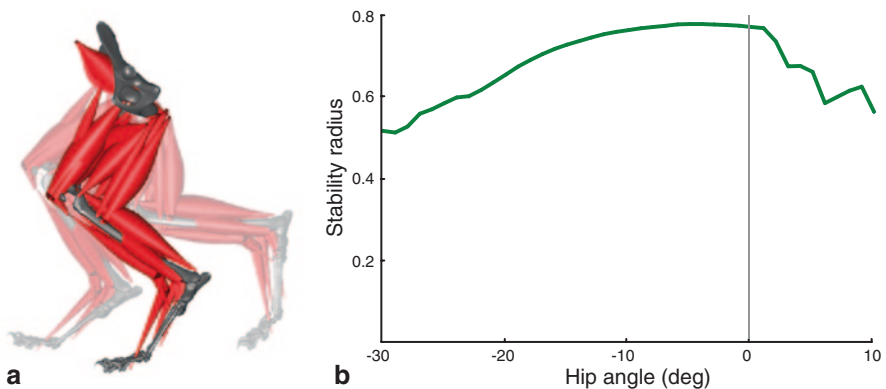


Fig. 1.6 a Cat hind-limb in neutral posture and *greyed* out postures showing limits of hip angle. b Stability radius across different hip angles with maximum stability occurring near the nominal posture

influencing factors by relying on optimization techniques instead of experimental data. The optimization is implemented in a Python module and the necessary files to reproduce the data are available at www.neuromechanic.com.

The leaping behavior is generated using the 2392 model with immobilized lumbar flexion/extension and metatarsophalangeal joints. A sequence of postures spanning a vertical center of mass displacement of 60 cm were generated with each posture satisfying the condition that no horizontal plane displacements of the center of mass took place and no angular momentum is required to transition between postures. A single normalized velocity \dot{v}_i was calculated at each posture as the average displacement between the current posture and the preceding and succeeding postures. Each normalized velocity was then replicated and scaled so that the vertical component of the center of mass velocity ranged between 0 and 4 m/s in intervals of 0.2 m/s. For each of the 2400 posture (q_i) and velocity (\dot{q}_i) combinations the maximum joint accelerations (a_i) that the model can produce which transitions the state to the “next” posture (q_{i+1}) was approximated by maximizing the vertical ground reaction force subject to:

$$\begin{bmatrix} \text{diag}(-2(q_{i+1} - q_i)) & \dot{v}_{i+1}^2 \end{bmatrix} \begin{bmatrix} a_i \\ \alpha^2 \end{bmatrix} = \dot{q}_i^2$$

where α is a free parameter that scales the dimensionless velocity (\dot{v}_{i+1}) at the q_{i+1} posture. The optimization was further constrained by the equations of motion (which relate muscle forces, endpoint forces, and joint accelerations) and the maximum and minimum forces that can be generated by each muscle. A Hill-type muscle model (Zajac 1989) was employed with an infinitely stiff series elastic element. The upper and lower muscle force bounds ultimately determine the maximum acceleration that can be generated and depend both upon instantaneous muscle kinematics defined by the particular posture/velocity combination and upon parameters of the muscle model including the active force-length curve.

To determine the sensitivity of maximal leap performance to the force/length relationship of muscle we calculated the maximal accelerations for each posture/velocity combination using three normalized force-length curve conditions called “constant”, “Thelen”, and “splined”. In the “constant” condition the active force-length curve is a constant value of 1 so that the force generated by a muscle does not depend on its length. The “Thelen” condition uses the Gaussian shaped Thelen-muscle model (Thelen 2003) implemented in OpenSim which allows muscles to produce active force even when operating outside physiological ranges. The “splined” condition uses a splined fit to a dimensionless adaptation of the original isometric force-length relationship (Gordon et al. 1966).

The maximum vertical center of mass acceleration determined for each posture/velocity combination under each of the three force length curve conditions are shown as contour plots in Fig. 1.7. White space in the contour plots indicates a state space location for which no feasible set of muscle forces can bring to pass the desired vertical transition. The relationship between fiber length (normalized to

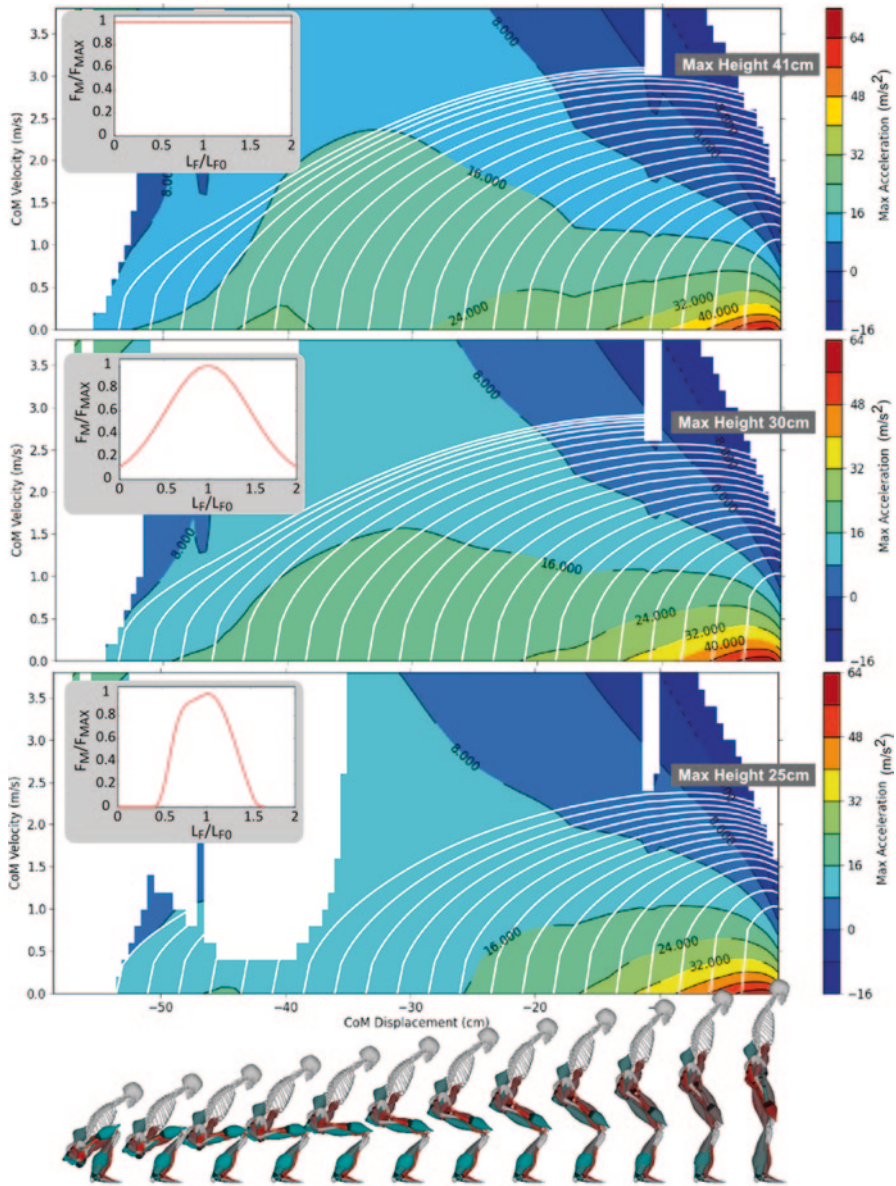


Fig. 1.7 Maximum acceleration contour plots for the 2392 OpenSim model during a vertical displacement of the center of mass under three force-length relationships: constant (*top*), Gaussian (*middle*) as in Thelen et al. 2003, and a splined fit of the force-length relationship reported in Gordon et al. 1966 (*bottom*). *White lines* through the contour plots indicate a trajectory which begins at zero joint velocity and moves vertically upward through the trajectories. The maximum final velocity achieved near the upright posture would result in a maximum vertical displacement of 41, 30, and 25 cm for the constant, Thelen, and splined relationships respectively

optimal fiber length) and isometric force (normalized to maximum force) for each condition are plotted and inset into the corresponding contour plot. Trajectories that follow the maximum accelerations and begin from a variety of initial conditions (various initial postures and zero joint velocities) are shown as white lines over the contour plots. Dynamic counter movements (where the center of mass moves down before moving up) provide no benefit in this model since the muscles do not have series elasticity for energy storage.

The contours of the maximum accelerations (and correspondingly the maximum leap trajectory) are smaller for the splined force-length curve (25 cm) than for the Thelen force-length curve (30 cm) and the constant force-length curve (41 cm). These values are not necessarily the true global maximum for each condition since some constraints placed on the system are overly stringent. For example allowing small net moments and horizontal plane ground reaction forces might increase the leap height as may slightly different joint trajectories, unlocking the metatarsophalangeal joint, and, of course, modeling the energy storage capacity of the musculotendon unit. However since the maximal leap behavior was not dictated *a priori* by experimental data but, instead, discovered through optimization techniques each of these possible constraints may be relaxed providing new behaviors and testable predictions in each case.

While knowing the performance characteristics (e.g. how high can my model jump?) of the neuromechanical model is useful it is more useful still for the modeler to have an understanding of the sensitivity of those performance characteristics to the particular parameters used. In this case the variation in contour plots between the three force-length relationship conditions provides an understanding of how the choice of muscle model affects real performance. In particular we can see the range of limb postures where the Thelen-model allows state transitions while a more traditional force-length curve prohibits them. A more complete analysis would assess sensitivity to the parameters discussed previously.

1.7 Conclusion

To assist in predictive modeling we have developed a platform, Neuromechanic, that emphasizes an integrated, performance and control-oriented approach to neuromusculoskeletal modeling. We believe that the predict and test paradigm is an important step towards simulations that are capable of aiding in understanding performance and safety of workers, athletes and soldiers. These tools also help to form testable hypotheses for understanding the fundamental interactions between neural control and biomechanics as well as for designing better rehabilitation and preventative strategies for a wide range of injuries and neurological and musculoskeletal disorders.

References

- Bai YF, Siu K, Liu CKR (2012) Synthesis of concurrent object manipulation task. *Acm Trans Graph* 31
- Bingham JT, Ting LH (2013) Stability radius as a method for comparing the dynamics of neuro-mechanical systems. *IEEE Trans Neural Syst Rehabil Eng* 21:840–848
- Bingham JT, Choi JT, Ting LH (2011) Stability in a frontal plane model of balance requires coupled changes to postural configuration and neural feedback control. *J Neurophysiol* 106:437–448
- Brock O, Khatib O (2002) Elastic strips: a framework for motion generation in human environments. *Int J Robot Res* 21:1031–1052
- Bunderson NE (2014) Real-time control of an interactive impulsive virtual prosthesis. *IEEE Trans Neural Syst Rehabil Eng* 22:363–370
- Bunderson NE, Burkholder TJ, Ting LH (2008) Reduction of neuromuscular redundancy for postural force generation using an intrinsic stability criterion. *J Biomech* 41:1537–1544
- Bunderson NE, Bingham JT, Sohn MH, Ting LH, Burkholder TJ (2012) Neuromechanic: a computational platform for simulation and analysis of the neural control of movement. *Int J Numer Meth BioMed Eng* 28:1015–1027
- Burkholder TJ, Nichols TR (2004) Three-dimensional model of the feline hindlimb. *J Morphol* 261:118–129
- Cofer D, Cymbalyuk G, Reid J, Zhu Y, Heitler WJ, Edwards DH (2010) AnimatLab: a 3D graphics environment for neuromechanical simulations. *J Neurosci Meth* 187:280–288
- Coros S, Beaudoin P, van de Panne M (2010) Generalized biped walking control. *Acm T Graphic* 29
- Davoodi R, Loeb GE (2012) Real-Time animation software for customized training to use motor prosthetic systems. *IEEE Trans Neural Syst Rehabil Eng* 20:134–142
- Delp SL, Anderson FC, Arnold AS, Loan P, Habib A, John CT et al. (2007) OpenSim: open-source software to create and analyze dynamic simulations of movement. *IEEE Trans Biomed Eng* 54:1940–1950
- Erez T, Lowrey K, Tassa Y, Kumar V, Koley S, Todorov E (2013) An integrated system for real-time model predictive control of humanoid robots. *IEEE/RAS International Conference on Humanoid Robot.* (ed.), Atlanta GA
- Fitts PM (1954) The information capacity of the human motor system in controlling the amplitude of movement. *J Exp Psychol* 47:381–391
- Geijtenbeek T, van de Panne M, van der Stappen AF (2013) Flexible muscle-based locomotion for bipedal creatures. *Acm T Graphic* 32
- Geyer H, Herr H (2010) A muscle-reflex model that encodes principles of legged mechanics produces human walking dynamics and muscle activities. *IEEE Trans Neural Syst Rehabil Eng* 18:263–273
- Gordon AM, Huxley AF, Julian FJ (1966) The variation in isometric tension with sarcomere length in vertebrate muscle fibres. *J Physiol* 184:170–192
- Huxley AF (1957) Muscle structure and theories of contraction. *Prog Biophys Biophys Chem* 7:255–318
- Ijspeert AJ (2008) Central pattern generators for locomotion control in animals and robots: a review. *Neural Netw* 21:642–653
- Jain S, Ye YT, Liu CK (2009) Optimization-Based interactive motion synthesis. *Acm Trans Graphic* 28
- Jo S, Massaquoi SG (2004) A model of cerebellum stabilized and scheduled hybrid long-loop control of upright balance. *Biol Cybern* 91:188–202
- Jo S, Massaquoi SG (2007) A model of cerebrotocerebello-spinomuscular interaction in the sagittal control of human walking. *Biol Cybern* 96:279–307
- Liddell EGT, Sherrington C (1924) Reflexes in response to stretch (myotatic reflexes). *Proc R Soc Lond Ser B Biol Sci* 96:212–242
- Markin SN, Klishko AN, Shevtsova NA, Lemay MA, Prilutsky BI, Rybak IA (2010) Afferent control of locomotor CPG: insights from a simple neuromechanical model. *Ann N Y Acad Sci* 1198:21–34

- Sarvestani IK, Kozlov A, Harischandra N, Grillner S, Ekeberg O (2013) A computational model of visually guided locomotion in lamprey. *Biol Cybern* 107:497–512
- Sherrington CS (1910) Flexion-reflex of the limb, crossed extension-reflex, and reflex stepping and standing. *J Physiol* 40:28–121
- Thelen DG (2003) Adjustment of muscle mechanics model parameters to simulate dynamic contractions in older adults. *J Biomech Eng* 125:70–77
- Todorov E, Erez T, Tassa Y (2012) MuJoCo: a physics engine for model-based control. 2012 Ieee/Rsj international conference on intelligent robots and systems (Iros), pp 5026–5033
- Tomita N, Yano M (2007) Bipedal robot controlled by the basal ganglia and brainstem systems adjusting to indefinite environment. 2007 Ieee/Icme international conference on complex medical engineering, vol 1–4, pp 116–121
- Welch TDJ, Ting LH (2008) A feedback model reproduces muscle activity during human postural responses to support-surface translations. *J Neurophysiol* 99:1032–1038
- Zajac FE (1989) Muscle and tendon—properties, models, scaling, and application to biomechanics and motor control. *Crit Rev Biomed Eng* 17:359–411