4-17-2009

Neuromechanical Analysis of Locust Jumping

David Wayne Cofer

NEUROMECHANICAL ANALYSIS OF LOCUST JUMPING

by

DAVID WAYNE COFER

Under the Direction of Dr. Donald H. Edwards

ABSTRACT

The nervous systems of animals evolved to exert dynamic control of behavior in response to the needs of the animal and changing signals from the environment. To understand the mechanisms of dynamic control, we need a means of predicting how individual neural and body elements will interact to produce the performance of the entire system. We have developed a neuromechanical application named AnimatLab that addresses this problem through simulation. A computational model of a body and nervous system can be constructed from simple components and situated in a virtual world for testing. Simulations and live experiments were used to investigate questions about locust jumping.

The neural circuitry and biomechanics of kicking in locusts have been extensively studied. It has been hypothesized that the same neural circuit and biomechanics governed both behaviors, but this hypothesis was not testable with current technology. We built a

neuromechanical model to test this and to gain a better understanding of the role of the semi-lunar process (SLP) in jump dynamics. The SLP are bands of cuticle that store energy for use during jumping. The results of the model were compared to a variety of published data and were similar. The SLP significantly increased jump distance, power, total energy, and duration of the jump impulse.

Locust can jump precisely to a target, but also exhibit tumbling. We proposed two mechanisms for controlling tumbling during the jump. The first was that locusts adjust the pitch of their body prior to the jump to move the center of mass closer to the thrust vector. The second was that contraction of the abdominal muscles during the jump produced torques that countered the torque due to thrust. There was a strong correlation relating increased pitch and takeoff angle. In simulations there was an optimal pitch-takeoff combination that minimized tumbling that was similar to the live data. The direction and magnitude of tumbling could be controlled by adjusting abdominal tension. Tumbling also influenced jump elevation.

Neuromechanical simulation addressed problems that would be difficult to examine using traditional physiological approaches. It is a powerful tool for understanding the neural basis of behavior.

INDEX WORDS: Neuromechanical, Biomechanics, Neural network, Simulation, Locust, Jumping, Kicking, Semi-lunar process, Flight, Tumbling, Elevation, Hill muscle, Invertebrate

NEUROMECHANICAL ANALYSIS OF LOCUST JUMPING

by

DAVID WAYNE COFER

A Dissertation Submitted in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

in the College of Arts and Sciences

Georgia State University

2009

NEUROMECHANICAL ANALYSIS OF LOCUST JUMPING

by

DAVID WAYNE COFER

Committee Chair: Donald H. Edwards

Committee: Gennady Cymbalyuk
William J. Heitler

Electronic Version Approved:

Office of Graduate Studies
College of Arts and Sciences
Georgia State University
May 2009

DEDICATION

I would like to dedicate this dissertation to my mother, Shirleyn Cofer. She was the strongest and most determined person that I have ever known. She faced her illness with dauntless courage, and she used her strength, courage, and decency to mold me into the person I am today. I hope to live up to her shining example. I would like to share with you one of her favorite poems.

When things go wrong as they sometimes will,
When the road you're trudging seems all uphill,
When the funds are low, and the debts are high,
And you want to smile, but you have to sigh,
When care is pressing you down a bit,
Rest if you must, but don't you quit.
Life is strange with its twist and turns,
As everyone of us sometimes learns,
And many a failure turns about,
When he might have won had he stuck it out;
Don't give up though the pace seems slow,
You may succeed with another blow.
Success is failure turned inside out,
The silver tint of the clouds of doubt,
And you can never tell how close you are,
It may be near when it seems so far;
So stick to the fight when you're hardest hit,
It's when things seem worst,
That you must not quit.

ACKNOWLEDGEMENTS

It has taken a lot of help to reach this point, and I would like to thank all of the people who have helped me get here. First, I would like to thank my advisor, Dr. Donald H. Edwards. His advice and guidance has been invaluable. He has taught me how to do scientific research, and more importantly, how to communicate that work in a clear and professional manner. I would also like to thank the other members of my committee, Dr. Gennady Cymbalyuk, and Dr. William Heitler. Both of them have been a valuable sounding board for ideas, and have helped edit and refine the work presented in this dissertation. Professionals in the world of academia are extremely busy, and I greatly appreciate the time they have all dedicated to helping my professional development.

I would also like to thank several members of the Edwards lab that have helped me during my time as a graduate student, Dr. Brian Antonsen, Dr. Nadia Spitzer, Dr. Jeff Triblehorn, and Dr. Pete Issa have all provided valuable feedback on posters, papers, and my thesis. Their meticulous attention to detail helped me learn how to produce figures, and text of professional quality.

Many thanks to the best graduate coordinator there is, LaTesha Warren for her support and for putting up with all the bureaucratic problems I ran into during my studies. I also thank the Brains & Behavior program of Georgia State University for providing me the financial support during my time as a graduate student here. I would also like to thank Dr. Karen Thompson of Agnes Scott College for providing me with locusts, and teaching me how to keep them alive and healthy.

Finally, I would like to thank my wife Stephanie and the rest of my family. The success of my studies would have not been possible without their support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

COM	Center of mass
CF	Coxa-femoral
FT	Femoral-tibia
SLP	Semi-lunar process
TC	Thoracic-coxa
TOPV	Takeoff pitch velocity
TUPV	Tumbling pitch velocity

CHAPTER 1.

GENERAL INTRODUCTION

A major goal of neuroscience is to understand how the nervous system is organized to control behavior. The nervous system gathers sensory information about the body's relationship to the world, and then makes decisions and issues motor commands which change that relationship. The dynamics of the interaction among the central nervous system, the body, and the world are central to the functional control of behavior (Chiel and Beer 1997). To govern behavior correctly, the nervous system must both predict and respond to the consequences of the animal's own movements and behavior, and do so on a millisecond to second time scale.

Despite many experimental successes in the analysis of the nervous system and behavior, the dynamic relationship between nervous function and the body is poorly understood. The kinematics and dynamics of many behaviors have been described, and the neural circuitry for some of these behaviors has been mapped in anesthetized or restrained animals, or in isolated tissue preparations. However, we lack a means of predicting how the function and behavior of individual neural and body elements will affect the performance of the entire system and the behavior of the animal.

Neuromechanical Simulation

The emerging field of neuromechanical simulation provides a promising approach to this problem. Computational models of the relevant neural circuits, body parts, and the physical world simulate the neural and biomechanical mechanisms of a behavior simultaneously in a physically accurate environment (Pearson, et al. 2006). These types of simulations have a number of benefits that complement the traditional, purely physiological approach to biology. It

is much easier to make changes to a simulated system, and this allows multiple alternative hypotheses to be tested quickly. Also, unlike in living systems, all of the neural and physical variables are available for viewing by the researcher. Often it would be very helpful to have a better understanding of which variables or parameters are more important, or more sensitive to changes. This can be quite difficult to do in living system, but it is a simple matter of varying the parameters and comparing the results from a number of simulations.

Simulation systems for either neural networks or biomechanics have been used for some time now. Genesis and Neuron are two of the more popular software systems for modeling biologically realistic neural systems (Bower and Beerman 2007, Hines and Carnevale 2001). They allow users to make detailed electro-chemical models of a specific neuron or of entire networks in order to gain a better understanding of the working of the brain. One of the most widely used biomechanical simulators is OpenSim, which was based on the popular SIMM application (Delp and Loan 2000, Delp, et al. 2007). It provides a way to make detailed models of the musculo-skeletal system that can be utilized in dynamic simulations of movement and posture.

Computers have only recently become powerful enough to combine these two types of simulations to close the sensory-motor feedback loop and begin investigating the neural basis of behavior. One of the ways that neuromechanical simulators have proved most useful is in determining the role of sensory feedback during the dynamic process of movement. This is an extremely difficult problem to tackle in the live animal due to technical limitations. It is often vary hard to selectively stimulate or remove different sensory receptors to determine their role in movement. Simulation provides a method to do this, and it has already proven to be quite useful for a number of different animal models.

Research on walking in cats suggested that the transition from stance to swing was heavily influenced by two sensory signals. One signal was the unloading of the leg near the end of stance (Duysens and Pearson 1980, Whelan, et al. 1995), and the other was the amount of hip extension (Grillner and Rossignol 1978, Hiebert, et al. 1996). However, it was not technically possible to completely isolate these signals to determine their relative importance. Neuromechanical simulation was able to address this problem (Ekeberg and Pearson 2005). They discovered that the leg unloading signal could on its own produce robust walking behavior, but when only the hip extension was used it led to abnormal walking patterns that eventually led to the cat tripping and falling. This led them to conclude that the leg unloading signal was probably the crucial circuit for stance to swing transition, and has helped them to generate new ways to test this hypothesis physiologically.

Neuromechanical simulation has also proven useful for studying insect locomotion. The central pattern generators (CPG) for the leg joints of the stick insect are unable to intrinsically coordinate the movements of the legs to produce walking behavior without sensory feedback (Uuml, et al. 1995). Neuromechanical simulation and robotic applications have been successful in reproducing the walking behavior of the stick insect (Beer, et al. 1997, Cruse, et al. 1995a, Cruse, et al. 1998). Examination of the sensory signals of the leg revealed that they had a direct effect on the timing of the neural bursts of the CPG (Ekeberg, et al. 2004). They were also able to postulate that a specific sensory signal was important for lifting the leg even though the sensory receptor for it has not yet been found.

Another animal model that has benefited from neuromechanical simulation is the lamprey. The neural network in the spinal cord responsible for swimming in the lamprey has been worked out in considerable detail (Grillner, et al. 1995, Grillner, et al. 1998, Grillner and

Wallen 2002). However, it has been difficult to determine the functional role of a group of stretch-activated sensory neurons that synapse directly onto the rhythm-generating circuitry (Di Prisco, et al. 1990). Neuromechanical simulations have been able to reproduce swimming behavior at different speeds, and the ability to produce turning behaviors (Ekeberg 1993, Ekeberg and Grillner 1999). They were also able to determine that the sensory receptors had little impact on swimming in calm water, but they did in turbulent situations. When the simulated lamprey attempted to swim through a region where the water flowed perpendicular to their motion they were unable to do so if they did not have stretch receptor feedback. When that sensory feedback was included they were able to do so easily (Ekeberg and Grillner 1999).

Problems in Neuromechanical Simulation

Most existing computational simulations of animal behavior are developed by writing programs specific to the project at hand. To do this, researchers need to develop sophisticated computer programs to solve complex mathematical equations related to neural functioning and physical interactions, maintain proper data structures, conduct simulation, record data, and visualize the results by animating two dimensional or three dimensional virtual animal models. The benefit of this approach is that users have control over every detail of the simulation. The drawback is that it requires expertise over a vast range of different disciplines. It also typically requires users to learn enough about programming to be able to configure the system. Some of these problems can be avoided by using toolkits such as Matlab or Simulink to develop the simulations. Although Matlab and Simulink are powerful toolkits and can relieve the user from writing and debugging complicated computer programs, they are not designed specifically for simulating neuro-musculo-skeletal systems. Therefore it is often necessary to integrate Matlab or

Simulink with other tools in order to build a complete simulation (Davoodi and Loeb 2002, Davoodi, et al. 2004, Ridderstrom 2003). Such integration and software development is often non-trivial, and requires a high level of computer programming experience. This means that most projects featuring neuromechanical simulation require the collaboration and management of a number of individuals, and this can be both time consuming and expensive.

Another problem is that most existing neuromechanical simulations were custom-built for human or a particular animal model. As a result, the model, simulation, and visualization are often tightly coupled, making it difficult for one research group to exchange or reuse software modules developed by other research groups. This drastically reduces the usefulness of the model because it cannot be easily shared and modified by the scientific community.

Finally, the user interfaces of the existing computer simulations are often very minimal and difficult to use. For example, the neural network model and the animal body model are often stored in script files (Ekeberg, et al. 2004, Ekeberg and Pearson 2005, Reichler and Delcomyn 2000). If a user wants to modify these models, they have to manually edit the script files. The usefulness of these types of simulations would be greatly improved by making it intuitively easy to build and modify the neural circuits and body models.

The Locust Jump

The locust jump is a system that is well suited to being studied with a neuromechanical simulation. Several decades of research has been performed to understand the details of the neural circuit and the biomechanics responsible for producing the jump behavior. This makes it a relatively straightforward process to build models of the hypothesized biomechanical and the neural control systems, and then connect them together to test the neural control of jumping.

This is useful because even though a great deal is known about how locusts jump, there are still unanswered questions that can be addressed. Below is a brief introduction of what has been discovered about how locusts jump and how they control their trajectory to jump precisely to a specific target.

Biomechanics and Neural Control of Locust Jumping

The ability to escape predators is of great evolutionary importance. Many animals, including the locust, have evolved to specialize in jumping as a method of escape and locomotion. A common problem faced by all jumping animals is that while very rapid movements are required to propel the body, the ability to generate force in muscles decreases with the speed of muscle shortening (Hill 1970). Consequently, it is difficult for muscle contraction to produce the larger forces required for high acceleration over the short time needed for escape. Locusts have evolved a specialization of the femur-tibia joint of their metathoracic legs that allow them to overcome this problem. On the distal end of each metathoracic femur, the locust has a pair of highly sclerotized portions of cuticle that are called the semi-lunar processes (SLP). The SLP bends like a bow to store energy that can later be used to power the jump (Bennet-Clark 1975). This allows the locust to store energy in the SLP by a slow contraction of the powerful extensor muscle over a span of hundreds of milliseconds, and then later release that energy rapidly to power the jump.

The motor circuitry responsible for the jump has been inferred from intracellular recordings from metathoracic neurons obtained during the kick, when the locust body is held stationary (Heitler 1988, Heitler and Burrows 1977a, Heitler and Burrows 1977b). The similarity between the motor program responsible for the kick and the neuromuscular activity produced

before and during the jump has led to the suggestion that the kick and jump motor programs are the same (Heitler and Burrows 1977a). Technical limitations currently prevent an explicit test of this hypothesis.

The locust motor program consists of three phases (Burrows 1995, Heitler and Burrows 1977a). The first phase is cocking the leg. The locust prepares for a jump by activating the flexor tibia muscle to bring the tibia into a fully flexed position. The second phase is a period of co-contraction where both the extensor and flexor tibia muscles are active. The extensor tibia muscle slowly contracts and stores energy in the extensor apodeme, in the SLP, and in the leg cuticle. As tension builds in the flexor tibia, its distal tendon passes over a cuticular invagination in the ventral aspect of the distal ventral femur (Heitler's lump, Bennet-Clark, 1975). This enables the flexor tibia tendon to attach to the tibia at an angle of nearly 90^{o} when the tibia is fully flexed, thereby maximizing its mechanical effectiveness in maintaining tibial flexion against the increasing tension in the much larger extensor tibia muscle (Heitler 1974). The lump also acts as a catch or lock on the tendon that helps keep the tibia flexed while the flexor muscle has tension. In the third phase, the jump is triggered when both the flexor muscle and its motor neurons are inhibited. When the flexor tension drops below a threshold, the tendon slips off the catch and the tibia is rapidly extended to produce the jump (Heitler 1974).

Locust Trajectory Control

Locusts can expertly control the trajectory and power of their jump to reach a specific target (Eriksson 1980). For a locust to jump precisely to a target it must be able to control a number of variables including the jump length, elevation, and yaw of the jump. Locusts use a peering behavior to determine the distance to a target (Eriksson 1980, Sobel 1990). During

peering they use their front legs to translate their body and head from side to side prior to jumping. This works because apparent motion of objects in their line of site depends on how far away those objects are located. Jump velocity is directly related to target distance, and when experiments were performed to artificially manipulate the motion parallax the jump distance and velocity changed correspondingly (Sobel 1990). The amount of power applied during the jump is controlled by the tension level of the tibia extensor muscle of the metathoracic legs (Bennet-Clark 1975). In defensive kicks the force is related to the number of FETi spikes and the duration of the co-contraction phase (Burrows 1995). The amount of energy that is stored determines the takeoff velocity, and thus how far it will jump (Bennet-Clark 1975, Sobel 1990).

Locusts can also direct their jump up to 50^{o} away from the direction of the long axis of their body when flexion begins. They control the azimuth angle of their jump by producing yawing movements of the body with the front and middle legs (Santer, et al. 2005). This was tested by rolling a ball down a ramp towards a freely behaving locust to induce an escape jump. The ramp was positioned to the side of the locust to see whether they would jump away from the ball. When flexion of the rear tibia began the front leg ipsilateral to the ball extended, while the contralateral leg flexed. This rotated the axis of the body away from the stimulus, and changed the azimuth direction of the jump. An alternate hypothesis was that differences in force levels of the two rear legs could alter jump trajectory, but no correlation was found between the power or trigger timing of the rear leg muscles and azimuth trajectory (Santer, et al. 2005).

Locusts must also be able to control the elevation of the jump. Initially it was thought that thrust for the jump was split into two separate components, with the initial portion of the thrust going downward and then shifting more diagonally later in the jump (Heitler 1977). A problem with this idea though was that elevation control would be directly tied to the amount of

thrust, making it difficult to set either variable independently. However, recent mathematical analysis has cleared up this problem and showed that control of elevation can be decoupled from the power for the jump, allowing these two variables to be controlled independently (Sutton and Burrows 2008). It was shown that thrust for the jump is applied continuously along a straight line drawn from the distal end of the tibia through the proximal end of the femur, and the angle of this line with the horizontal plane is termed the beta angle. This allows elevation to be controlled by rotating the metathoracic femur, thus altering the beta angle and the thrust vector.

These three mechanisms provide the locust with the means to aim its jump to a specific target and hit it accurately. While locusts have the ability to jump precisely, high speed video shows that they often rapidly tumble during the jump, sometimes making several complete revolutions throughout the jump trajectory (Visual observations, and Pond, 1972). This makes the orientation at their final destination, and during the ballistic phase of the jump, unpredictable. In such a tumbling situation, they are as likely to land on their back or head as they are to land on their belly. Since they are able to jump in both a predictable, controlled manner, and in an erratic manner, then it seems likely that they have some mechanism for controlling tumbling during takeoff, and that this ability may have evolutionary advantages that make it important. The existing mechanisms do not account for the role of tumbling in trajectory control.

Neuromechanical Simulation and Locust Jumping

There are still a number of open questions related to the neural and biomechanical control of locust jumping. Due to technical limitations, a neuromechanical simulation is the best way of addressing some of these issues. A good example of this is the fact that what is known about the neural motor program that is thought to be responsible for initiating the jump was actually

learned by studying the kick. These behaviors are so similar that it seems reasonable to conclude that they both produced by the same neural circuits, but since it is not currently possible to do the necessary electrophysiology experiments during a jump it is not possible to know for sure. A neuromechanical simulation will also not be verify that the locust uses the same circuit for both behaviors, but what it could do is verify that the same neural circuit is sufficient to produce both behaviors.

Simulation could also be helpful in analyzing the role of the SLP in jump dynamics. The function of the SLP during the jump has been inferred from its movement during kicks as recorded by high-speed video and from calculations of the energy stored in the extensor apodeme, the SLP, and femur cuticle (Bennet-Clark 1975, Burrows and Morris 2001). The energy stored in the SLP and the timing of its release appears to be important for jump performance. High-speed video of locusts kicks show that the SLP does not begin unfurling until the tibia has rotated by more than 30^{o} (Burrows and Morris 2001). This suggests that the energy stored in the SLP may play a more important role in the later part of the jump impulse than it does in the beginning. However, damage to the SLP, which is an integral part of the leg joint, makes the locust unable to jump, and so makes comparisons of the jump performance of animals with and without a functional SLP almost impossible. However, addressing this issue in a neuromechanical simulation is straightforward, and it may provide valuable insights into the role of the SLP.

The control of tumbling in locusts is another area where neuromechanical simulation may prove useful. Comparisons between the behavior of simulated and real locust jumps may provide useful insights into which parameters are important for the control of tumbling. The

simulation could then provide a simple method of varying those parameters over a wide range to observe the resulting behavior.

Dissertation Outline

The second chapter of this dissertation is devoted to describing the neuromechanical simulation software we have built that is called AnimatLab. The chapter describes the components of the application and how they work and interact. It then goes on to demonstrate how AnimatLab can be used by presenting a brief example of a human stretch and withdrawal reflex.

The third chapter details the neural and biomechanical locust model, and compares jumping in the model to behavior in the real animal. An analysis of the role of the SLP in jump dynamics was also performed in this chapter. I found that the neural circuit for the kick motor program was able to reproduce the jump behavior. Simulations also verified that the SLP was important for the jump. The delay in SLP unfurling was analyzed and shown to be an important component in the ability of the SLP to increase the power of the jump.

The forth chapter examines the control of tumbling in a jump. Two control mechanisms are hypothesized and explored in more detail by comparing the results from live locusts to simulations. Data from the live locusts supported the two hypotheses, while simulations allowed the hypotheses to be explicitly tested in a virtual environment.

CHAPTER 2.

ANIMATLAB: A 3-D GRAPHICS ENVIRONMENT FOR NEUROMECHANICAL SIMULATIONS

In Preparation for submission: David Cofer, Gennady Cymbalyuk, James Reid, Ying Zhu, William J. Heitler, and Donald H. Edwards

David Cofer was responsible for the design, coding, and testing of the bulk of the AnimatLab application. The graphics subsystem used in the body plan editor was the work of James Reid, and the code for the Integrate-and-fire neural model originally came from Dr. Heitler, and was subsequently modified by David Cofer to add new functionality and work within the AnimatLab environment. The human reflex model was created by David Cofer. The text and figures were produced by David Cofer with contributions from Dr. Edwards, and both people were also heavily involved in the editing and rewriting process. Dr. Cymbalyuk and Dr. Heitler provided revisions for the final versions of the document.

Introduction

AnimatLab is a free, open-source, Windows®-based software tool written to provide a general simulator for neuromechanical processes of skeletal animals, both vertebrate and invertebrate (www.AnimatLab.com). It allows users to build neural circuit and biomechanical body models in a virtual physical environment, and then to record time series of any variable(s) while viewing an interactive, 3D animation of the simulated behavior.

Both model construction and the simulations are carried out in an integrated environment without having to cope with programming details. AnimatLab implements a simple point and click graphical interface that allows users to construct and edit neural network models and 3D biomechanical body models in a way similar to that in professional CAD tools or 3D modeling tools like Maya or 3DS Max (usa.autodesk.com). Written in C++ and .Net, the program requires no programming knowledge, but does assume familiarity with neural and muscle physiology. It is available with 45 video tutorials, and over 100 pages of help files that will guide new users in a click-by-click fashion through model construction and use of the program's different capabilities.

AnimatLab's object-oriented, modular architecture makes it highly extensible. Users who prefer to write their own programs to extend, supplement, or substitute capabilities will be able to do so by plugging their software modules into AnimatLab through a compatible interface. Users may concentrate on developing components of interest without having to develop the entire system. Simulations and modules can be readily shared between investigators to allow others to examine and extend a simulation.

AnimatLab Overview

AnimatLab has three interactive components: a graphical user interface (GUI) that enables model building and data graphing, coupled solvers for the neural circuit and biomechanics simulations, and an interactive 3-D animation of the model's behavior in a virtual Newtonian world. In the model building portion, a "Project Workspace" (Fig. 2.1A) sets environmental parameters, including gravitational acceleration, maximal surface friction, and the physics simulation time step. It sets animation and graphical display parameters, stimulus

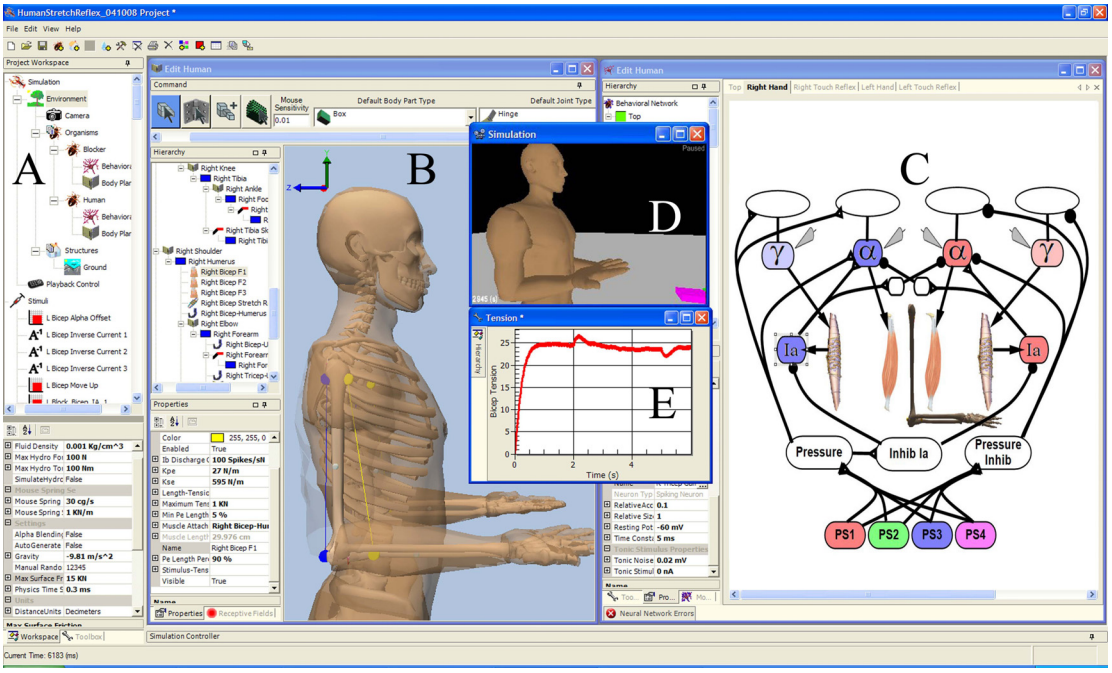

Figure 2.1. AnimatLab Screenshot. Screenshot from AnimatLab showing the different editing and simulation components of the program. **(A)** The Project Workspace panel where users manage all simulation properties and components, as well as playback, stimuli, and data plots. **(B)** The Body Plan editor that allows users to build organisms in a point-and-click manner. The top bar of this window lets users switch between viewing rigid bodies, joints and receptive fields, and it allows them to select the default body and joint types that are used when new parts are added. The panel in the top-left of this window displays the hierarchal connectivity of the body parts, and the property table below that lists all properties for an individual part. **(C)** The Behavior Editor allows users to drag-and-drop neurons, muscles, and other electrically excitable parts to create neural circuits, and then draw synaptic connections between these elements. Tabs at the top of the diagram list all the pages for the entire network, and the property panel on the bottom-left lists properties of the selected neuron. **(D)** The simulation window shows the 3-D graphical display of the simulated movement. Users can alter the 3-D view, and manipulate objects using the mouse. **(E)** Data charts allow the user to plot a time-series of any set of parameters in the simulation.

parameters, and it selects, lists, and provides access to editors for all the objects in the simulation: the nervous system and body of each animal, the ground and water surfaces, and all the fixed objects in the simulated environment. The "Body Plan Editor" (Fig. 2.1B) allows the user to assemble each model animal's body in a point-and-click, Lego®-like fashion from a variety of different part types. The "Behavior Editor" (Fig. 2.1C) allows users to construct the neurons and neural circuits that control the behavior of the model organism from another set of parts in a similar drag-and-drop fashion. Links are established between common elements (e.g., muscles, sensors) in the two editors, and the simulation is run by simultaneously operating and interacting solvers, the Vortex® simulator from CM-Labs for the biomechanics, and custom-made solvers for neural interactions. New neural solver plug-ins can be added by users, and each one can operate on a different time scale from the others and from the physics engine. The model animal's movements in the virtual environment are under neural control as it responds to simulated physical and experimental stimuli. The autonomous behavior of the model is displayed graphically in a 3-D animation (Fig. 2.1D) as the simulation runs, together with plots of the time-series responses of any designated set of neural or physical variables (e.g., membrane potentials, muscle forces, spatial displacements; Fig. 2.1E). Both the 3D animation and the data time-series can be recorded for off-line analysis, and the model parameter space can be explored by running multiple simulations with different parameter values simultaneously on different nodes of a grid computer. This makes it easy to systematically alter a variable to see how it influences the final behavior, and to see how sensitive it is to change and how important that variable is to the ultimate behavior of the animal.

Assembly of a Body Model

To build an animal model, users define rigid body parts in the Body Plan Editor (Fig. 2.1B), connect them with standard joints, enable them to move with actuators, and provide them with sensors to detect the environment. The connectivity of parts in the model is represented in a tree diagram (Fig. 2.1B, left). Several different types of body structures are currently available, including boxes, spheres, cones, cylinders, and polygon mesh models. A polygon mesh is a set of vertices and triangular faces that define the volume for that part. All parts are assumed to have a uniform density, and the distribution of mass throughout the mesh volume determines the moment of inertia for that part. Meshes allow animal models to have more realistic structures, both visually and dynamically, by providing a more accurate representation of the body than is possible with simple geometric shapes. Rigid body parts have user-specified physical properties including dimensions, density, center of mass, and drag; actuators include muscles, muscle spindles, motors, and springs; sensors include receptors for stretch, touch, odors, and tastes. Each body part is selected from a drop-down box and then placed, oriented, scaled, and shaped with a mouse in a 3-D GUI. A fill-in "Properties" table is visible (Fig. 2.1B, bottom left) for exact placements and specifications, including object density and color. Properties tables also contain parameter cells that are specific to the particular type of object, such as cells for the dashpot and spring constants of a Hill muscle model, or the angular limits of a planar hinge. A "description" cell in each properties table permits entry of text describing the object, or references to sources of the parameter values, or hyperlinks to those sources. Documentation of this sort is essential to distinguish between parameters and features based on experimental measurements from those that are made up to enable the model to work. This feature enables the model to be used as a database for the animal's neural circuitry and body structure.

Joints are modeled as movement constraints that prevent any motion of the connected body parts that is not allowed by that joint type. The currently available joint types are hinge, ball and socket, prismatic, and fixed. A hinge joint allows two bodies to rotate around a defined axis. The ball and socket joint allows two bodies to rotate freely around a common point. The prismatic joint allows relative translational movement along a single defined axis. The fixed joint welds two parts together into a unified component and prevents both rotational and translational movements.

Sensory receptors for touch and odor, for stretch of a muscle spindle, and for chemosensory stimuli are implemented in AnimatLab (photoreceptors and auditory receptors are planned for a future version). Representations of each sensory receptor are created in both the body model and the neural circuit model to map the field of sensory receptors onto the population of sensory neurons. In the body model, single mechano- or chemoreceptors, or an array of such receptors, can be placed on the body surface where contact with an appropriate stimulus will activate it (Fig. 2.2A). Each mechanoreceptor has a 2-D receptive field that describes its sensitivity to stimuli applied on the body surface in its vicinity. The distance from the contact to the center of the field is used to scale the force of the contact using a Gaussian function (Fig. 2.2A2). In the neural circuit model, the corresponding representation of a body receptor is linked through a transduction adapter to a neuron compartment that represents the sensory neuron excited by the receptor. The scaled force produced at the receptor is transduced by a sensory adapter into a generator current (Fig. 2.2A3) and passed into the sensory neuron (Fig. 2.2A4). Receptors with overlapping receptive fields may detect the same physical contact and evoke spike responses in their sensory neurons that reflect the different transduction current amplitudes. All physical processes, including joint rotation and extension of a stretch receptor,

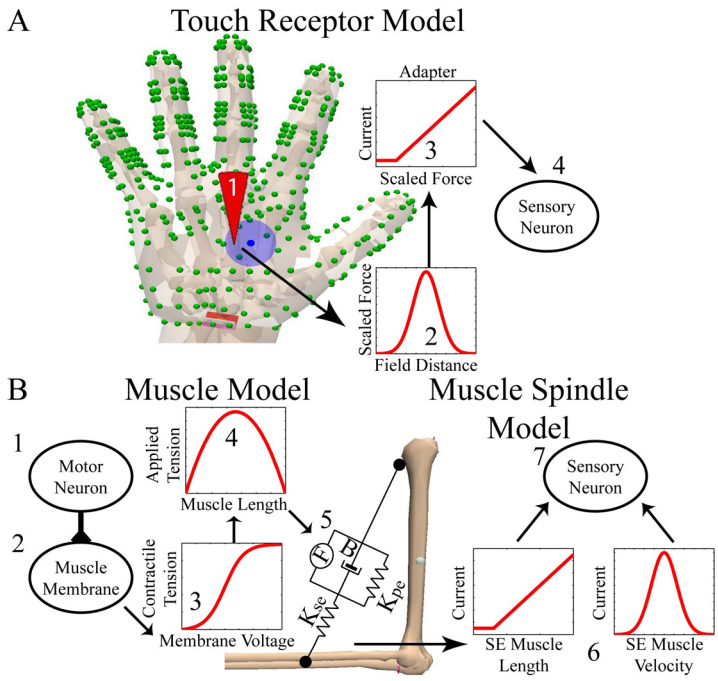

Figure 2.2. Modeling of sensory and motor systems in AnimatLab. (A) Mechanosensory receptive fields are located over the surface of the hand. The center of each field is shown as a green dot. When an object contacts the hand **(1)**, the field distance, or distance from the contact point to the center of the field, is calculated and used to scale the force of the contact **(2)**. Scaled force is used by an adapter **(3)** to calculate a generator current that is passed into a sensory neuron **(4)** as a result of stimulation. **(B)** Tension in a muscle or spindle is controlled by firing of one or more motor neurons **(1)**, which produce depolarizations in the membrane of the muscle **(2)**. The muscle membrane voltage is transduced into a contractile tension **(3)**, which is proportionally scaled based on the current muscle length **(4)**, and then applied at the force generator **(F)** in the Hill muscle model **(5)** to produce joint torques and movements. Stretch receptors use an identical model, but muscle properties like length and velocity are transduced into currents **(6)** that are applied to a sensory neuron **(7)**.

use this adapter transduction mechanism to produce a generator current that can be applied to a sensory neuron.

Both muscles and muscle spindles (or their invertebrate analogs) are represented by Hill-based models (Hill 1970, Shadmehr and Wise 2005a) that consist of a serial spring in series with the parallel combination of a parallel spring, a dashpot, and a force actuator (Fig. 2.1B, yellow biceps and blue triceps; Fig. 2.2B). Users attach a muscle to the body by adding attachment points to body segments on either side of a joint, and then stringing the muscle between these points. Multiple attachment points can be used to allow the muscle to span multiple joints, or to control the direction of the applied force. Muscle model properties are determined by the resting muscle length, the spring and dashpot constants, the stimulus-tension curve, and the length-tension curve. Neural control of muscle tension is mediated by a motorneuron (MN) (Fig. 2.2B1) that synapses onto a compartment that represents the common electrical properties of the muscle fibers that compose the muscle (Fig. 2.2B2). A sigmoidal stimulus-tension curve relates the force level of the actuator to the muscle membrane potential (Fig. 2.2B3). The actuator force is further scaled by the length-tension curve, an inverse parabola that determines the percentage of actuator force that is applied at a given muscle length (Fig. 2.2B4). The actuator force is then applied to the muscle to cause contractions (Fig. 2.2B5).

A muscle spindle responds to both imposed and excited stretch through an adapter that translates muscle properties into a generator current that is passed into a sensory neuron. Multiple adapters can be used to allow sensory responses to be as complex as required. For example, a Ia afferent may respond to current produced by one adapter sensitive to the serial-elastic (SE) length of a muscle and to current produced by a second adapter sensitive to the rate of change of SE length.

Motors are another way that movements can be generated by models within AnimatLab. The hinge and prismatic joints can be motorized to produce controlled rotational and translational motion. This can be useful in simulating passive movements imposed on a limb during an experiment, or in substituting for patterns of muscular force generation around a joint when only joint dynamics are known. A motor can be configured as a DC motor or as a servo motor. In a DC motor, an adapter converts a membrane voltage into a desired velocity. In a servo motor, the value from the adapter specifies the rotational position of the motor, and a feedback system maintains that position against perturbations.

Neuron and Network Models

In the Behavior Editor (Fig. 2.1C), single- or multi-compartment neurons and neural circuits are constructed by dragging elements from a toolbox onto a circuit editing page, and linking them by electrical or chemical synapses. Available elements include spiking integrate-and-fire neurons, non-spiking neurons, firing rate neurons, muscles, and muscle spindles. Neurons can be modeled as single compartments or as multiple compartments linked by ohmic coupling conductances. Each compartment is characterized by an input conductance, rest potential, and membrane time constant, and a set of other parameters that can include an initial spike threshold, after-hyperpolarization conductance, absolute refractory period, spike-frequency accommodation, calcium conductances with activation and inactivation parameters, Hodgkin-Huxley-like ionic conductances, and membrane potential noise. All parameters are displayed in a "Property Table" where they can be changed (Fig. 2.1C, bottom left).

Both conductance-based and current-injection synapses are available and can be added to the network by connecting neuron elements with cursor-drawn lines. Users can define new

instances of these synapse types as the need arises. The neural circuit can be enlarged by adding new circuit editing pages from the neural editor. Any individual neuron (or compartment) on one page can be represented on another page by an 'Off-Page Connector'. The network can also be divided into hierarchal neural subsystems. This simplifies the organization and maintenance of the entire network and allows it to be split out into functional subsystems. For example, it is possible to create a sub-network that controls movement of a leg, and then copy, paste, and modify it for each of the remaining legs.

Electrical properties

The primary neuronal element in AnimatLab is a single electrical compartment that consists of a capacitance in parallel with the serial combination of a conductance and constant voltage source that determines the resting membrane potential (MacGregor and Lewis 1977). The three different types of neuron models arise from this: Integrate-and-fire (I&F), non-spiking (NS), and firing rate (FR) models. In the I&F model, an action potential can be triggered when the membrane potential exceeds a voltage threshold. The membrane potential is then shifted to an adjustable peak voltage in one integration time step, during which no current flows. This spike is followed in the next time step by an "after hyperpolarizing potential" (AHP) membrane conductance that falls along an exponential time-course from a user-defined maximum back to zero with a defined time constant. The conductance is in series with an AHP voltage source, and together they produce the AHP that follows the spike. The spike threshold is set infinitely high for a brief period after a spike to produce an absolute refractory period. Spike threshold also varies continuously with "accommodation" as the membrane potential varies from rest potential. The threshold changes from its initial specified value toward a maximum that is proportional to

the difference between the cell's membrane potential and rest potential. The proportionality constant is the user-defined "relative accommodation" (between 0 and 1), and the rate of threshold change is governed by the user-defined accommodation time constant. Thus a rapid depolarization above threshold can induce an initial high rate of spiking that will slow or stop as the threshold rises, while a hyperpolarization will cause threshold to fall.

Other membrane conductances and series potentials can be implemented in the I&F model to provide additional current pathways. Voltage- and time-dependent Hodgkin-Huxley (HH) –like conductances can recreate both full action potentials and subthreshold responses, while calcium conductances enable plateau potentials and bursting responses. In both cases, users specify parameters that determine the voltage- and time-dependence of activation and inactivation variables that together determine the ionic conductance. Membrane potential noise can be added as random changes in potential that occur every time-step and are evenly distributed over the amplitude of the noise around the current membrane potential.

The NS model differs from the I&F model in having no voltage-threshold spiking mechanism. The HH and calcium conductances can be implemented to produce full action potentials, delayed rectification, plateau potentials and bursting.

Single I&F and NS compartment models can be linked together with electrical resistances, supplied in AnimatLab as "electrical synapses", to create multicompartment neuronal models in which each compartment represents a local region of the cell, with its own complement of membrane current paths.

The FR model is a more abstract model that is particularly useful in representing the average or combined effects of many parallel neurons (Beer 1990). The FR model uses the same parallel conductance and capacitance circuit to calculate the membrane voltage within the

neuron. No spikes are modeled in this system. Instead, the firing frequency is treated as a linear, continuous function of the membrane voltage once it has exceeded a threshold level. Firing frequency limits and the sensitivity of the spike frequency to membrane potential can be set, and the frequency is subject to accommodation as described above. Specific ionic currents are not implemented.

Synapses

A spike mediated chemical synapse is modeled as a rapid increase in post-synaptic membrane conductance triggered by a spike in a pre-synaptic I&F compartment. A delay between the pre-synaptic spike and the post-synaptic conductance rise can be set to reflect conduction time or synaptic delay; the conductance then declines exponentially back to zero with a user-defined time constant. The post-synaptic conductance is in series with a user-defined reversal potential that enables the synapse to be excitatory or inhibitory. Spiking synapses can also be defined as facilitating or anti-facilitating, voltage-dependent, or Hebbian. For each type, the postsynaptic conductance depends on another variable, such as the presynaptic spike frequency, the postsynaptic voltage, or the relative timing of pre- and postsynaptic spikes.

A non-spiking chemical synapse is one where the post-synaptic conductance depends on the pre-synaptic membrane potential of an NS or I&F compartment, and not on a presynaptic spike. The post-synaptic conductance varies linearly with pre-synaptic membrane voltage between threshold and saturation voltages, and is a constant minimum or maximum value outside that range.

An electrical synapse is modeled as a non-specific electrical conductance linking two neurons, either of which can be NS or I&F. Current flows from one neuron to the other in

proportion to the product of the difference between the neurons' membrane potentials and the fixed junctional conductance. In non-rectifying, ohmic synapses, the junctional conductance is constant. In rectifying electrical synapses, the junctional conductance varies linearly with the difference between the pre- and post-synaptic potentials.

Three types of synapses are available for the FR neurons; current, gating, and modulatory synapses. Current synapses directly pass current into the post-synaptic neuron based on the weight of the synapse and the firing rate of the pre-synaptic neuron. The gating and modulatory synapses function heterosynaptically. A gating synapse enables the presynaptic neuron to determine whether or not a current synapse from another presynaptic cell onto a common postsynaptic neuron will function. A modulatory synapse is similar, but instead of gating all of the heterosynaptic current, the modulatory synapse scales it in proportion to both the modulatory synapse's weight and the modulatory neuron's firing rate.

Simulation: Human Arm Flexion and Avoidance Reflex

We created a model of human arm flexion and avoidance reflexes to demonstrate AnimatLab's capabilities. A model of the human body was developed from a 3-D polygon mesh that was purchased online (www.exchange3d.com) and then broken into individual bones and body segments using the graphics program Blender (www.blender.org; Fig. 2.1B). Each bone and body segment was re-scaled to the appropriate dimensions to match published values (Clauser, et al. 1969). All parts are assumed to have a uniform density, and the distribution of mass throughout the mesh volume determined the moment of inertia for that part. The model consisted of separate bone and non-skeletal elements, in which the bones were inside the non-skeletal elements. The density of all bones was set to 1.9 g/cm^3 (Cameron, et al. 1999), and the

densities of the overlying non-skeletal segments (representing the skin, muscle mass, connective tissue and blood) were calculated by subtracting the mass and volume of the bone from the mass and volume of the overlapping body segment (Clauser, et al. 1969), and calculating the resulting mass/volume ratio. The bone and non-skeletal elements of a limb segment (e.g., the forearm) were bound together in the model by a fixed joint.

Here the focus was on arm flexion at the elbow, so the upper arm and body were fixed in space, the forearm and hand rotated around a hinge joint at the elbow, and the hand was connected to the forearm by a fixed joint (Fig. 2.1A). The position, orientation, and angular limit of the elbow hinge joint was set so that only the normal range of arm movements could occur.

Forearm movement in the model is driven by a biceps flexor muscle and a triceps extensor muscle and controlled by a pair of corresponding muscle spindles. Values for the springs and dashpot coefficients of the biceps and triceps muscle and muscle spindle models were obtained from Massone and Myers (Massone and Myers 1996, Myers and Massone 1997). The measured maximum tension for these muscles (Myers and Massone 1997) set the upper limits for the stimulus-tension curves, and the length-tension curve was assumed to follow measurements obtained for similar muscle types (McMahon 1984).

Reflex Circuitry

The model myotactic stretch reflex circuit (Fig. 2.3A) (Pearson and Gordon 2000, Windhorst 2007) enables the arm to resist perturbations away from its equilibrium position. The circuit contains descending tonic command neurons (FR models) that drive the alpha and gamma MNs (I&F models); they, in turn, excite the working and spindle muscles, respectively. Spindle

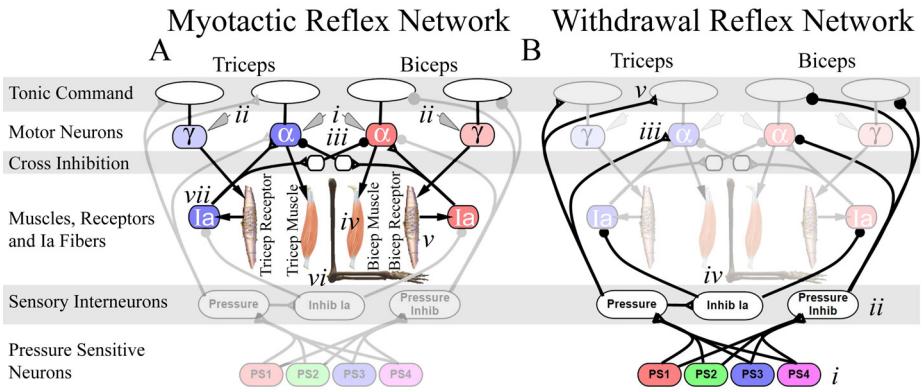

Figure 2.3. Model neural circuits controlling movements of the arm. **(A)** Tonic commands to alpha and gamma motor neurons (MNs) set the tonic MN activity level to obtain the initial arm equilibrium position. The myotactic reflex network mediates responses to deviations from the planned movement of the arm. To initiate arm flexion, depolarizing and hyperpolarizing currents (*i*) drive flexor and extensor alpha MNs (*iii*), respectively, to cause biceps contraction and triceps relaxation (*iv*), while gamma MNs are stimulated (*ii*) to maintain tension in the muscle spindles (*v*). Tensions in muscles produce joint torque and movement of the arm (*vi*). Responses of the Ia sensory neurons (*vii*) signal an error in the trajectory of a movement. **(B)** The withdrawal reflex mediates responses to noxious contacts. The reflex is activated by strong contacts on the palm of the hand that excite pressure sensitive neurons PS1-4 (*i*). The sensory neurons excite local interneurons and inhibit the myotactic reflex (*ii*) in a fast spinal feedback loop (iii) to move the hand away from the contacting stimulus (*iv*), while a slower central feedback loop changes the descending tonic commands (*v*) to produce a new limb posture.

tension generates depolarizing membrane current in a Ia sensory neuron (I&F) that may fire to provide afferent feedback to the alpha MNs and (I&F) inhibitory interneurons. The inhibitory neurons inhibit the antagonist alpha MNs.

The model withdrawal reflex (Fig. 2.3B) (Pearson and Gordon 2000) extends the arm rapidly away from a strong, noxious contact on the upturned palm of the hand and causes the limb to adopt a new posture away from the stimulus. The stimulus falls within the receptive fields of four model mechanosensory neurons (Figs. 2.2A, 2.3B: PS1-4) that each respond in proportion to the amount of force they experience. These sensory neurons excite a central interneuron (Fig. 2.3B, "Pressure") that excites the extensor MNs and inhibitory interneurons of both the flexor MNs and the stretch reflex for both muscles. This promotes extension of the arm and inhibits the stretch reflex. This is the spinal feedback loop. In the longer central feedback loop, the interneurons ascend to the tonic command neurons to alter their firing rate.

Responses

A voluntary arm flexion was simulated by command current activation of the MNs. At the outset, the tonic activity of the alpha MNs caused the flexor (bicep) and extensor (tricep) muscles to maintain constant tensions that kept the forearm at a 90^{o} angle relative to the upper arm (Fig. 2.4A). A step excitatory command current was applied at 1.0s to the biceps alpha MNs to simulate an arm flexion motor command (Fig. 2.4B). The brief rise in flexor tension caused the arm to flex rapidly at first, and then more slowly to adopt a new, flexed equilibrium position. When the command current was turned off, the equilibrium position reverted to the 90^{o} angle and the arm returned to that position. To produce a faster movement, a multiphase pattern of current stimuli evoked a corresponding pattern of flexor and extensor MN activity that started

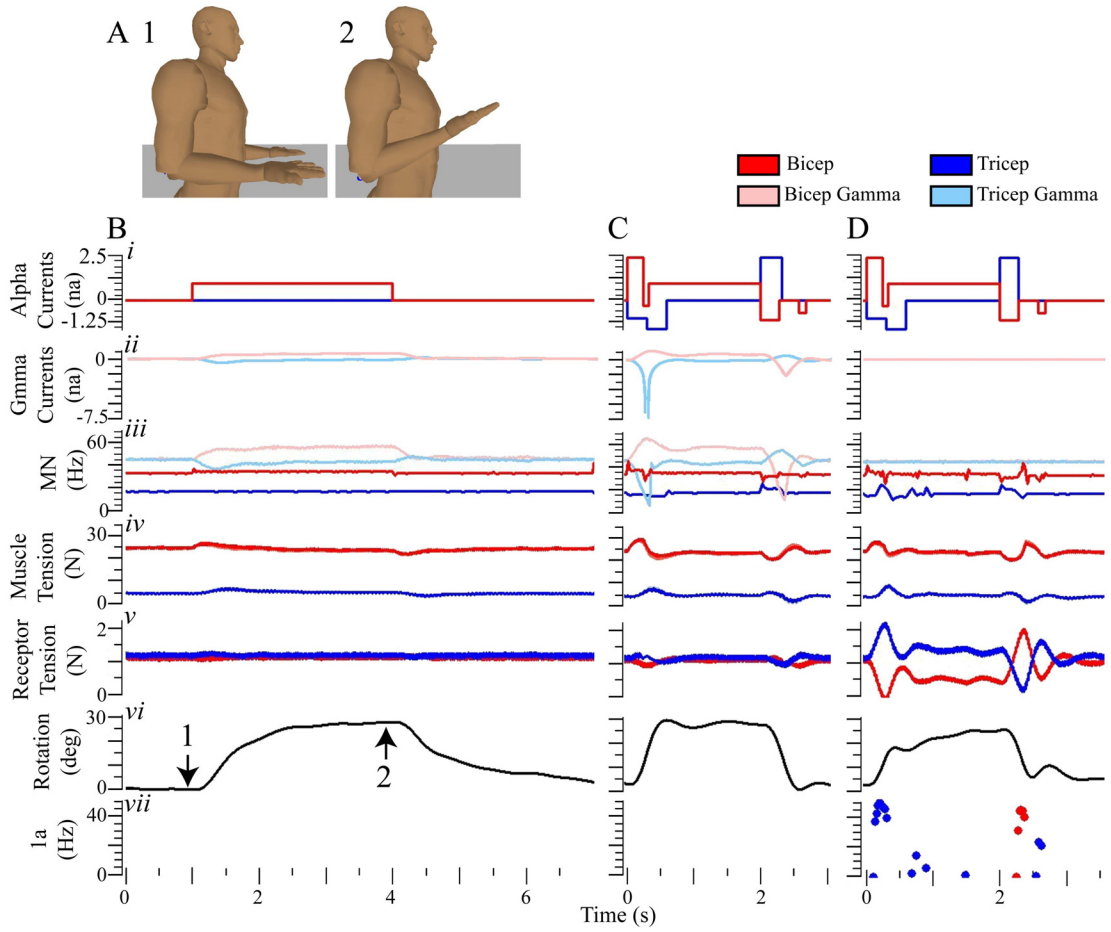

Figure 2.4. Simulation of voluntary arm flexion. (A) Screenshots of the animated arm movements from the beginning position (**1**) to the final position (**2**). **(B-D)** Plots of model variables; the italicized Roman numeral of each plot corresponds to a numbered item in the model diagram of Fig. 2.3A. *i*) The current stimuli applied to biceps and triceps alpha MNs. *ii*) The current stimuli applied to the gamma MNs. *iii*) The MN firing frequencies. *iv*) Tension of the biceps and triceps muscles. *v*) Tension of the muscle spindles (Receptors). *vi*) Elbow rotation. *vii*) Ia frequency response. **(B)** Responses to a current step applied only to the biceps alpha MN and to depolarizing and hyperpolarizing currents applied to the gamma MNs of both muscles. **(C)** Responses to a multiphasic command current pattern applied to the alpha and gamma MNs that produced a much faster movement by accelerating the arm quickly, braking its motion, and then maintaining the new position. **(D)** Responses to the same current pattern as in (C), but with gamma co-activation disabled.

the arm to flex quickly, then slowed the flexion, and finally maintained the arm at the new flexed equilibrium position (Windhorst 2007) (Fig. 2.4C). This pattern of stimulation produced a smooth movement of the arm to the target position in about half a second.

Command currents were also passed to the flexor and extensor gamma MNs to keep spindle tension constant during the movement. The biceps shortened during arm flexion and the triceps were stretched, which reduced the tension in the flexor spindle and increased it in the extensor spindle. One hypothesis of Ia reafference is that deviations from the flexor and extensor spindle base tension levels are interpreted as errors from the predicted voluntary movement and lead to changes in Ia afferent activity; these changes excite resistance reflexes to restore the spindle tension (Shadmehr and Wise 2005a). Increased excitation of the flexor gamma MN during arm flexion helped maintain the flexor spindle tension, while decreased excitation of the extensor gamma MN maintained the extensor spindle tension (Fig. 2.4B, C). A custom gamma efferent current is included in AnimatLab that calculates the command current time-course required for the spindle muscle to maintain a constant tension throughout a planned movement. This current emulates the hypothesized higher-level commands that produce the gamma co-activation and ensures that the muscle spindle reports little or no change in tension during unperturbed voluntary movements. If the arm is moved without gamma co-activation, the changes in spindle tension lead to activation of the stretch reflex, which will fight the movement and attempt to return the arm to its original position. This is illustrated in Fig. 2.4D, where a three-phase command current produced arm flexion, but the gamma co-activation was disabled so that gamma MNs received only tonic excitation. The tension in both the biceps and triceps spindles varied widely and oppositely from the desired constant levels, which led to excitation of the flexor Ia afferents during flexion, and the extensor Ia afferents during re-extension. These Ia

responses excited MNs to oppose the movements, which became jerky and failed to reach the flexion target.

The myotactic reflex was activated when the hand encountered a small, heavy block in its path as the arm flexed in response to the triphasic command (Fig. 2.5). The block was on a hinge joint that allowed it to move out of the way if the arm produced sufficient force. The arm flexed until it hit the block, which momentarily stopped the movement. The resulting deviation from the planned motion produced a rise in the biceps spindle tension when the interrupted flexion prevented the spindle from shortening in response to its gamma MN command (Fig. 2.5B, Receptor tension). The increased flexor spindle tension excited the Ia afferent, which added to the flexor alpha MN excitation. The extra flexor force was sufficient to push the arm past the block and overshoot the goal position. Additional reflex activity allowed the arm to settle to its target position.

The contact-withdrawal reflex was activated when the hand encountered a sharp spike placed in the path of the hand's movement during arm flexion (Fig. 2.6A). Four sensory receptors with overlapping pressure-sensitive receptive fields in the palm of the hand (Fig. 2.6B) responded to the contact by producing generator currents in the corresponding sensory neurons (Fig. 2.6C). The circular receptive fields have bell-shaped spatial-sensitivity functions, so that the current produced in each cell depended on the magnitude of the force and the distance between the point of force application and the field center. In the example presented, pressure sensor 2 (Fig. 2.6B, PS2) produced the largest current, while the sensory receptors on either side produced less current; the firing frequencies of their neurons were proportional to these currents. The sensory neurons excited a set of central interneurons (Fig. 2.3B) that triggered the withdrawal response. Of necessity, the withdrawal response had two phases, an immediate

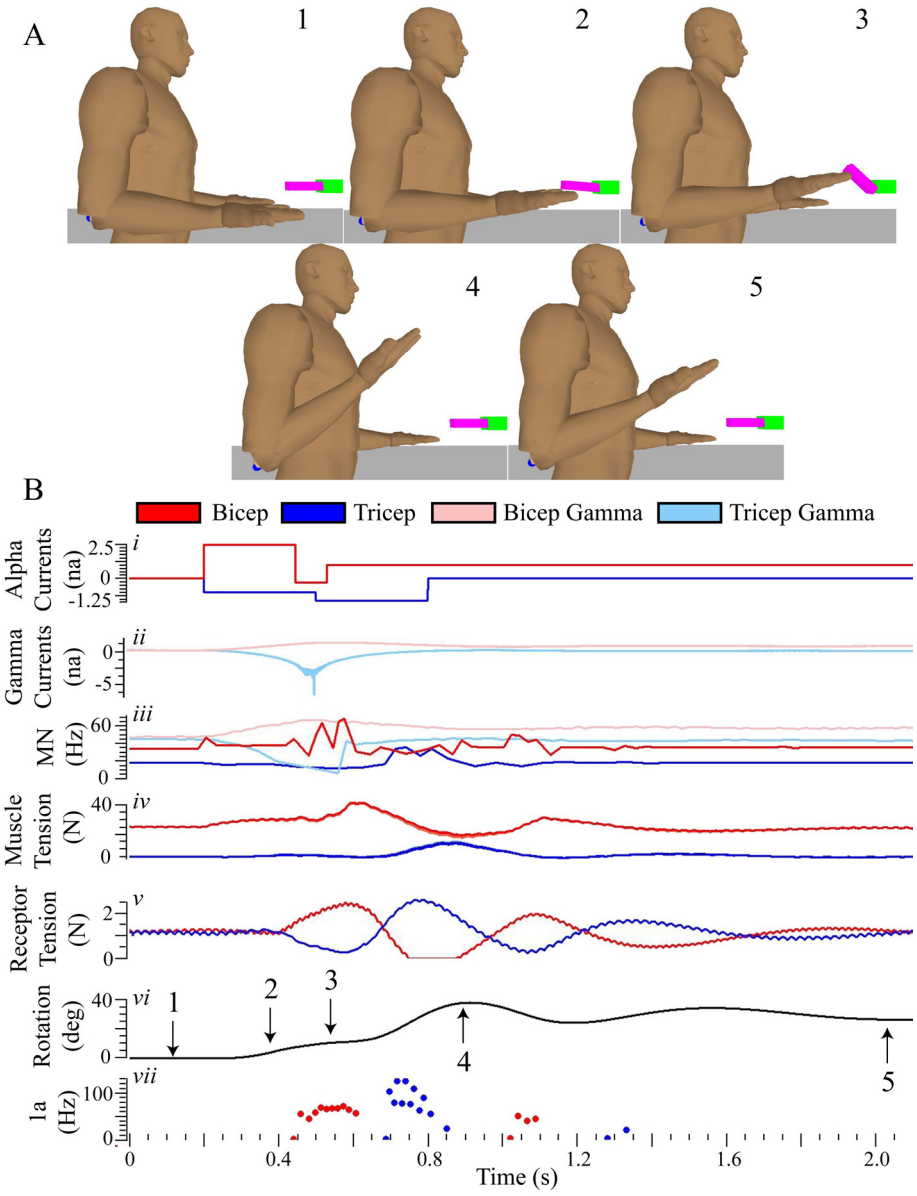

Figure 2.5. Stretch reflex response to perturbation during arm flexion. (A) Simulator animation showing the arm flexing **(1)**, then blocked **(2, 3)**, overcoming the hinged block and overshooting its target position **(4)**, and reaching its target position **(5)**. **(B)** The italicized Roman numeral in each plot refers to the corresponding numbers in Figs. 2.3A and 4B. Stimulation of the MNs as in Fig. 2.4C caused the arm to flex quickly and the biceps spindle to shorten, thereby keeping its tension constant. When the hand struck the hinged block, the biceps spindle no longer shortened at the planned rate and the spindle tension increased, exciting the biceps Ia sensory neurons. The Ias excited the bicep alpha MN, and inhibited the triceps alpha MN, producing an increased flexion torque that was enabled the arm to overcome the block. Overcompensation then dampened to leave the arm at the target location. The numbers in plot *vi* mark the times of the images in A.

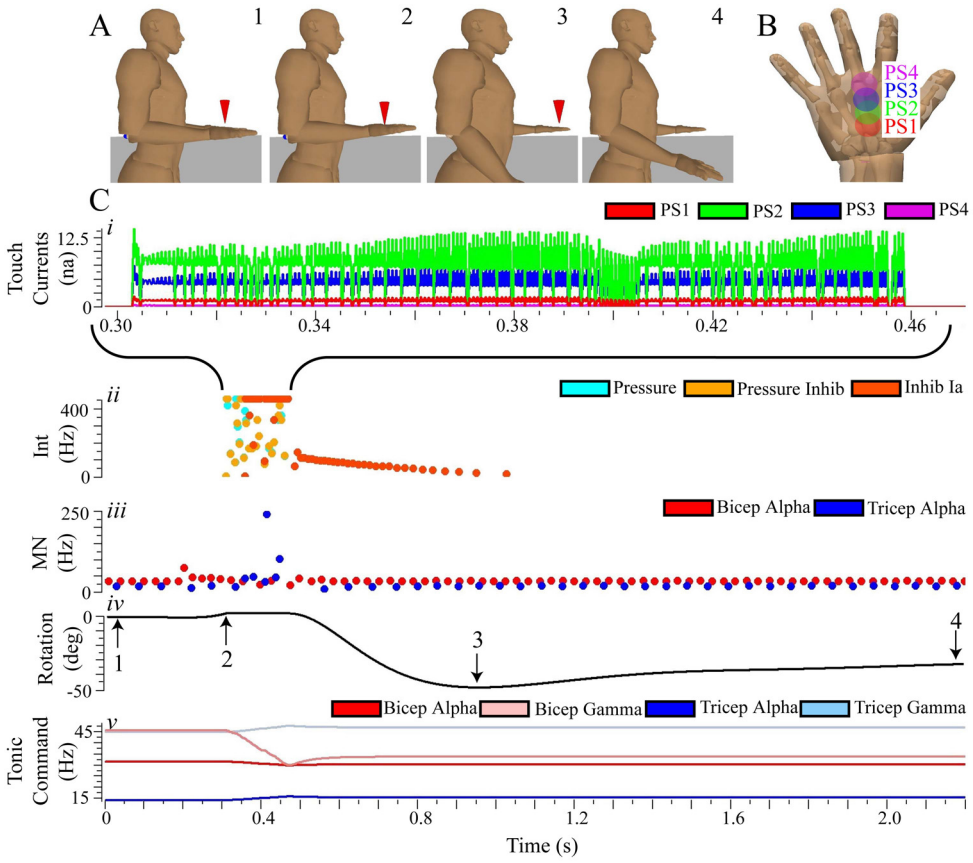

Figure 2.6. Contact withdrawal response. Withdrawal response to a sharp stimulus to the palm of the right hand. **(A)** Images from the animated model's response, showing the initial position of the right hand (1), contact upon right arm arm flexion (2), extension of the right arm (3), and the new equilibrium position of the right arm (4). **(B)** The location of the four receptive fields PS1-4 on the hand. **(C)** Each graph axis corresponds to the italicized Roman numeral labels in Fig. 2.3B. Colors are coded to match the corresponding receptive field or neuron. *i*) Generator currents in the sensory neurons during contact with the obstacle. *ii*) Firing frequency response of the sensory interneurons. *iii*) Alpha MN responses. *iv*) Rotation of the arm; numbered arrows mark the times of occurrence of the frames in A. *v*) Reflex change in the tonic command to establish a new equilibrium arm position.

'spinal' phase, and a longer-latency, persistent phase. The sensory interneurons excited arm extensor MNs and inhibited arm flexors to reverse the direction of the hand movement away from the spike (Fig. 2.6A, C). By itself, this spinal reflex was insufficient, as the arm would try to return to its new, flexed equilibrium position as soon as contact with the spike had ceased. To prevent this, the sensory interneurons also acted on the higher tonic command neurons (Fig. 2.3B) to change the position of the equilibrium point away from the spike (Fig. 2.6A, C).

Discussion

The arm flexion reflex simulations presented here demonstrate some of AnimatLab's facilities for modeling dynamic neuromechanical feedback and control. We demonstrated how appropriately scaled and timed triphasic alpha MN commands can produce rapid, smooth arm flexion movements (Fig. 2.4), and how gamma MN commands can provide a "plan" of the intended movement to enable myotactic reflexes to correct unexpected deviations from the plan produced by external perturbations (Fig. 2.5). Finally, they demonstrated how spinal contact avoidance reflexes have to be coupled with inhibition of myotactic reflexes and excitation of long-loop reflexes to allow the arm to move quickly to a new equilibrium point. These and other capabilities can be used to explore control of multijoint or multi-limb movements in humans or any skeletal animal.

AnimatLab has also been used to study the dynamic neural control of movements in a variety of animals, including cat paw shake (Klishko, et al. 2008), locust jump (Cofer, et al. 2008), and crayfish escape (Cofer, et al. 2006). In each instance, AnimatLab proved useful in helping to determine whether and how the proposed neural circuit was able to produce patterns of muscle contraction that generated movements like those displayed by the animal.

SIMM and OPENSIM are widely used general simulators for biomechanics, while NEURON and GENESIS play a similar role for neurophysiology. AnimatLab provides a general simulation tool for neuromechanics, at the intersection of biomechanics and neurophysiology, where no comparable general simulator exists. It is our hope that AnimatLab will prove as generally useful for this area as these other simulators have for theirs.

CHAPTER 3.

ROLE OF THE SEMI-LUNAR PROCESSES ON JUMP DYNAMICS IN THE LOCUST

In Preparation for submission: David Cofer, Gennady Cymbalyuk, William J. Heitler, and Donald H. Edwards

David Cofer was responsible for building and testing the locust model. The text and figures were produced by David Cofer with contributions from Dr. Edwards, and both people were also heavily involved in the editing and rewriting process. Dr. Cymbalyuk and Dr. Heitler provided revisions for the final versions of the document.

Introduction

The neural circuitry that is thought to be responsible for the locust jump has been determined by studying the electrophysiology of the locust kick with the assumption that the two behaviors use the same circuitry. Technical limitations prevent the tests necessary to confirm this assumption. This issue will be addressed in this chapter by using AnimatLab to build a neuromechanical model of the locust biomechanics and the kick neural circuit, and then testing it to see if it produces jumping behavior similar to the live animal.

There are also questions regarding the role of the SLP in the jump dynamics. It is difficult, or impossible, to directly test the role of the SLP because damage to it prevents the locust from being able to jump. Furthermore, high-speed video has shown that there is a delay in the unfurling of the SLP that may prove to be important in the timing of the application of power

during the jump. The neuromechanical simulation will allow us to examine these questions in a straightforward manner.

Materials and Methods

Animals

Adult locusts, *Shistocerca americana*, were obtained from a breeding colony at Agnes Scott College, kept caged in small groups at 37 ^{o} under a 12hr L:D cycle, and fed fresh organic lettuce and 2/1 mixture of fresh wheat germ and powdered milk. Individuals were taken from the cage to a video-recording room and placed on a jumping platform. The platform contained a heating element that could adjust the local temperature and was covered by very fine sandpaper to allow the locust a slip-free surface for jumping. A 25x30 cm yellow wooden target was placed 30 cm from the platform, and jumps to the target were induced by either gentle touches of the abdomen by a hand-held wand or by raising the temperature of the platform. Animals were retrieved after the jump and returned to the platform for another attempt. Jumps were evoked at about 5 minute intervals; individuals were returned to their cage after 10 jumps. Locust jumps were recorded at 500 fps and a resolution of 512x240 pixels by two Photron PIC R2 Fastcam video cameras with an exposure time of 0.5 ms.

Locust model

To distinguish references to the model and its parts from references to the locust, the model part names have been given the italicized names of the corresponding locust body parts, while references to the locust and its body parts are made in normal font.

The 3-D graphical model of the *locust body* was developed from a 3-D polygon mesh that was purchased online (www.turbosquid.com) and then separated into individual *body segments* using the graphics program Blender (www.blender.org). A polygon mesh is a set of vertices and triangular faces that define the volume for that segment. The dimensions of each segment were re-scaled to match published anatomical measurements (Bennet-Clark 1975, Heitler 1974). All segments were assumed to have a uniform density, and the distribution of mass throughout the mesh volume determined the moment of inertia for that segment. The model has a body length of 48 mm and a total mass of 2.5 grams, with *metathoracic femur* and *tibia* lengths of 26 mm (Bennet-Clark 1975). Individual body and limb segments were connected with either static or planar hinge joints in AnimatLab to assemble the locust body model. Angular limits on the hinge joints were set to restrict the movement of each joint to the normal range of the corresponding animal's joint. To ensure that the center of mass (COM) of the whole *locust* was located appropriately, small weighted masses were placed along the *body* axis of the *thorax* and *abdominal segments* to adjust the distribution of mass within the *body* (Bennet-Clark 1975). The COM was then determined by pinning the *body* to a hinge joint and allowing it to rotate freely. Mass was re-distributed until the *locust* balanced both vertically and horizontally at the desired location.

Biomechanics

The geometry and biomechanical properties of the femur-tibia joint of the metathoracic leg play a crucial role in the energy storage for the jump (Fig. 3.1). The *tibia extensor muscle/apodeme* is shown as a red line that attaches to the *tibia* (Fig. 3.1A), while the *tibia flexor muscle/apodeme* is shown in green. It wraps over *Heitler's lump* (Fig. 3.1H) and attaches to the

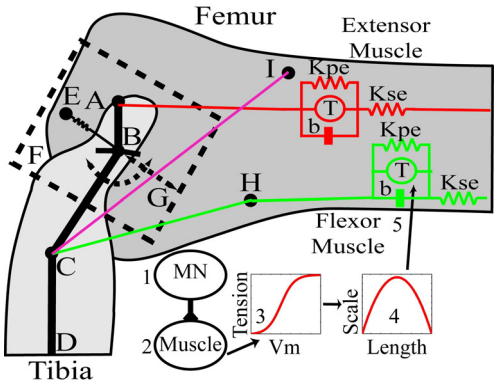

Figure 3.1. Model of the *femur-tibia* (FT) joint of the *metathoracic leg*. (A) *Extensor apodeme* attachment point on the *tibia*. **(B)** *FT hinge joint* and connection of *SLP* spring. **(C)** *Flexor apodeme attachment* point on the *tibia*. **(D)** A point further down tibia. **(E)** The *SLP* spring is attached between the *femur* and the *tibia*. **(F)** The *SLP mass* moves along the slider joint **(G)** oriented between the points B and G that is at an inclination of 36.9°. **(H)** *Heitler's lump*. The *flexor muscle* wraps over this lump to alter its orientation with respect to the *tibia* as the *leg* is moved. **(I)** The *tendon lock* is modeled as a spring located between points C and I. It is only enabled when the *tibia* is fully flexed and *flexor muscle* has a tension greater than 0.15 N (Bennet-Clark 1975, Heitler 1974). The distance between AB is 0.76 mm, BC is 1.64 mm. The angle ABC is 144°, and BCD is 143° (Heitler 1974). The muscle model is shown for the flexor and extensor muscles. This consists of a spring (Kpe) in parallel with a tension generator (T), and a dashpot (b), in series with another spring (Kse). **(1)** The muscle is activated by firing of a motor neuron (MN). **(2)** This depolarizes the muscle membrane. **(3)** Changes in the membrane voltage are converted to a tension value using a sigmoidal function. **(4)** The tension value is scaled based on the muscle length. **(5)** The scaled tension is applied to the muscle in the force generator producing a contraction.

tibia (Fig. 3.1C). The *femur* was connected to a small block of mass 1.6 mg that represented the *SLP* (Fig. 3.1F) (Bennet-Clark 1975). A sliding prismatic joint connected the *SLP* to the *femur* (Fig. 3.1G). During normal co-contraction, the distal end of the SLP (where the tibia attaches) moves 0.3 mm ventrally and 0.4 mm proximally (Burrows and Morris 2001). The slider joint was oriented to allow the *SLP* mass to move in the same direction (Fig. 3.1 B to G). A spring attached the *SLP* mass to the *femur* and was oriented along the direction of movement of the slider joint (Fig. 3.1E). The stiffness of the semi-lunar spring was calculated from a stress-strain curve obtained for the SLP (Bennet-Clark 1975). Straining the process parallel to the extensor apodeme by 0.4 mm required approximately 14.2 N of force (Bennet-Clark 1975). However, the semi-lunar process moves both proximally and ventrally, and this amount of proximal strain corresponds to 0.3 mm of ventral strain, for a total of 0.5 mm total strain. From this strain, we calculated the stiffness of the SLP as 28.4 KN/m. The *semi-lunar* spring constant was set to this value.

The *tibia* was connected to the *SLP* mass by a hinge joint that allowed the *tibia* to rotate between 5^{o} and 160^{o} (Fig. 3.1B). The *femur-tibia* hinge joint is connected to the SLP mass, so that during co-contraction and *tibial* extension, the hinge joint will move along the slider with the *SLP* mass to approximate the joint movement observed in the locust (Burrows and Morris 2001). The distances and angles that define the relationships between the *extensor* attachment, *femur-tibia* hinge joint, and *flexor* attachment were set to published measured values (AB is 0.76 mm, BC is 1.64 mm. Angle ABC is 144^{o}, and BCD is 143^{o}) (Heitler 1974).

Muscle is represented in AnimatLab by a linear Hill muscle model (Hill 1970, McMahon 1984, Shadmehr and Wise 2005a, Shadmehr and Wise 2005b). Each muscle model consists of a serial spring in series with the parallel combination of a parallel spring, a dashpot, and a force

actuator. Muscle model properties are determined by the resting muscle length, the spring and dashpot constants, the stimulus-tension curve, and the length-tension curve. The stimulus-tension curve is a sigmoidal function that relates the force level of the actuator to muscle membrane depolarization. The length-tension curve is an inverse parabola that determines the percentage of actuator force that is applied at a given muscle length.

Only two muscles for each of the rear legs are modeled in this simulation, the flexor tibiae and extensor tibiae. The maximum force that can be produced by the extensor is 15 N, which is achieved upon depolarization after a latency of 300-800 ms (Bennet-Clark 1975). The serial spring constant of the *extensor* was calculated using the Young's modulus of 18.9 $kN.mm^{-2}$ found for the extensor apodeme by Bennet-Clark (Bennet-Clark 1975). The average size of the apodeme test pieces was 3 mm long x 0.25 mm wide x 40 um thick, and so they have an area of 0.01 mm^2, and a length of 3 mm. These values allowed us to calculate the spring constant as 63 kN/m from Young's modulus using the equation K=YA/L, where Y is the modulus, A is the area, and L is the length. In the absence of published measurements that would allow calculation of the parallel spring constant, we used a value of 20 N/m because it produced a small but noticeable tension when the *extensor muscle* was stretched. The damping coefficient of the *extensor muscle* was set by hand to 700 Ns/m to produce a rise time to peak tension of approximately 400 ms. The stimulus-tension curve and the response properties of the non-spiking *neuron* that represents the muscle membrane were configured to reproduce the twitch response of the extensor muscle to a single *FETi* spike at a femur-tibia angle of 90^o (Heitler 1988). The length-tension curve was also reproduced from muscle twitch values that were taken at various femur-tibia angles using direct stimulation of the muscle (Bennet-Clark 1975). Direct stimulation of the *extensor muscle* produced twitch responses very similar to those recorded

from extensor muscle in response to a FETi spike. The resulting length-tension curve of the *extensor muscle* reached the maximum at the fully flexed position and was reduced as the *leg* extended.

Recordings from the flexor muscle showed that it produces a maximum tension of around 0.75 N in response to tetanizing stimulation, and that it reached maximum tension 35-40 ms after a latency of 15 ms (Bennet-Clark 1975). The following parameter values enabled the *flexor muscle* model to reproduce the recorded peak tension and tension time course: the serial spring constant, K_{se}, was 100 N/m, the parallel spring constant K_{pe} was 20 N/m, and the damping coefficient, b, was10 Ns/m. The stimulus-tension curve was configured to produce the desired maximum tension. As with the *extensor* model, the *flexor* length-tension curve was near its maximum value when the *leg* was fully flexed and near its minimum value when the *leg* was extended.

The tendon on the flexor muscle of the locust contains a pocket. When the tibia is fully flexed and the flexor has a tension greater than 0.15 N, this pocket is caught on Heitler's lump, which helps keep the tendon locked in place (Bennet-Clark 1975, Heitler 1974). This tendon lock property plays an important role in the jump after co-contraction when the flexor muscle and motor neurons are being inhibited. The lock helps keep the tibia fully flexed even while the flexor tension is dropping. This prevents premature extension of the tibia and initiates the jump once the flexor tension drops below a threshold value for maintaining the lock. In the model, the tendon lock was represented by a spring that connects the *flexor attachment* to a point on the *femur* (Fig. 3.1, Magenta line between C and I). The spring was disabled and produced no tension unless the *tibia* was fully flexed and the *flexor* tension was greater than 0.15 N. Once those two criteria were met the spring was enabled and it provided a force sufficient to keep the

tibia flexed. When the tension in the *flexor* dropped below the threshold the *tendon lock* was disengaged and the *tibia* rapidly extended and produced the thrust for the jump.

Neural Model

A conductance-based, integrate-and-fire neuron model was used in this simulation. Neurons were modeled as a single equipotential compartment characterized by a set of user-specifiable parameters, including membrane time-constant, size (i.e., input conductance), membrane voltage, current noise, initial spike threshold, spike-frequency accommodation, spike after-hyperpolarization conductance, and calcium conductances with activation and inactivation variables (MacGregor and Lewis 1977).

The neural network used to generate both the *kick and jump motor programs* was designed to apply the correct motor signals in a sequence and duration that mimics the motor program seen during kicking in locusts (Heitler and Burrows 1977a, Heitler and Burrows 1977b). Initial flexion of the *tibia* begins when the nine *fast flexor tibia motor neurons are stimulated to fire* (Fig. 3.2A, Green *FLTi neurons*) (Burrows 1995, Burrows 1996). These neurons synapse onto the *flexor muscle membrane* (Fig. 3.2F, Light blue FM node) causing *muscle* depolarization and *flexor muscle* contraction. The *fast extensor of the tibia motor neuron* (Fig. 3.2B, Red *FETi neuron*) synapses onto the *extensor muscle membrane* (Fig. 3.2E, Light blue EM node) causing it to contract. A central excitatory *synapse* connects the *FETi neuron* to the *fast flexor motor neurons* (B to A) (Burrows 1996, Heitler and Burrows 1977b). There are also two *inhibitory interneurons* that are involved in triggering the jump. The multimodal *'M' interneuron* (Fig. 3.2C, Gold *M neuron*) inhibits the excitatory *flexor motor neurons*, while the inhibitory *flexor inhibitor motor neuron* (Fig. 3.D Yellow *FI neuron*) *synapses* onto the *flexor*

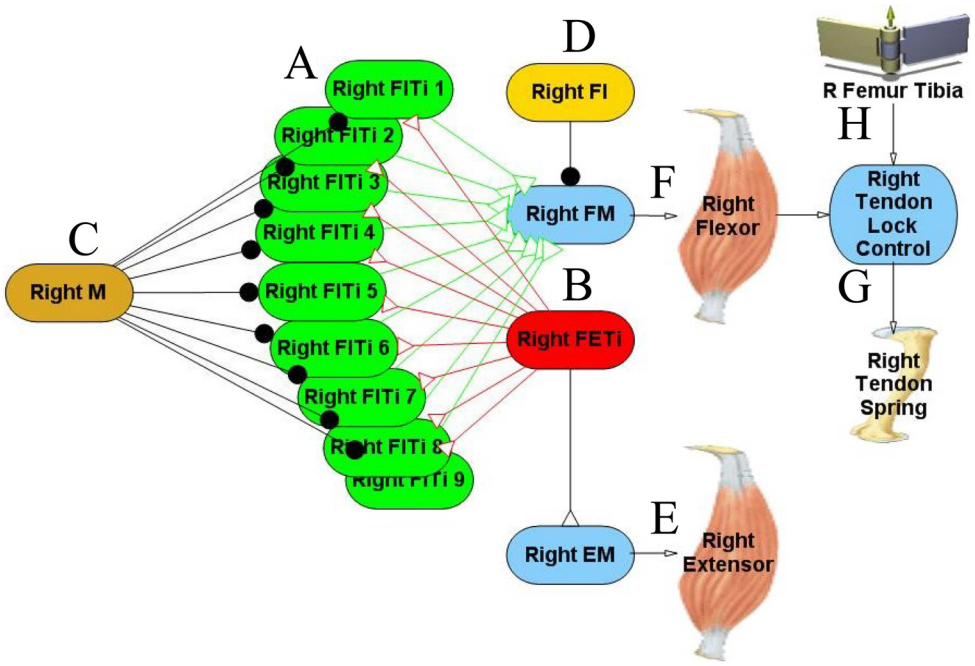

Figure 3.2. Neural network model of the *jump* motor program. Network shown is for the right leg. **(A)** Nine *fast flexor tibia motor neurons* (green *FlTi*). *FlTi*s synapse onto the *flexor muscle* membrane (Light blue *FM*). **(B)** A single *fast extensor tibia motor neuron* (red *FETi*). *FETi* synapses onto the *extensor muscle membrane* (light blue *EM*). **(C)** The *multimodal interneuron* (Gold *M*) inhibits the *FlTi*s. **(D)** The *flexor inhibitor* (yellow *FI*) directly inhibits the *flexor muscle*. **(E)** Depolarization of the *extensor muscle membrane* causes the *extensor muscle* to contract. **(F)** Depolarization of *the flexor muscle membrane* causes the *flexor muscle* to contract. **(G)** The *tendon lock* control node (Light Blue) controls when the *tendon lock spring* is enabled based on the rotation of the *tibia* and the tension in the *flexor muscle*. **(H)** When the *jump* is triggered the *femur-tibia* joint rotates rapidly to produce the *jump* or *kick*.

muscle and inhibits it directly (Burrows 1995, Pearson, et al. 1980). The *Tendon Lock control node* (Fig. 3.2G, Light blue) is responsible for enabling the tendon lock spring when the *tibia* is sufficiently flexed and the *flexor* has a tension above the lock threshold. The network that governs the right *metathoracic leg* is shown in Fig. 3.2; an identical network, also excited by the same *Flexion Command neuron*, governs the left *metathoracic leg*.

Neurons were configured to reproduce the observed firing frequencies. Peak *FETi* neuron firing ranged between 60-100 Hz (Heitler and Burrows 1977a), while the *FLTi* neurons fired around 60 Hz (Heitler and Burrows 1977a) . The central *excitatory synapse* connecting the *FETi* to the *FLTi neurons* was configured by reproducing an experiment in which the *FETi* was stimulated to fire at roughly 10 Hz while the *synaptic response* of the *FLTi* was monitored (Heitler and Burrows 1977b). The first *FETi spike* produced a 20 mV *EPSP* in all of the *FLTi motorneurons*; the *EPSPs* decayed in approximately 100 ms (Burrows 1996, Heitler and Burrows 1977b). Responses to subsequent *spikes* were reduced by *synaptic depression* in a manner similar to that observed experimentally (Heitler and Burrows 1977b).

All *neurons* had a random tonic noise of 0.3 mV added to their *membrane potentials* at each time step. The pseudo-random number generator that controlled the noise was initialized using a random seed value at the beginning of each simulation. This caused each simulation with a different seed to produce slightly different results because changes in the *neuron* voltages led to alterations in the timing of the *motor program* and the rise and fall times of the tension in each of the *muscles*. To compare the effects of differences in model parameters on performance, the same seed value of the random number generator was used to create the same initial conditions for simulations with both models.

Procedures for Simulation of Experiments

During the *kick* simulations, the *locust* was suspended above the *ground* and rotated so that its ventral surface was uppermost, and pinned in place so it could not fall. All *leg joints* except the *femur-tibia* and *tibia-tarsus joints* of the *metathoracic legs* were locked to prevent rotation. The *kick motor program* caused the *tibia* to flex initially and then kick out at high speed. This allowed us to measure the movement of the *SLP* and *tibial* rotation. *SLP* torque relative to the *extensor attachment* was calculated by recording the coordinates of the *femur-tibia joint*, *extensor attachment*, and the *SLP* force vector. These values were used to calculate the moment arm of the *SLP* force vector relative to the *extensor attachment*, and this was used to calculate the torque applied by the *SLP*. *Kick* velocity was measured as the peak velocity between the beginning of the *kick* and end of the *kick* when the *tibia* had fully rotated by 160^{o}. *Kick* duration was the time from the beginning of the *kick* till full rotation of the *tibia*.

Simulations of the locust jump began with the *locust* held 4.5 cm above the *ground*, and then dropped to the *ground*. Initially, only the *coxa-femur joints* of the *rear legs* were free to rotate. All other *joints* were locked and unmoving. After the *locust* came to rest, the other *joints* in the *rear leg* were unlocked so that the *motor program* could proceed. Once the *tibia* was fully flexed, the *metathoracic coxa-femur joint* was adjusted to fix the angle of the *leg* with respect to the *ground* at 45^{o}. The *joints* for all the other *legs* remained locked throughout the *jump motor program* in order to maintain a stable and consistent posture, and the posture of the *front legs* was adjusted to fix the initial *body pitch* of the *animal* at 3^{o}. The locks on the *joints* of the *front* and *middle legs* were disabled when the *jump* was triggered. This allowed all the *legs* to move freely throughout the take-off and ballistic phase of the jump. The *SLP* was disabled for tests by locking the *SLP* sliding prismatic joint to prevent it from moving and by disabling the *SLP*

spring to prevent it from generating tension. The maximum tension in the *extensor tibia muscle* was varied by adjusting the maximum tension that the *muscle* could generate.

Power for the *jump* was calculated using the same method outlined in Bennet-Clark (Bennet-Clark 1975). The force acting on the *body* during the *jump* impulse was multiplied as the dot product of the velocity vector acting on the *body* to obtain the power. Energy was calculated by integrating the power curve over the time period of the impulse. The same stimulus pattern that was used for *kicks* was applied to produce the motor pattern for the *jump*. *Jump* distance was measured using the position of the *locust* at the beginning of the *jump* and when either the *body* or one of the rear *legs* first touched the *ground*. The beginning of a *jump* or *kick* was always measured from when the tendon lock was disengaged. *Jump* duration was the time from the beginning till the end of the *jump*. *Jump* impulse duration was the time from the beginning of the *jump* until one of the *legs* lost contact with the *ground*. *Jump* velocity and acceleration was the peak of those values obtained during the *jump* impulse. All data analysis was performed in Matlab (Matlab R2007a, Mathworks Inc.), and statistical comparisons were made using its Anova1 one-way analysis of variance function.

The influence of *SLP* flexion torque on the *flexor muscle* was determined by comparing the tension level at which the *extensor muscle* was able to overcome the tension in the *flexor* when the *SLP spring* was intact and when it was disabled. The *extensor* tension was set to 5 N and the *tendon lock* was disabled for both tests. The first test was performed with the *SLP* intact, while in the second test the *SLP spring* was disabled. Once the *extensor* reached the 5 N tension level the *flexor* was inhibited and its tension dropped until it reached a point where the *extensor muscle* was able to overcome it and extend the *leg*.

Results

The simulated motor program and patterns of *muscle* activity responsible for the *kick* are shown in Fig. 3.3. The *kick motor program* began by stimulating the nine *FlTi* motorneurons on the right and left side to fire at about 60 Hz, which produced tension in the *flexor muscles* that caused the left and right *tibiae* to become fully flexed (Fig. 3.3A). A train of current stimuli applied to the *FETi motorneurons* on both sides began the co-contraction phase (Fig. 3.3B). Each current stimulus evoked a corresponding spike in *FETi*. In addition to driving the *extensor muscle*, the *FETi* excited the *FlTi*s on the same side (Heitler 1988) to enable the *flexor muscle* to keep the *tibia* flexed despite the mounting *extensor* tension. The kick was triggered after the *extensor* tension reached the needed level by direct stimulation of the *FI* (Fig. 3.3D) and *M* (Fig. 3.3C) *inhibitory neurons* with an applied current for 80 ms, which caused them to fire at approximately 200 Hz. The *FI neuron* directly inhibited the *flexor muscle*, causing the *flexor* tension to decline rapidly. Simultaneously, the *M neuron* inhibited the *FlTi*s to remove the drive on the flexor muscle. Rapid inhibition of the *FlTi*s and the *flexor muscle* triggered the kick by reducing the *flexor* tension below the level needed to maintain the *tendon lock* (Fig. 3.3F and G) (Heitler and Burrows 1977a, Pearson, et al. 1980). With the *flexor tendon lock* removed, the *tibia* began to extend very rapidly, completing extension in 5 ms, the same time-course that was recorded with high-speed videography (Fig. 3.3H) (Burrows and Morris 2001).

To test the hypothesis that the same neural circuitry and motor program could produce both the kick and the jump, we used the model of the kick circuit and motor program to evoke a simulated jump. The motor program was activated when the *locust* was resting normally on the *ground*. An expanded view of the data for both the kick and jump are shown in Fig. 3.4. Although the same motor program controlled both simulated behaviors, the unloaded *leg*

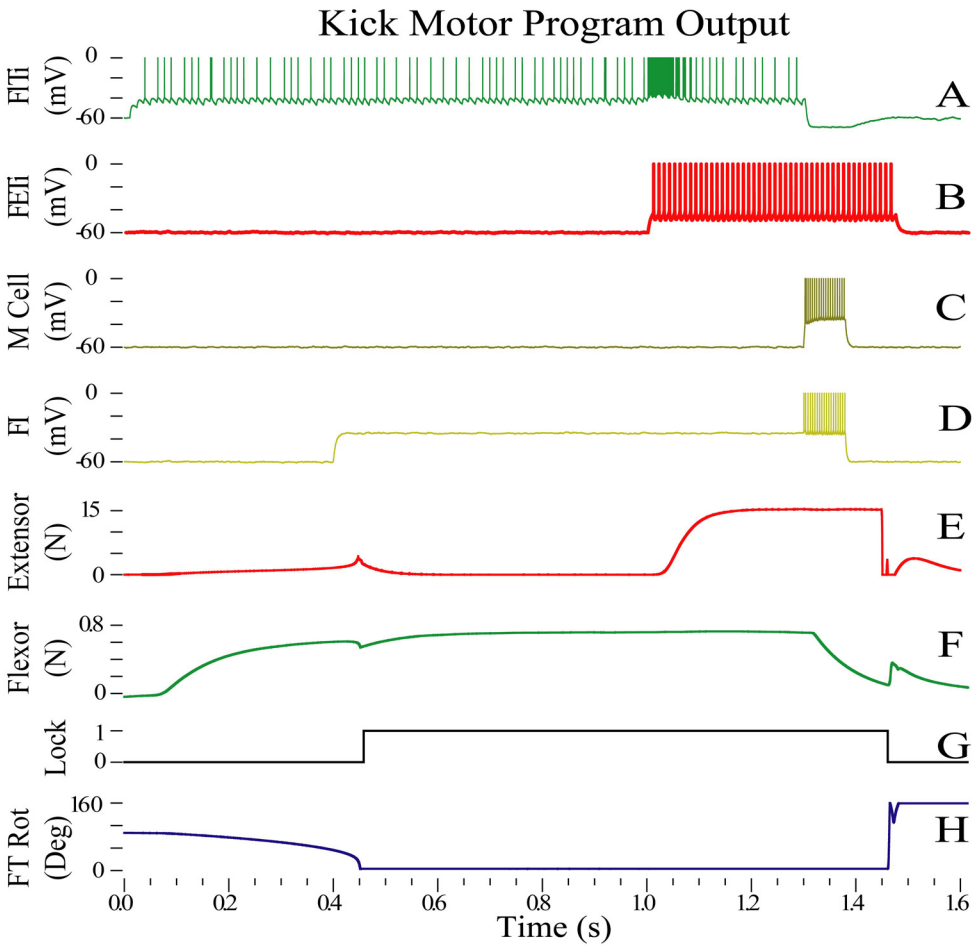

Figure 3.3. Neural output of the jump motor network. The nine flexor motor neurons were stimulated to fire **(A)** during the cocking phase and rotated the tibia into a fully flexed position. The extensor motor neuron FETi **(B)** began firing during co-contraction and the central excitatory synaptic connection from FETi to FLTi caused a brief increase in flexor frequency. The inhibitory interneurons M **(C)** and FI **(D)** then began firing once the extensor had reached the desired tension level **(E)**. This caused the tension in the flexor **(F)** to fall below the threshold of the tendon lock **(G)** and this disabled the tendon spring. This produced a rapid extension of the tibia **(H)**. Each chart corresponds to the output from a labeled element from figure 3.2.

extended in less than 5 ms to produce the *kick* (Fig. 3.4A), whereas the load imposed by the *body* caused the *leg* to extend much more gradually to produce the *jump* (Fig. 3.4B). Comparison of the simulated movements of the *locust* with a series of frames taken of a locust's jump (Shistocerca americana) by a high-speed camera operating at 500 fps (Fig. 3.5) shows that the simulation has captured the most salient features of the resulting jump behavior.

Published measures of locust (Shistocerca gregaria) jump behavior (Table 3.1) provide benchmarks with which to compare the model *locust* jump behavior. Model *locust* jumps were performed with randomly seeded noise added to the *membrane potentials* of all the model *neurons* and *muscles*, while all other model parameters were kept constant. Randomly seeded membrane potential noise ensured that the *locust* behaved slightly differently for each jump due solely to the randomness in the neurons and their affect on the biomechanics, and this provided a method to measure the variance of the *behavior* when all other initial conditions were identical. Several indicators of *jump* performance were measured, compared to the published benchmark values for live locusts (Table 3.1), and found to be essentially the same. This similarity demonstrates that the jump performance of the *locust* using the kick motor program closely matched that of the live locusts.

The contribution of the *SLP* to the *locust* jump was analyzed by comparing *jumps* made with the *SLP* intact to *jumps* made with it disabled. The *extensor* tension was varied from 7 to 15 N in steps of 2 N. For all *extensor* tension levels tested there was a significant reduction in the distance *jumped* when the *SLP* was disabled (Fig. 3.6A). The average percentage difference in *jump* distance was 37.3 ± 5 % at the maximum tension of 15 N, and this decreased to 24.8 ± 5.8 % at 7 N (Fig. 3.6B). The peak power of the *jump* impulse when the *SLP* was intact was significantly higher than when the *SLP* was disabled (2.04 ± 0.07 with, 1.28 ± 0.07 mW without,

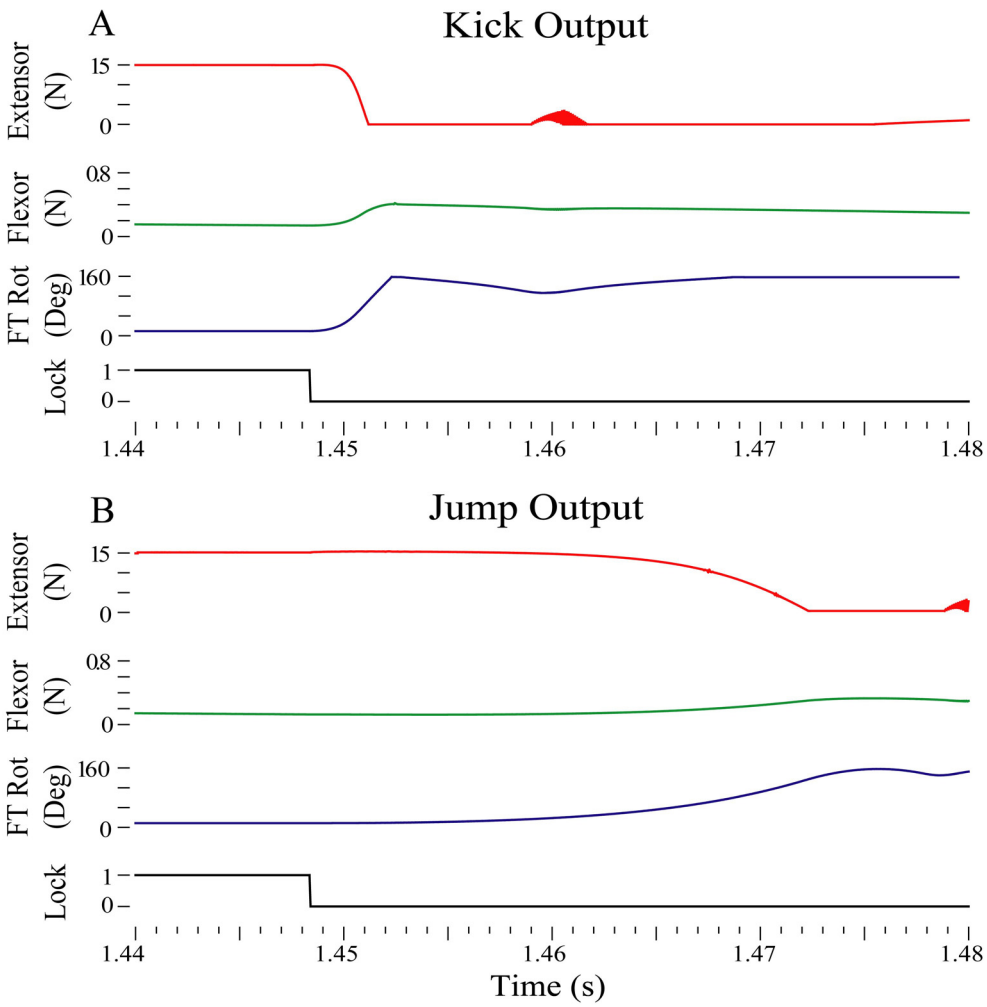

Figure 3.4. Expanded view of *jump* or *kick* data. *Kick* data from fig. 3.3 is expanded and compared to data from a *jump*. The output of the motor program was the same for both the *kick* and the *jump* so it was omitted here. Each chart shows the tension in the *extensor* and *flexor* muscle of the *left metathoracic leg*, the rotation of the *FT joint*, and the status of the *tendon lock*. **(A)** To produce a kick, the *tibia* began to rotate very rapidly after the *tendon lock* was disabled, and completed full extension in 4.1 ms. **(B)** The *jump* used the same motor program, but the *leg* rotated more slowly because the *tibia* was loaded, and so reached its maximum value after 28.35 ms.

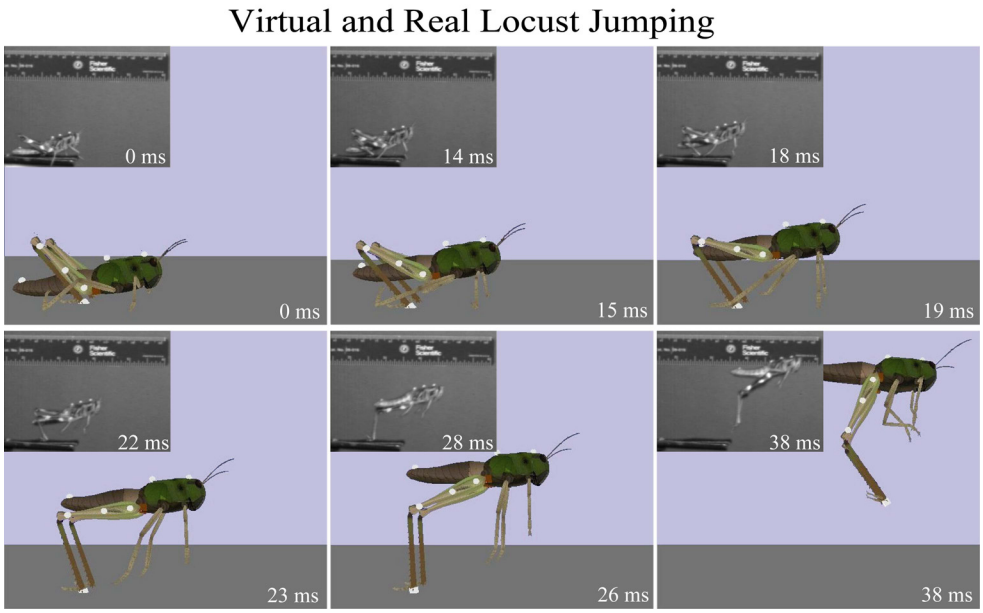

Figure 3.5. Screenshots of the simulated and real locust jumping. Images of a real locust jump are in the insets. The simulated locust produces a jump very similar to those recorded from live locusts. Live locust images are sequential frames taken using a high speed camera at 500 fps.

Table 3.1. Comparison of real and simulated locust jump performance. A number of key parameters of the virtual locust jump were compared with values published in the literature for live locusts. Simulations were performed using three different *extensor* tension values at 5, 8, and 15 N to demonstrate the wide range of the virtual locust's jump performance. The published values from live locusts were closest to the simulated results at 8 N and were within the ranges for 5 and 15 N simulations. All simulated results shown are the average and standard deviation for that measurement for 20 *jumps* at each tension level. (1) (Bennet-Clark 1975), (2) (Burrows and Morris 2001), (*) Estimate at 15 N extensor tension.

Measure	Units	5 N	8 N	15 N	Experimental
Jump Distance	(m)	0.34±0.013	0.62±0.033	1.122±0.268	0.5-0.7[1]
Jump Duration	(s)	0.33±0.027	0.458±0.041	0.587±0.142	0.31-0.43[1]
Jump Impulse Duration	(ms)	51.4±0.6	35.3±0.34	23.5±5.5	25-30[1]
Jump Velocity	(m/s)	1.8±0.117	2.51±0.054	3.6±0.07	2.2-3.2[1]
Jump Acceleration	(m/s^2)	104±6	176±6.7	290±6.6	180[1]
Peak Jump Power	(mW)	0.371±0.038	0.87±0.04	2.04±0.08	0.75[1]
Jump Energy	(mJ)	4±0.46	7.8±0.33	16±0.44	14[1*]
Kick Velocity	(deg/ms)	34.47±0.1	44.3±0.134	62.58±0.145	54.5±1.3[2]
Kick Duration	(ms)	8.41±0.018	5.96±0.02	4.1±0.01	3-6[2]

$p<10^{-30}$; Fig. 3.7A). A higher peak power also resulted in a significant increase in the total energy for the *jump* ($16 \pm 0.6*10^{-3}$ with, $9.9 \pm 0.4*10^{-3}$ mJ without, $p<10^{-30}$). In addition, when the *SLP* was disabled, the power ended significantly sooner than it did when the *SLP* was intact ($22.8 \pm 0.27*10^{-3}$ with, $22.3 \pm 0.34*10^{-3}$ ms without, $p<10^{-5}$). The percentage difference between the two *jumps* in this pair were calculated, and then the average and standard deviation of all pairs was determined (Fig. 3.7B). This showed that the average percentage difference in *jump* power remained just below 40% throughout most of the *jump* impulse. Near the end it rose quickly because the power for the *jump* without the *SLP* fell sooner than when the *SLP* was intact.

High-speed video of locust kicks have shown that the SLP does not unfurl until after the tibia has extended by more than 30^{o} degrees (Burrows and Morris 2001). To analyze this result through simulations, the locust kick was reproduced by the *locust*. Fig. 3.8A shows the amount of strain of the *SLP* plotted with the *FT joint* rotation against the time of the *jump*. The filled black squares mark the values of the *SLP* strain at 1 ms intervals as would be recorded by a high-speed video camera operating at 1000 fps, like those reported in Fig. 3.3B of Burrows and Morris (2001). Dashed line 2 is the first instance where a significant unfurling of the *SLP* would be detected at 1000 fps, and this corresponds to a *FT* rotation of 38.12^{o}. In order to understand what caused this delay it is useful to look at the *SLP* torque (Fig. 3.8B), which was generated by the *SLP* force relative to the *extensor* attachment. At the beginning of the *kick*, with the *tibia* still flexed, the *SLP* torque was negative, and acted to keep the *leg* flexed (Fig. 3.8C). Moreover, the tension in the *extensor apodeme* kept the *SLP* contracted. When the negative torque produced by the *flexor muscle* was removed by inhibition, the positive torque exerted by the *extensor muscle* began to extend the *leg*. The *extensor attachment* was pulled through the *SLP* force vector,

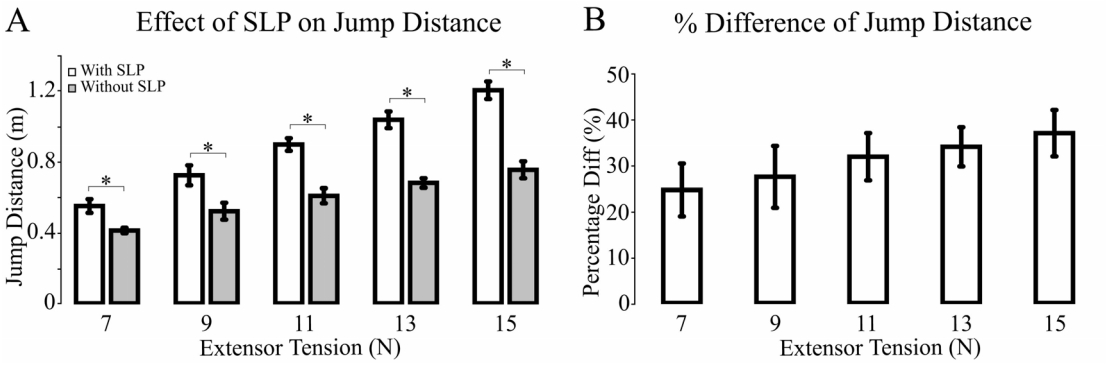

Figure 3.6. Effect of *SLP* on *jump* distance. (A) Jumps were performed with and without the *SLP* for a variety of different *extensor* tension values. At all values tested there was a significant difference between the distance *jumped* with and without the *SLP*. ($p<10^{-13}$) **(B)** The percentage difference for the maximum *extensor* tension value of 15 N was 37.3±5 percent, and this value declined slightly for decreasing values of *extensor* tension. Each test was performed with an N=20.

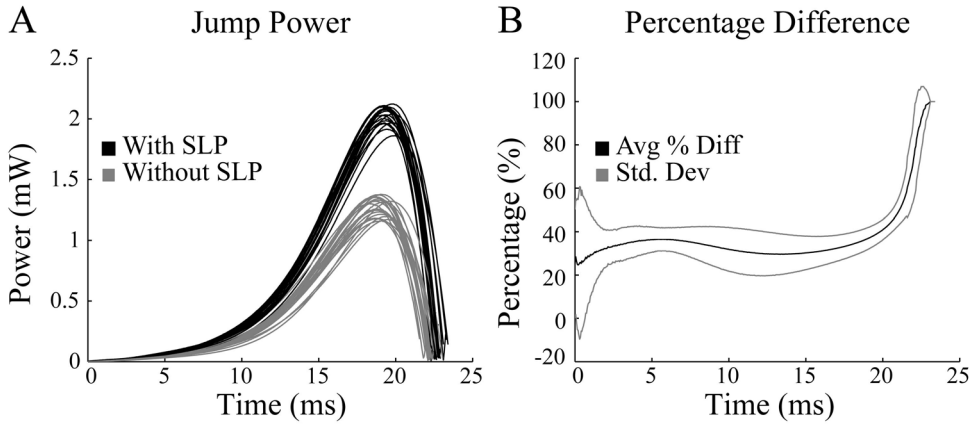

Figure 3.7. *Jump* power with and without *SLP.* *Extensor* tension is at 15 N. **(A)** *Jump* power with the *SLP* intact (black) and with the *SLP* disabled (grey). There is a significant difference in the magnitude of the peak power (2.04±0.07 with, 1.28±0.07 mW without, $p<10^{-30}$), the total energy during the *jump* impulse ($16±0.6*10^{-3}$ with, $9.9±0.4*10^{-3}$ mJ without, $p<10^{-30}$), and the duration of the impulse ($22.8±0.27*10^{-3}$ with, $22.3±0.34*10^{-3}$ ms without, $p<10^{-5}$). **(B)** The percentage difference for each pair of *jumps* with and without the *SLP* was calculated and averaged. The average percent difference between the two *jumps* remained steady near 40% for most of the duration of the *jump* impulse and then increased near the end because the power without the *SLP* dropped off more quickly than when the *SLP* was intact.

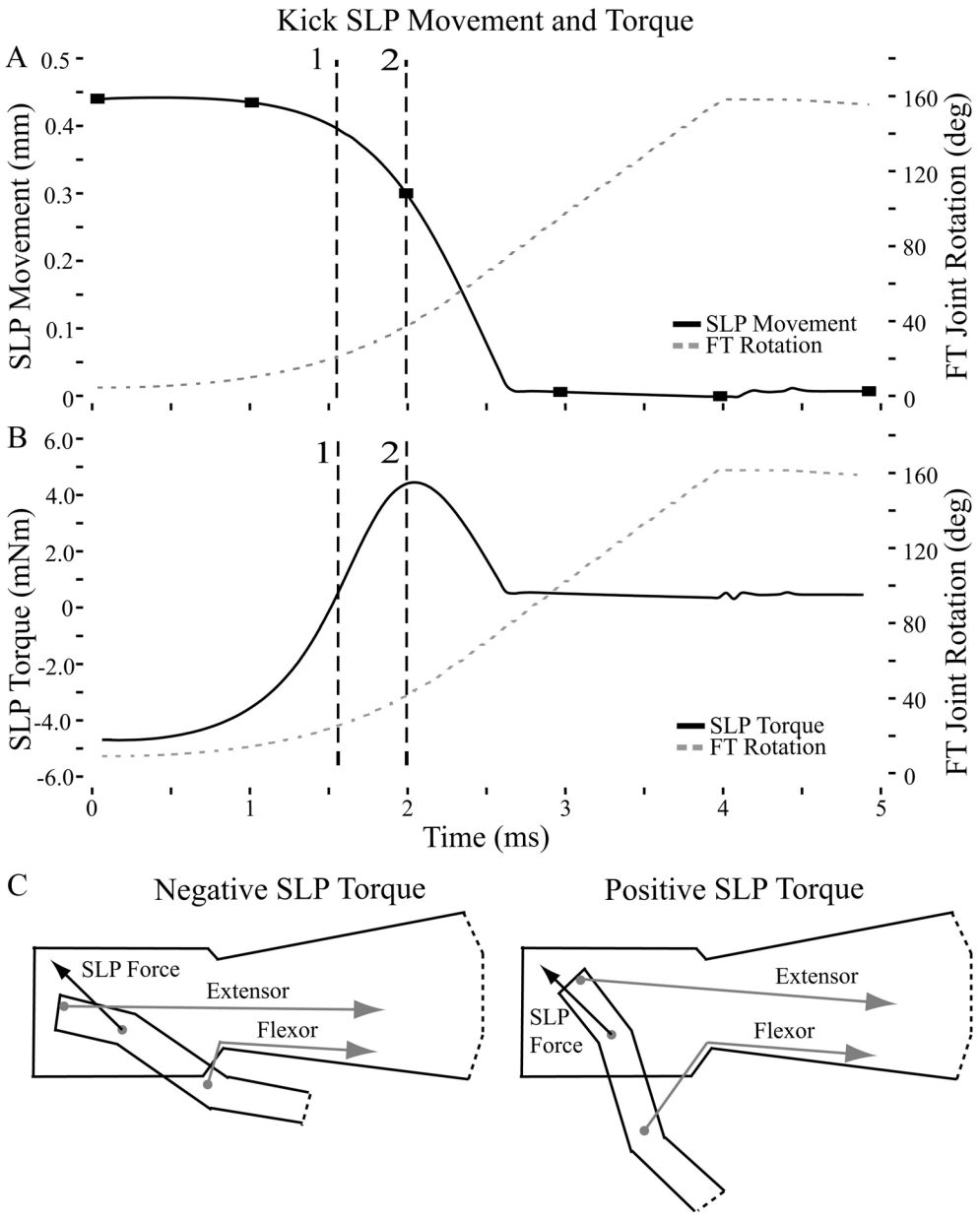

Figure 3.8. *SLP* movement and torque during a *kick*. Data was taken from the left *metathoracic leg* at an *extensor* tension of 15 N. **(A)** *SLP* movement is the black solid line with its axis on the left side, while the rotation of the *femur-tibia joint* is the grey dashed line with an axis on the right. The *FT joint* rotated with a maximum velocity of $63^{o}ms^{-1}$ and took 4 ms to reach its full extension of 160^{o}. The *SLP* began with a strain of 0.447 mm and rapidly unfurled during the *kick*. The filled black squares represent the values at 1 ms time intervals where a high speed camera at 1000 fps would take images to provide a comparison with the plot Fig. 3.3B of (Burrows and Morris 2001). The first point where a noticeable decrease in the *SLP* would be visible at 1000 fps is shown with the line **(2)**. This corresponds to a *FT* rotation of 38.12^{o}. **(B)** *SLP* torque is the black solid line with its axis on the left side, while *FT* rotation is again shown in dashed grey on the right side. *SLP* torque is the torque generated solely by the *SLP* force relative to the *extensor* attachment point. The torque started negative and this helped prevent the

SLP from unfurling. It was only after the torque became positive that significant and rapid unfurling occurred. This time is shown at dashed line **(1)**. Before this time the *SLP* unfurled less than 10%, while immediately afterwards the rest of the unfurling occurred. **(C)** Negative *SLP* torque occurred when the force applied by the *SLP* caused torques that retarded *tibia* rotation, while positive torque enhanced *tibia* rotation. Positive torque only occurred after the *leg* had rotated enough to move the *extensor* attachment point to the opposite side of the *SLP* force vector.

causing the negative *SLP* torque to decrease and change sign to become positive (Fig. 3.8C). In that position, the *SLP* torque enhanced the *femoral-tibia joint* rotation, and the *extensor* tension could no longer prevent it from unfurling. After the torque became positive, the strain in the *SLP* rapidly decreased. Prior to this time (dashed line 1), the strain had decreased by less than 10%.

The effect of the *SLP flexion* torque on the *flexor muscle* was determined by comparing the amount of *flexor* tension that was required to keep the *leg* in a fully flexed position when the *SLP spring* was intact and when it was disabled. When the *SLP spring* was intact the *flexor muscle* was able to hold the *tibia* in a fully flexed position until the tension dropped to a minimum of 0.26 N. When the *SLP spring* was disabled then flexion torque generated by the *SLP* was removed and the minimum *flexor* tension required to keep the *leg* from extending rose to a value of 0.56 N. When the *SLP spring* was present it required 54% less tension in *the flexor muscle* to keep the *leg* from extending when the *tibia lock* was not active.

Discussion

The neural control and biomechanics of locust kicking have been well described because the kick can be elicited while the locust is dissected and restrained, thus allowing simultaneous intracellular recordings from neurons in the ganglia and EMG in the muscles. Although it is currently not possible to make these recordings in a jumping locust, the similarities between the kick and jump have led to the assumption that the same motor program may produce both behaviors. A virtual neuromechanical model of the locust has allowed us to determine whether the neural circuit, motor program, and biomechanical configuration of the locust legs are sufficient to account for the jump as well as the kick. Our results demonstrate that the kick motor program is capable of reproducing the full range of jump behaviors that have been

described in the literature, and that the control of the key variable of extensor tension gives the locust the ability to alter important jump variables like the jump distance, and take-off velocity. These results strongly support the hypothesis that the kick motor program is used for both kicking and jumping.

The locust jump is an important behavior of locusts for both locomotion and escape from predators. The SLP is an evolutionary adaptation that has been thought to allow locusts to jump significantly farther than they could without it. However, because of the biomechanics of the metathoracic FT joint, direct tests of the effects of the SLP on jump performance of live locusts are difficult or impossible. The neuromechanical model of a locust presented here has allowed us to identify the contributions that the SLP makes to the jump. The simulations indicate that the SLP can act first to help keep the leg flexed until the rotation of the tibia changes the sign of the SLP torque to favor extension. They also showed how the *SLP* increased the power by almost 40% throughout most of the *jump*, and that release of most of the *SLP*'s energy occurred as the *extensor muscle* neared its peak power output.

The primary role of the flexor muscle of the metathorcic leg during jumping and kicking is to keep the tibia in a fully flexed position while co-contraction is occurring. The flexor muscle must be big enough to carry out this role, but any excess volume devoted to the flexor muscle is wasteful and that volume would be better used by the extensor muscle to provide more power for the jump. When the *SLP spring* was disabled any flexion torque that it generated was removed, and it took 54% more *flexor* tension to keep the *tibia* from extending. The flexion torque generated by the *SLP* was able to assist the *flexor muscle* and help keep the *tibia* flexed. This meant that the flexor could be less powerful, and thus more volume could be devoted to the extensor muscle.

High speed video of locust kicking suggested that this might be the case because unfurling of the SLP was delayed until after the tibia had rotated by greater than 30^{o} (Burrows and Morris 2001). The locust model helps to explain why there is a delay in the unfurling of the SLP, and how this affects the jump abilities of locust. As shown in Fig. 3.7, the delay appears to depend on the movement of the *extensor tibiae* attachment point through the *SLP* force vector, changing an *SLP* torque that produced flexion to one that promotes *leg* extension. Finally, the simulations show that the *SLP* enables the *locust* to *jump* between 25% and 40% farther than without it, depending on the force applied by the *extensor tibia*. The SLP is a specialization that has evolved to allow the locust to jump significantly further than it could with muscle power alone. It allows energy to be stored slowly, but be released very quickly, something that muscle cannot do well.

Storing energy using deformation of cuticle or strain in apodemes are common methods of overcoming the limitations of skeletal muscle that is required for arthropods to make quick movements like jumping and snapping. Some of these animals have evolved elaborate specializations to allow them to perform these rapid movements. The rabbit flea *Spilopsyllus cuniculus* stores energy for its jump in a resilin pad in the internal skeleton of its thorax. The energy is quickly released by the contraction of a small muscle that shifts the point of action of the depressor muscle, allowing the femur to be depressed (Bennet-Clark and Lucey 1967). The froghopper, *Philaenus spumarius*, stores energy for the jump by bending a bow-like cuticle formation in the pleural arch. A friction locking system prevents the legs from moving until tension in the depressors exceeds the holding force of the lock (Burrows 2003, Burrows 2006, Burrows, et al. 2008). Both the trap-jaw ant, *Odontomachus*, and the snapping shrimp, *Alpheus californiensis*, store energy for the rapid closing of their mandible or claw in the apodeme of the

muscle and by deformation of the cuticle of the exoskeleton (Gronenberg 1995a, Gronenberg 1995b, Ritzmann 1973). While all these mechanisms are similar, there is a difference that appears to be unique to the SLP mechanism of the locust. Energy from the jump primarily comes from the apodeme of the extensor and the SLP. Unlike the case with the resilin pad in the flea and pleural arches in the froghopper, the SLP and apodeme both produce forces that are on very different directions in the locust, and this is a key feature for causing the delay in unfurling the SLP.

CHAPTER 4.

CONTROL OF TUMBLING DURING LOCUST JUMPING

In Preparation for submission: David Cofer, Gennady Cymbalyuk, William J.Heitler, and Donald H. Edwards

David Cofer was responsible for building and testing the locust model. The text and figures were produced by David Cofer with contributions from Dr. Edwards, and both people were also heavily involved in the editing and rewriting process. Dr. Cymbalyuk and Dr. Heitler provided revisions for the final versions of the document. David Cofer and Lisa Blumke recorded the high speed videos of locust jumping.

Introduction

A locust has the ability to jump a large distance and land precisely on a specific target like a twig (Eriksson 1980). High speed video demonstrates that in some situations they can takeoff with a body pitch velocity that remains low throughout the jump, while in others their body pitches rapidly causing tumbling (Visual observations, and Pond, 1972). This chapter will explore mechanisms that the locust may use to control tumbling during the jump.

Physics predicts that if the thrust of the jump is directly through the center of mass (COM) of the animal then no tumbling will occur. So if the COM were located directly along the line of thrust described by the beta angle then tumbling would not be a problem. However, while the COM is very close to the coxa-femoral (CF) joint, it is not located directly at that point (Albrecht 1953, Alexander 1968, Bennet-Clark 1975). This means that as the pitch of the body

changes, the location of the COM relative to the thrust vector will change. If the COM is positioned below the thrust vector then a negative downward torque will be generated that will cause the locust to tumble downward rapidly (Fig. 4.1A), and if the COM is above the thrust then a positive upward torque will cause the body to rotate upward (Fig. 4.1B).

In fact, any error between the COM and thrust vector should be magnified during the jump impulse since downward torque will move the COM further from the thrust vector and increase the torque acting on it. One way to counter this problem is to generate a counter-torque to resist the torque generated by the thrust vector. A possible mechanism to generate the counter-torque would be to activate the muscles between the thorax and abdomen. If a negative torque is generated by the thrust vector then activating the dorsal abdominal muscle generates a positive torque that offsets it and maintains the COM near the thrust vector throughout the jump, or even pulls it up past the thrust vector (Fig. 4.1C). When the thrust is finished the downward torque is removed, but the abdominal muscles are still strongly activated to generate the counter-torque. This causes the abdomen to flex upwards (Fig. 4.1D). Activation of the ventral abdominal muscle near the end of the jump impulse would help to reduce the magnitude of the dorsal flexion and quickly bring the abdomen back into line with the body.

The locusts abdomen is used extensively during flight. Bending of the abdomen in the horizontal plane allows it to act as a type of rudder that is controlled by two separate mechanisms. The fastest system is controlled by the direction of the wind on cephalic wind-receptor hairs, while the slower system uses proprioceptive information from the cervical hairs (Camhi 1970b, Gettrup and Wilson 1964). Abdominal curling is also used during flight to increase lift when it is near stalling speeds (Camhi 1970a). If the abdomen plays such a role in

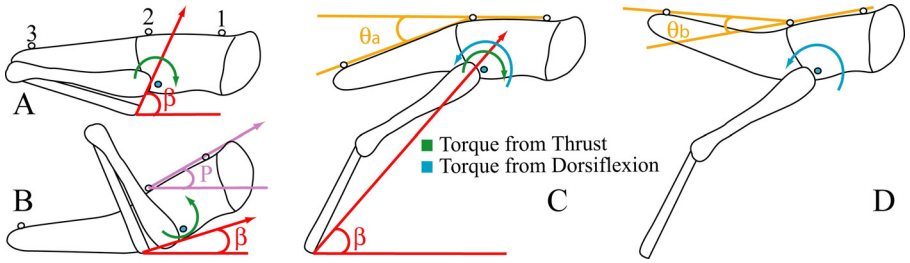

Figure 4.1. The physics of tumbling. Jump elevation is determined by the beta angle (β), which is the angle between the distal end of the tibia through the proximal end of the femur. The blue dot near the femur is the COM, and the three beads along the dorsal thorax and the abdomen were used to measure pitch and abdominal flexion. Pitch (Purple P) is determined using the first two beads. Abdominal flexion is the difference in abdominal angle 6 ms after the feet leave the ground and just before they left the ground ($\theta_{flex} = \theta_b - \theta_a$). **(A)** When the beta angle is large and the initial pitch is small the COM is below the thrust vector. This causes a downward torque and negative pitch velocities. **(B)** If the beta angle is small and the pitch is large then the COM is above the thrust vector. This causes upward torques and positive pitch velocities. **(C)** Thrust is applied throughout the jump impulse. In this example the COM is below the thrust vector so a downward torque is generated (green arrow). Activation of the dorsal abdominal muscle creates an upward counter-torque that overcomes the torque from thrust. **(D)** After the feet leave the ground the thrust is over and the torque generated from it ends. This leaves only the counter-torque from the abdominal muscle, and this causes a visible flexion of the abdomen.

the stabilization of flight, then it is not impossible to imagine that it may also play a role in stabilization of the jump.

Materials and Methods

Live Animal Analysis

Adult locusts, *Shistocerca americana*, were obtained from a breeding colony at Agnes Scott College, kept caged in small groups at 37^{o} under a 12hr L:D cycle, and fed fresh organic lettuce and 2/1 mixture of fresh wheat germ and powdered milk. Each individual was removed from their cage and had their wings clipped off near the base, and lightweight beads weighing 8.96 mg were glued onto the dorsal surface of the thorax and abdomen using superglue. One bead was placed near the head, another near the end of the thorax, and a third near the end of the abdomen. Three highly reflective 1 mm sequins were cut in half and glued onto the metathoracic leg. One was placed near the coxa-femoral (CF) joint, another at the midpoint of the femur, and a third near the femoro-tibial (FT) joint. After this treatment individuals were returned to their cage for a minimum of 4 hours before being tested.

To perform tests individuals were taken from the cage to a video-recording room and placed on a jumping platform. The platform contained a heating element that could adjust the local temperature and was covered by very fine sandpaper to allow the locust a slip free surface for jumping. A 25x30 cm yellow wooden target was placed 30 cm from the platform, and jumps to the target were induced by either gentle touches of the abdomen by a hand-held wand or by raising the temperature of the platform. Animals were retrieved after the jump and returned to the platform for another attempt. Jumps were evoked at about 5min intervals; individuals were

returned to their cage after 10 jumps. Locust jumps were recorded at 500 fps and a resolution of 512x240 pixels by two Photron PIC R2 Fastcam video cameras with an exposure time of 0.5 ms.

Four of the seven animals that were analyzed also had a BB weighing 0.33 g glued onto the pronotum near the head in order to create a downward biasing torque to extend the range of behaviors. A minimum of 6 jumps were performed with and without the weight for each animal. Two animals were jumped with the weight first and the other two with it attached last.

Only jumps that were perpendicular to the camera and where the locust did not slip were analyzed. Analysis of the jump was performed using four video frames. The first was at the beginning of the jump. The second was just before the feet left the ground, and the third was 6 ms afterwards. The fourth was the final frame where all the beads and sequins were still visible. A custom Matlab application called MarkerCollector was used to load in those four frames for all jumps and to manually select the center of each bead and sequin that was being tracked (Matlab R2007a, Mathworks Inc.). The pitch of the locust in each frame was measured using the two beads on the dorsal thorax and the angle they made with the horizontal. The initial pitch was the pitch of the locust at the beginning of the jump. The takeoff angle was determined using the sequin attached near the CF joint in two separate video frames. The first frame was just before the jump, and second frame was just before the feet left the ground. The takeoff angle was the angle the CF point in the two frames and the horizontal surface. The abdominal angle was calculated using all three beads on the back of the locust. It was the angle between the straight line formed by the two beads on the thorax with the bead attached to the end of the abdomen (Fig. 4.1 C,D). Abdominal flexion was the difference between the abdominal angle 6 ms after the feet left the ground and just before they left the ground. A dorsal flexion of the abdomen was a positive value, while a ventral extension was negative. The takeoff pitch

velocity (TOPV) was the velocity of the pitch between the beginning of the jump and when the feet left the ground. Positive velocity was upward. The tumbling pitch velocity (TUPV) was the velocity of the pitch between the time the feet left the ground and the last frame that was analyzed.

COM Location

The chosen locust was placed alive in a refrigerator at 4 C^o for 2 hours. It was weighed and then killed by placing it in a freezer at -14 C^o overnight and was reweighed in the morning. There was no detectable difference in the weight after freezing for one night. The frozen locust was suspended using a thread by melting plastacine wax with a soldering iron onto the dorsal surface of the thorax or abdomen, and then embedding the end of the thread into the hardening wax. The locust was hung from a pole and photographed with an 8 megapixel Kodak Z812IS camera. A metric ruler was visible within the image and was at the approximately the same distance from the camera as the locust. The wax attachment was moved from the tip of the pronotum to the abdomen while a picture was obtained at each new location. At each thread position the orientation of the locust changed based on the location of the COM relative to the attachment point. MarkerCollector was again used to manually select data points for each image. The tip of the pronotum was the first data point and it was used as a reference. This location was chosen because it narrowed to a well defined point and it was clearly visible in all images. The next data point was where the thread attached to the locust. The final points were taken from the ruler in order to determine scale length within the image.

The distance from the reference point to the thread attachment was measured, along with the angle made between the dorsal surface of the thorax and the vertical thread attachment point.

Thread attachment distances were measured and a linear regression was used to determine the distance from the reference that corresponded to a perfectly balanced locust when the body angle was 90^{o}. This value was used as the horizontal position of the COM. The vertical position of the COM was calculated by rotating a line at the determined distance to match the rotation of the locust body in that image, and then finding where that line crossed the vertical thread attachment line. The average of that point for all images was used as the vertical position of the COM. Any data points that were more than 1.5 times the inter-quartile range were excluded from the analysis. Once the COM measurement was completed the locust legs were removed and each component piece was measured, weighed, and photographed for use in the simulation.

Neuromechanical Simulation

To distinguish references to the model and its parts from references to the locust, the model part names have been given the italicized names of the corresponding locust body parts, while references to the locust and its body parts are made in normal font.

A neuromechanical simulation of one of the locusts was built using the software AnimatLab. The details of the *locust body* and neural control system have been previously described in chapter 3. That model was used here with only a few modifications. Specifically, the sizes, masses of the *body parts*, and *COM* were altered to match a single locust from the high speed video analysis. Each part of the *locust body* included an internal mass. The *COM* was set by pinning the *locust* in place and allowing it to rotate on a hinge joint. The densities of the internal masses were adjusted to get the *locust* to balance both horizontally and vertically at the chosen *COM*.

Hill-based muscle models were used to create *dorsal* and *ventral abdominal muscles* between the *abdomen* and *thorax* (Albrecht 1953), and the hinge joint connecting those two parts was set to allow a motion from 45^{o} to -45^{o} (Hill 1970, Shadmehr and Wise 2005a) The properties of the *muscles* were configured to be similar to the settings for the *tibia flexor muscle of the metathoracic leg* that was previously described, but they were altered slightly to allow the *muscle* to respond more quickly. The dashpot constants were set to 5 Ns/m, and the spring constants K_{se} to 20 N/m, and K_{pe} to 1 N/m. The maximum tension attainable by the *muscles* was 1 N. The length-tension curve of each was set to be a maximum at its resting length and to decrease as the *muscle* shortened, while the stimulus-tension curve was configured to facilitate quick movements.

A postural control feedback system was also used that allowed the *beta, thoracic-coxal (TC) joint* and *initial pitch* angles to be set to approximate values. The *TC joint* was set to the desired value while the other two variables used feedback to approach the value set by the user. *Beta angle* was measured for the *metathoracic legs* using the angle between the end of the *tibia*, the *CF joint* and the horizontal (Fig. 4.1A). The *beta angle* was set by rotating the *CF joint* to the correct position. The *initial pitch* was measured using two spherical beads on the dorsal surface of the *thorax* as was done for the live animals. The *pitch* was set by rotating the *CF joint* of the *prothoracic legs* to raise or lower the *body* to the desired *pitch* level. Changes in one of these variables affected the other. While the feedback system was able to get all three variables close to the desired values, the fact that two of the variables were strongly linked made it difficult for the system to reach the exact value set by the user. However, the feedback system provided sufficient control for the tests outlined below.

Simulation Jump Procedure

The simulation for the *jump* began with the *locust* held 1.5 cm above the *ground* and it was released and allowed to fall. The *metathoracic FT* joints were allowed to move freely under power of the *muscles* throughout the entire simulation. The *CF joints* of the *metathoracic legs* were elevated to 30°, similar to a live locust preparing for a kick. This allowed the *tibia* to flex fully without any interference from the *ground* substrate. Once the *tibias* were fully flexed a *tendon lock* in each *leg* was engaged that held the *tibias* in position during the co-contraction phase. The *TC joints* were then moved to the user-defined angle, while the postural control system rotated the *CF joints* of the *front* and *rear legs* to attain the user-specified *beta* and *initial pitch angles*. The *mesothoracic legs* were locked in an elevated position until just prior to the *jump* to prevent them from interfering with the postural control system. Near the time for the *jump* they were released and allowed to move freely. The other *leg joints* were locked until the beginning of the *jump*, which was defined as when the *tendon lock* system disengaged because the tension in the *flexor tibia muscle* had fallen below a threshold value (Heitler 1974). The other *joints* were then allowed to move freely throughout the jump. *Jumps* that included a dorsal flexion of the *abdomen* passed a command current to a firing rate neuron that controlled the tension in the *dorsal abdominal muscle*. The current was linear with a user defined magnitude, and start and end times.

A single jump from the high-speed video data was chosen to analyze in more detail. The jump was chosen because it was near the center of the takeoff angles that could be reproduced in the simulation, and it had a small initial pitch angle that made it less likely for the legs to slip on the *substrate*. The *beta angle* was set to match the measured takeoff angle of 29°, and the *initial pitch* was adjusted to get a value close to the measured pitch of 1.72°. The *locust* performed the

jumps while the magnitude of the current stimulus was increased to determine the affects of *abdominal* dorsal flexion on *locust* tumbling. The first *jump* occurred with no current, and it was subsequently varied between 17.5 and 19 nA.

To gain a better understanding of the tumbling behavior over a wide range of parameters the *beta angle* was varied between 20^o and 40^o in 2^o increments, while the *initial pitch* was varied between -4^o and 16^o in 2^o steps. The live animal performed jumps with takeoff angles ranging from 9^o and 38^o, and initial pitch values from -19^o to 16^o. Only a portion of this range was tested because when the *beta angle* went below 20^o the *locust feet* began to slip during the *jump*. Also, the live tests were performed on a platform that allowed the locust to bend its thorax over the open space. Tests in the simulation were done on a flat *surface*, and this made it difficult to use initial *pitches* smaller than -4^o. Low *initial pitches* also increased the chances of *foot* slippage. *Takeoff angle* was measured the same as for the live data, but used the *CF joint* position instead of the sequin location. *Takeoff and tumbling pitch velocities* were measured the same as for the live data, but the final frame used to calculate *TUPV* was always 100 ms after the *feet* left the *ground*. Any simulation where the *locust* failed to jump further than 0.2 m, or where the *feet* visibly slipped were excluded from the data sets.

A set of simulations were run using the parameters outlined above with no dorsal flexion included. Next, all simulations where the *locust* had a negative *TOPV* were re-run while a linear command current was applied to produce a dorsal flexion of the *abdomen*. The magnitude and start time of the current was varied manually and set to a value that changed the negative *TUPV* into a small positive value.

Results

COM Location

The horizontal location of the COM was determined using the regression shown in figure 4.2A. The plot shows the relationship between how far away the thread was attached from the reference point (Fig. 4.2B Red dot), and the angle that the locust made with respect to the horizontal (R^2=0.983, p=7.7e-5). The horizontal location of the COM was determined to be 15.79 ± 1.95 mm from the reference point. This was the distance of the attachment when the body angle was 90^o. The vertical location was calculated to be 9.47 ± 0.88 mm. This was determined by finding the intersection of a line at that distance with the line of the thread attachment. The projected locations of the COM for each data point are shown as green dots in figure 4.2B. There was a single outlier that was greater than 1.5 times the inter-quartile range that was excluded from the analysis (Fig. 4.2B Magenta dot). This outlier was produced when the locust was almost balanced and the two lines were nearly parallel. The projected location of the COM is shown as a light blue dot in figure 4.2B. The internal masses of the virtual *locust* were adjusted to approximate the COM calculated from the live animal. The location of the *COM* used in the simulations is shown as a light blue dot in figure 4.2C. It provides a close match to the location calculated for the live animal.

Live Locust Analysis

There was a strong positive correlation between the initial pitch and the takeoff angle for all the locusts that were analyzed (Fig. 4.3A). The correlation coefficient and significance for each locust, and for the combined data, is shown in table 4.1. Takeoff angle is determined by the posture and beta angle adopted just prior to jumping (Sutton and Burrows 2008). As the takeoff /

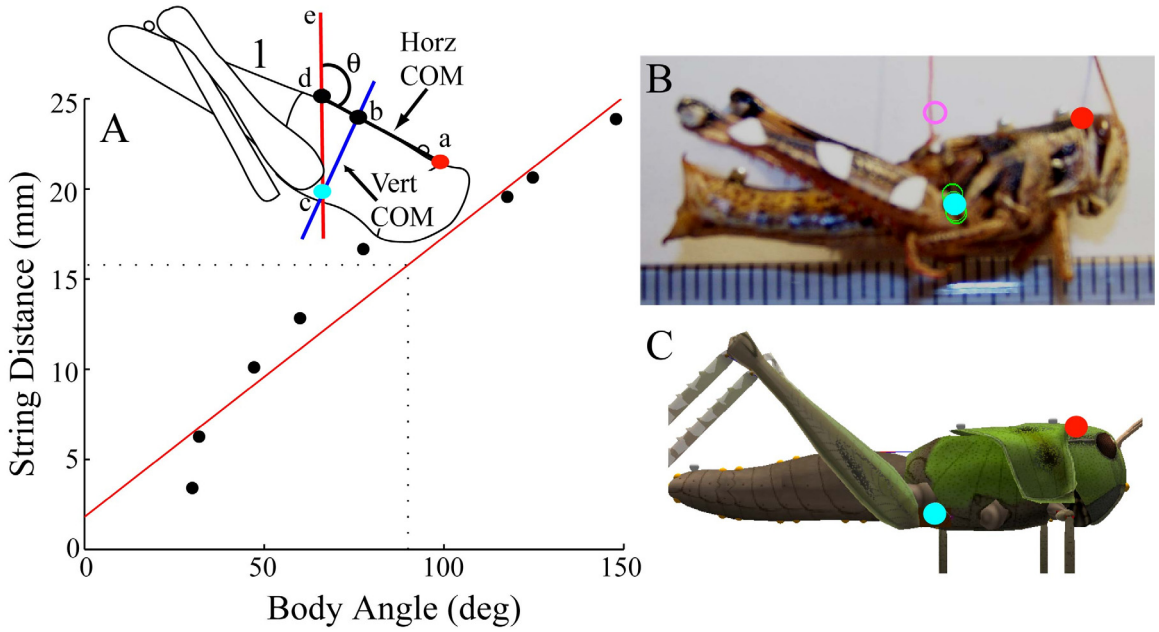

Figure 4.2. Determination of COM. (A) A thread was attached to the dorsal surface of the locust with wax. The angle of the body and the distance of the attachment point to a reference (red dot) were measured and a regression determined (R^2=0.983, p=7.7e-5). The distance to the horizontal COM was read from the regression for an angle of 90°. **(B)** Vertical COM was determined by finding the intersection of a line at the horizontal COM distance and the thread attachment. Green dots were the estimated locations for COM for each image, while the blue dot was the average location of the COM for all the images used. The magenta dot was an outlier that was excluded. The red dot was the reference point. **(C)** Image of the virtual *locust* with the position of the COM marked as a blue dot.

beta angle increased all of the locusts increased their initial pitch. Locust #2 was chosen as the model for the simulations. It showed a strong correlation between takeoff angle and initial pitch (Fig. 4.3B). In addition, each data point was color coded to show the TUPV which ranged from a minimum of 237 ^{o}/s to a maximum of 1577 ^{o}/s, with an average of 744 ± 335 ^{o}/s. The locust produced jumps over a wide range of takeoff angles and initial pitches, but the TUPV remained positive and relatively small for all them. The data point circled in red will be further analyzed through simulations (Fig. 4.3B).

A negative correlation that was primarily confined to the upper right quadrant was found that related TOPV and abdominal flexion (Fig. 4.4A). The TOPV was determined by the net torque acting on the body during the jump impulse. The greater the downward torque was, the more dorsal abdominal muscle activation was required to counter it, thus resulting in a greater flexion after the thrust was stopped. The majority of abdominal flexions were positive (114 of 126), with only a few negative ones that were of much smaller magnitude than the typical positive flexion. Negative flexion values correspond to a ventral extension.

A negative correlation was also found relating the TUPV and abdominal flexion, with the highest velocities occurring when no flexion was detected (Fig. 4.4B). Velocities decreased to zero as flexions increased. Importantly, there were very few instances of negative tumbling pitch velocities (5 of 126). Locusts overwhelmingly pitched upward during a jump with a positive TUPV regardless of their initial posture.

It was possible to recreate the results from the live animal jumps using simulations. The initial *pitch* and the current applied to the *motor neuron* of the *dorsal abdominal muscle* were controlled to reproduce the TOPV results of the live animals (Fig. 4.4C). Downward torque was greatest when the initial *pitch* was at its lowest value. When the *pitch* was lowest was when the

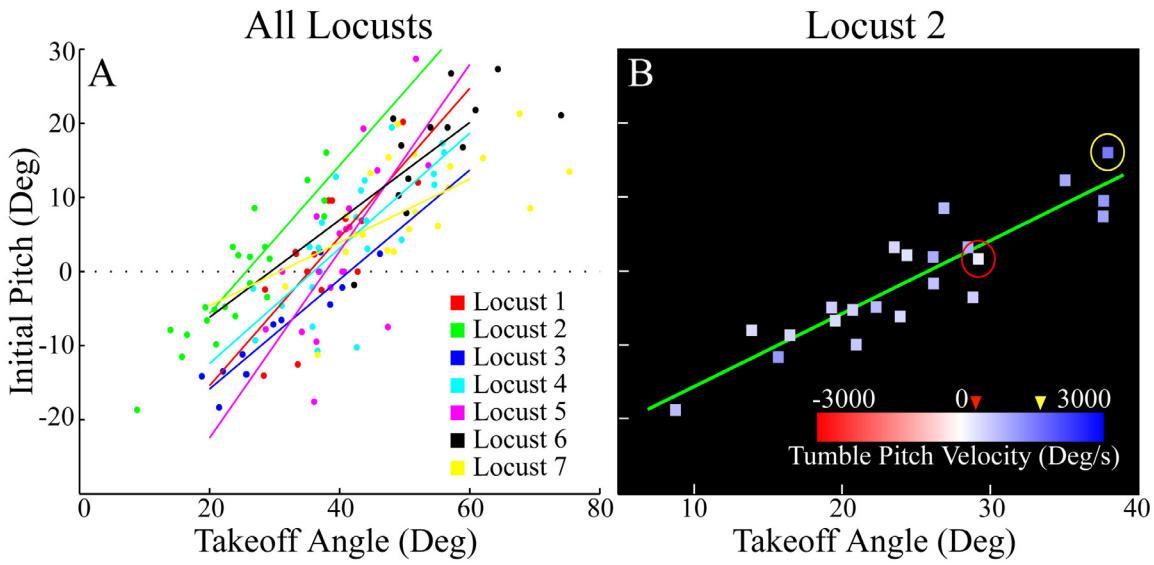

Figure 4.3. Live initial pitch vs. takeoff angle. (A) Correlation for all jumps from seven live locusts. Regression lines for each individual locust are shown in different colors, and correlation coefficients and significance are located in table 4.1. As the takeoff angle, and thus the beta angle, increased the locust also increased its initial pitch. **(B)** Locust #2 was used as the model for the simulations. This plot shows the data specific for that locust. Each data point was color coded based on the tumbling pitch velocity. The yellow and red circled points were the minimum (237 ^{o}/s) and maximum (1577 ^{o}/s) pitch velocities. All of the jumps had a positive tumbling velocity regardless of the initial pitch or takeoff angle. The data point circled in red was analyzed in more detail through simulations.

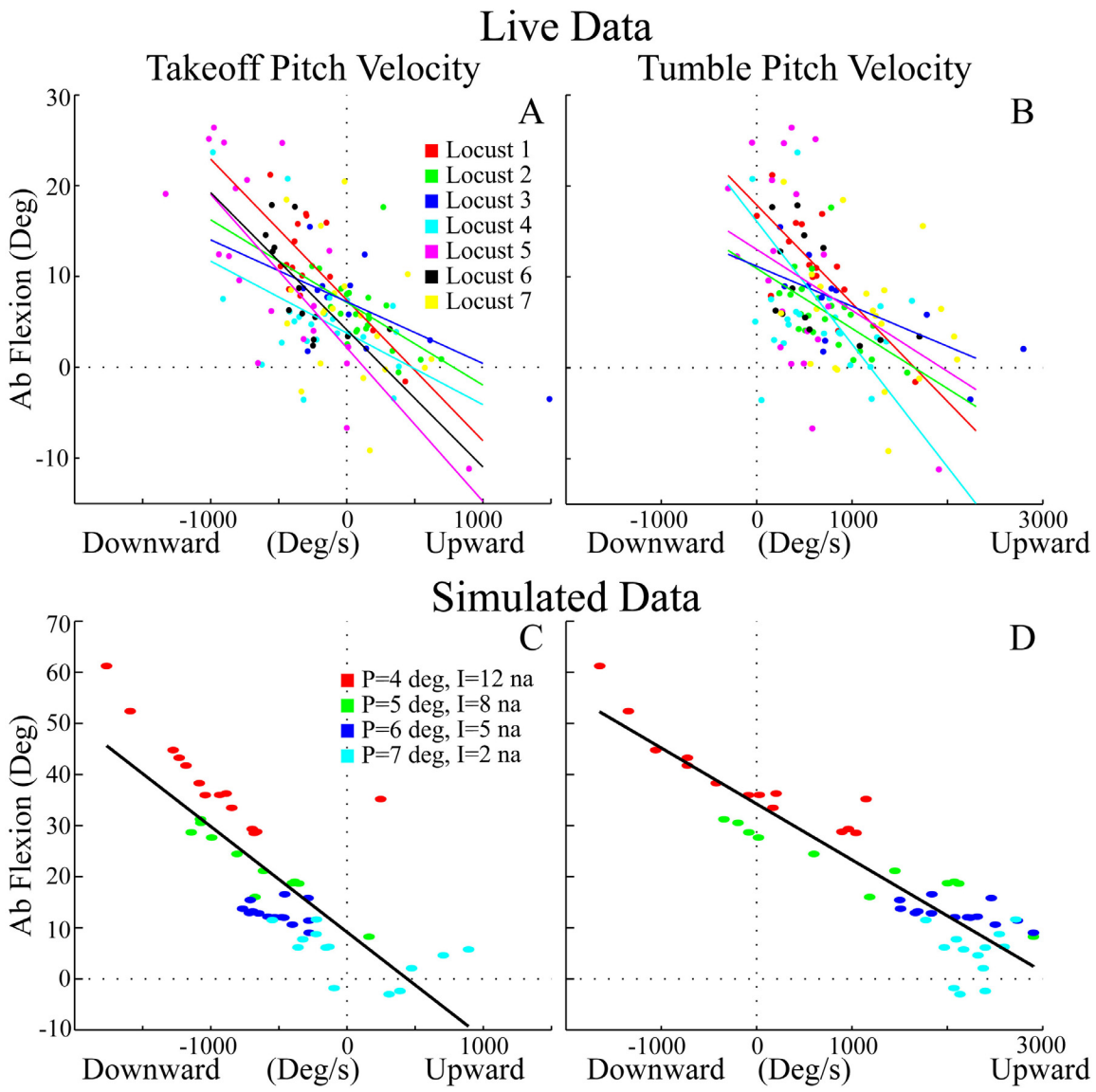

Figure 4.4. Abdominal flexion vs. pitch velocities. Data for charts A and B is for all jumps from seven live locusts. Regression lines and data points for each individual locust are shown in different colors, and correlation coefficients and significance are located in table 4.1. Only regression lines for those locusts that were significant are shown. Data for charts C and D are from simulated jumps. **(A)** There was a negative correlation between abdominal flexion and takeoff pitch velocity. When the downward torque was largest the locust generated the greatest amount of abdominal flexion. The majority of abdominal flexions were positive (114 of 126). **(B)** A negative correlation was also found between abdominal flexion and tumbling pitch velocity. Pitch velocity decreased towards zero as the amount of flexion increased. There were very few jumps that had a negative tumbling pitch velocity (5 of 126). **(C)** The current applied to the *motor neurons* of the *dorsal abdominal muscles* and the initial *pitch* of the *body* were setup to reproduce the results of live animal jumps in A. Downward torque was greatest when the initial *pitch* was at its lowest value. When the pitch was lowest was when the most current was required to overcome the downward torque, and this produced the greatest amount of

abdominal dorsal flexion. **(D)** The current and pitch were setup to reproduce the live jumps shown in chart A and the *TUPV* was measured. Adding *dorsal flexion* and controlling the *initial pitch* allowed the simulated *locust* to reproduce the results for TUPV automatically.

Table 4.1. Data for the live locusts jumps. N is the number of jumps. NW is the number of jumps without a weight attached. W is the number with a weight, and T is the total of all jumps for the locust. R^2 is the correlation coefficient for the three charts that were presented, and p is the significance. Data that was not significant at the 0.05 level was highlighted in grey, and their regression lines were excluded from charts.

	N			Initial Pitch Vs. Takeoff Angle		Ab Difference Vs. Takeoff Pitch Velocity		Ab Difference Vs. Tumble Pitch Velocity	
Animal	**NW**	**W**	**T**	**R^2**	***P***	**R^2**	***P***	**R^2**	***P***
1	0	15	15	0.598	7.2E-04	0.524	2.3E-03	0.582	9.5E-04
2	0	22	22	0.823	5.8E-09	0.261	1.5E-02	0.286	0.011
3	0	12	12	0.852	1.8E-05	0.462	1.2E-03	0.486	0.012
4	10	13	23	0.537	7.0E-05	0.228	2.1E-02	0.076	0.203
5	12	9	21	0.580	9.5E-05	0.610	4.8E-05	0.312	0.010
6	6	7	13	0.464	0.010	0.464	0.010	0.310	0.048
7	11	9	20	0.375	3.2E-03	0.120	0.124	0.084	0.204
Combined	39	87	126	0.592	6.0E-26	0.615	1.9E-14	0.479	1.4E-08

most current was required to overcome the downward torque, and this produced the greatest amount of *abdominal* dorsal flexion. Adding *dorsal flexion* and controlling the *initial pitch* allowed the simulated *locust* to reproduce the results for TUPV automatically.

Comparison of Live Locust to Simulation

A single data point for locust #2 was chosen to be analyzed in more detail through simulation (Fig. 4.3B, Red circle). The *beta angle* and *pitch* were set to the takeoff angle and pitch measured from the high-speed video, and the simulation was able to achieve results that closely matched those values (Table 4.2). Images from the high-speed video of the jump were compared with snapshots from the virtual *locust* simulations in figure 4.5. The locust began with a small initial pitch of 1.72^{o} (Fig. 4.5A1). As its feet left the ground its pitch increased by less than a degree with a TOPV of 16.4 ^{o}/s (Fig. 4.5A2), and 6 ms later a small abdominal flexion of 8.1^{o} occurred (Fig. 4.5A3). Finally, the locust pitched upwards with a small TUPV of 237 ^{o}/s (Fig. 4.5A4).

Two different simulations of this data point are also shown. The first was when no dorsal flexion was added. In this case the *locust* began with an *initial pitch* similar to the live locust (Fig. 4.5B1), but the behavior of the simulation quickly diverged from what happened with the real animal. During the takeoff phase the *body* rapidly rotated downward and attained a *TOPV* of -2260 ^{o}/s (Fig. 4.5B2). The rotation continued until the *joints* between the *body* and the *femurs* reached their limits, and then the rest of the *body* somersaulted over it (Fig. 4.5B3). A small positive *TUPV* of 280 ^{o}/s (Fig. 4.5B4) was measured because the *body* rotated down so quickly that it bounced off the limits of the *joints* and transferred enough momentum to make the *body* go back up slightly, but overall the whole *body* of the *locust* was in a negative tumble for the

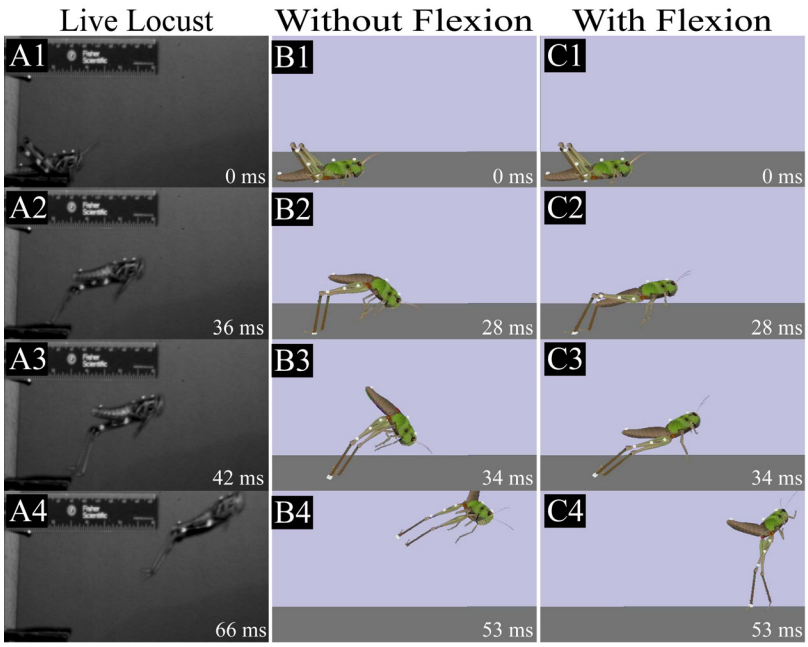

Figure 4.5. Comparison of live and simulated jumps with and without dorsal flexion. (**A1)** Locust #2 just prior to jumping. The initial pitch is small (1.72^{o}). **(A2)** Locust just before its feet left the ground. The initial pitch has remained steady throughout the jump (TOPV = 16.4 ^{o}/s). **(A3)** Locust 6 ms after feet left the ground. A small dorsal flexion of the abdomen is visible. **(A4)** Final trackable frame of the jump. The locust has pitched up slowly. **(B1)** Simulated *locust* prior to jumping. Initial conditions were set to attempt to recreate the jump of the live locust. No dorsal flexion was included. **(B2)** Unlike with the live animal the *pitch* has rapidly decreased (*TOPV* = -2260 ^{o}/s). **(B3)** The whole *body* somersaults over in a negative tumble. **(B4)** When the *joint* limits for the *CF joints* were reached the *body* and *abdomen* rebounded slightly. **(C1)** Simulated *locust* prior to jumping. Dorsal flexion was included. **(C2)** This time the *pitch* was very similar to the live locust and remained almost constant throughout the jump impulse (*TOPV* = -2.59 ^{o}/s). **(C3)** Immediately after the thrust was finished a dorsal flexion began. **(C4)** The *locust* only activated the dorsal *abdominal muscle*, so the flexion reached its maximum angle and was maintained throughout the jump. The tumbling rate was positive (*TUPV* = 365 ^{o}/s).

remainder of the *jump*. This is what was predicted by physics for an initial posture with a COM below the thrust vector. It produced a downward torque that accelerated the *COM* further from the thrust vector during the *jump* impulse, and lead to a rapid downward tumble while thrust was active.

In the second simulation a dorsal flexion command stimulus was included. The *locust* began identically to the other simulation (Fig. 4.5C1). However, the behavior during thrust application was considerably different (Fig. 4.5C2). Now, instead of a strong negative *TOPV* as seen in the other simulation it was much closer to the behavior from the live locust with a *TOPV* of -2.59 ^{o}/s. Soon after the thrust ended an *abdominal* flexion occurred (Fig. 4.5C3), and the *locust* had a positive *TUPV* of 365 ^{o}/s (Fig. 4.5C4). One difference between the behavior of this simulation and the live locust was that its abdominal flexion was limited and the abdomen quickly returned to level with the body pitch, while in the simulation the *abdominal* flexion goes to a maximum value and remains there. This was due to the fact that we were only modeling the initial contraction of the dorsal *muscle* to counteract the torque of the *jump* impulse. The live locust most likely also activated its ventral abdominal muscles near the end of the thrust to counter abdominal flexion that will occur and help bring the abdomen back into line. However, to simplify the simulation this aspect of the *jump* was not included in the model.

It is also possible to control the *takeoff and tumbling pitch velocities* by varying the amount of tension in the dorsal *abdominal muscle* during the *jump* impulse. The tension of the *muscle* was set by varying the magnitude of the command current passed into the *motor neuron*. Initially no current was used (Fig. 4.5B, Fig. 4.6 Red), and then the magnitude was varied between 17.5 and 19 nA. The simulation snapshots from figure 4.5 B and C were taken when the command current was 0 and 18.3 nA respectively. The tensions were initially similar, with only a small increase

for each increase in current (Fig. 4.6A). The point at which the extra tension was able to overcome the downward torque occurred later for each smaller current level. When there was no command current the angle between the *abdomen* and *thorax* rapidly went to its maximum value (Fig. 4.6B Red), while the *pitch* of the *thorax* also rapidly declined (Fig. 4.6C Red). As the current was increased the dorsal flexion was better able to counter the downward torque. The *abdomen* angle dipped slightly during the thrust before a dorsal flexion occurred, and when the magnitude of the current was increased the dip was entirely eliminated. By varying the current magnitude the *TOPV* could be controlled so that it varied between a highly negative -1078 ^{o}/s at 17.5 nA, and a highly positive 1337 ^{o}/s at 19 nA (Table 4.2). For most of the current magnitudes that were tested the *TUPV* was positive, but once again the exact value at 100 ms after the *feet* left the *ground* was somewhat dependent on whether momentum transfers occurred

Variation of Beta and Pitch

To explore these results in more detail the *beta angle* and *initial pitch* were systematically varied while *takeoff angle* and *pitch velocities* were measured, initially no dorsal flexion was included (Fig. 4.7 A and B). The green line in each panel of the figure is the regression line relating initial pitch and takeoff angle for the live locust the model was based upon. This line clearly splits the data into two separate components based on the *pitch velocities*. Near the line the *TOPV* was close to zero, while below it became more negative the further away the initial posture moved from the line. Moving upward from the line resulted in an increase in the *TOPV* (Fig. 4.7A). Similar results were seen for *TUPV* (Fig. 4.7B), but velocities in both directions were not as extreme.

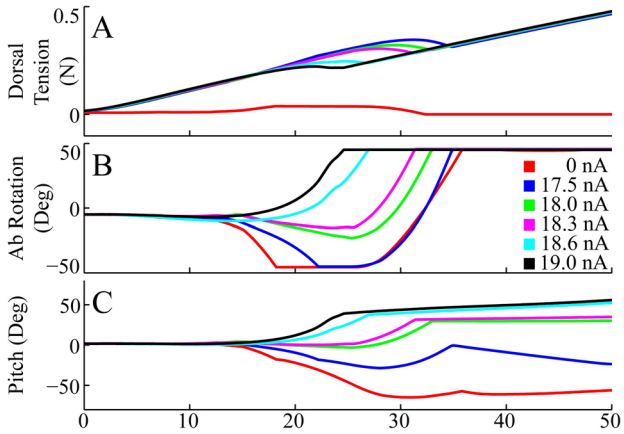

Figure 4.6. Effect of varying dorsal flexion command current. Command current was absent, or varied between -17.5 and 19 nA. Details are shown in table 4.2. **(A)** Increasing the command current led to small increases in the dorsal *abdominal muscle* tension. Increased tension was better able to counter the downward torques generated by thrust. **(B)** As the command current increased the *locust* became better at maintaining the angle between the *thorax* and *abdomen*. Eventually, the tension was able to completely overcome the downward torque and the angle increased throughout the jump. **(C)** This allowed the *pitch velocities* to be controlled. As the command current was increased the *pitch velocities* went from strongly negative to strongly positive.

Table 4.2. Comparison of live and simulated locust jump. The command current was initially off in the simulations, and was then varied from 17.5 to 19 nA. *Beta angle* was set to the takeoff angle measured from the live locust, and the *initial pitch* was adjusted to be near the value from the real animal. *Takeoff angle* and *pitch velocities* were measured for the simulations. *Takeoff angle* was large when no command current was present, and decreased as the current was increased. *Takeoff pitch velocity* varied from highly negative to highly positive, while *tumble pitch velocity* at 100 ms was somewhat influenced by momentum transfers, but tended to go from negative to positive as the command current was increased.

Jump Type	Flexion Stimulus (nA)	Beta Angle (Deg)	Initial Pitch (Deg)	Takeoff Angle (Deg)	Takeoff Pitch Velocity (Deg/s)	Tumble Pitch Velocity (Deg/s)
Live Locust	-	-	1.72	29.23	16.44	237
Simulated	0.0	29.1	1.79	49.78	-2260	280
Simulated	17.5	28.5	1.54	41	-1078	-796
Simulated	18.0	28.5	1.54	38	-158	529
Simulated	18.3	28.5	1.54	35.3	-2.59	365
Simulated	18.6	29.4	1.82	29.3	767	613
Simulated	19.0	28.5	1.54	25.2	1337	455

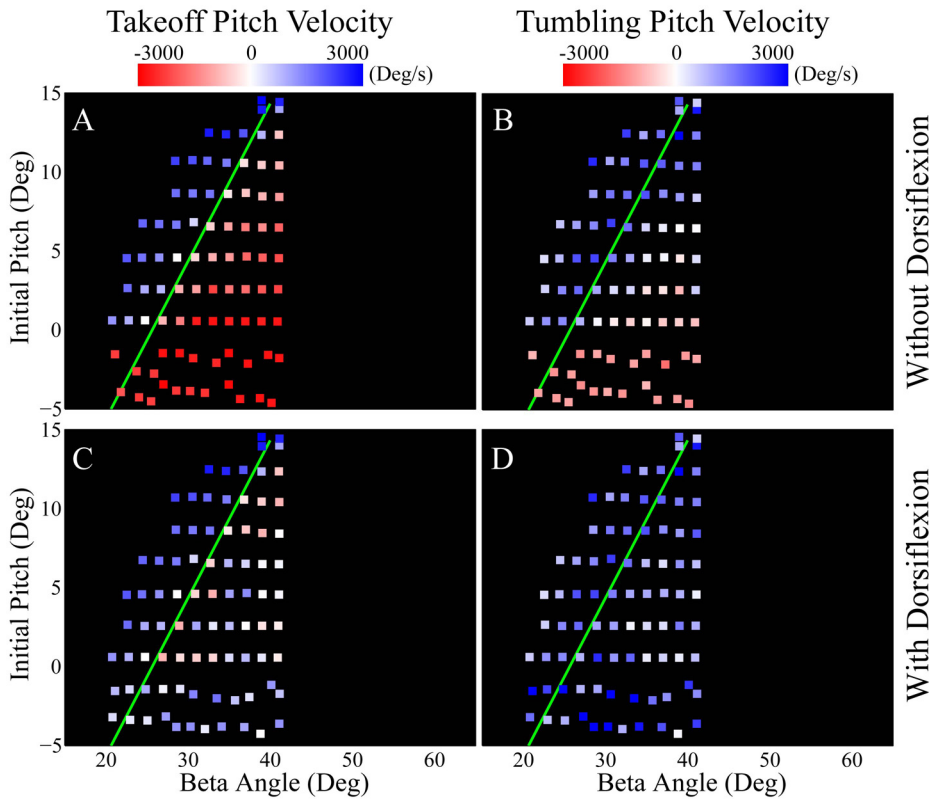

Figure 4.7. Simulated initial pitch vs. beta angle. Simulations were done both with (bottom row) and without (top row) dorsal flexion. *Takeoff* (left column) and *tumbling* (right column) *pitch velocities* are color coded. The green line is the regression of initial pitch vs. takeoff angle for locust #2. **(A)** When no dorsal flexion was included *takeoff rates* were split into two distinct sections relative to the regression line. Near the line velocities were small. The velocities increased as the points moved above the line, and decreased by moving below it. This was very different from what was seen for the live animal jumps. **(B)** Similar, but less drastic, results were seen for *tumbling pitch velocity*. This is because the bulk of the movement occurred during the thrust phase. **(C)** When dorsal flexion was included the behavior becomes very similar to the live locust results. The highly negative velocities are eliminated. **(D)** All *tumbling pitch velocities* were converted from negative to positive by using a dorsal flexion.

Next, a dorsal flexion was added to the simulation of any *jump* that had a negative *TOPV*. The addition of a dorsal flexion radically altered the behavior of the *locust* (Fig. 4.7 C, D). The *TOPV* no longer rapidly decreased as it did before. Instead, it remained near zero, or dipped slightly negative (Fig. 4.7C). The *TOPV* for all points below the regression line increased significantly from an average of -1682 ± 1083 ^{o}/s when there was no dorsal flexion to a value of -290 ± 705 ^{o}/s when dorsal flexions were used ($p<1e-15$). The tension in the *abdominal muscle* was able to counter the downward torque due to thrust and convert the *TUPV* for all *jumps* from negative to positive (Fig. 4.7D).

A comparison of figures 4.2 and 4.7 rests on the assumption that the *beta angle* and the *takeoff angle* are linearly related as previously reported (Sutton and Burrows 2008). However, when the plot was changed to use the measured *takeoff angle* instead of the *beta angle* the data was noticeably altered (Fig. 4.8A). The data related to positive *TOPV* values was changed slightly from before, but was still similar. The data related to negative *TOPV* values was drastically different. Data points that corresponded to large negative velocities were strongly skewed towards a *takeoff angle* of 60^{o}, and the closer the velocity was to zero the less the *takeoff angle* was skewed. When dorsal flexions were added to the *jumps* the data skewing was greatly reduced (Fig. 4.8B). Comparison of the *beta angle* in simulations with and without a dorsal flexion showed that it was initially the same in both cases, but once the *body pitch* began decreasing (Fig. 4.9A) the *beta angle* increased sharply to over 50^{o} (Fig. 4.9B). Dorsal flexion kept the *body pitch* more even and prevented this increase in the *beta angle*.

The influence of *pitch velocity* on *takeoff angle* is seen more clearly in a plot of *beta* versus *takeoff angle* (Fig. 4.10). *Jumps* without dorsal flexion and a positive velocity line up linearly, while those with negative velocities are highly skewed to a takeoff angle of 60^{o}

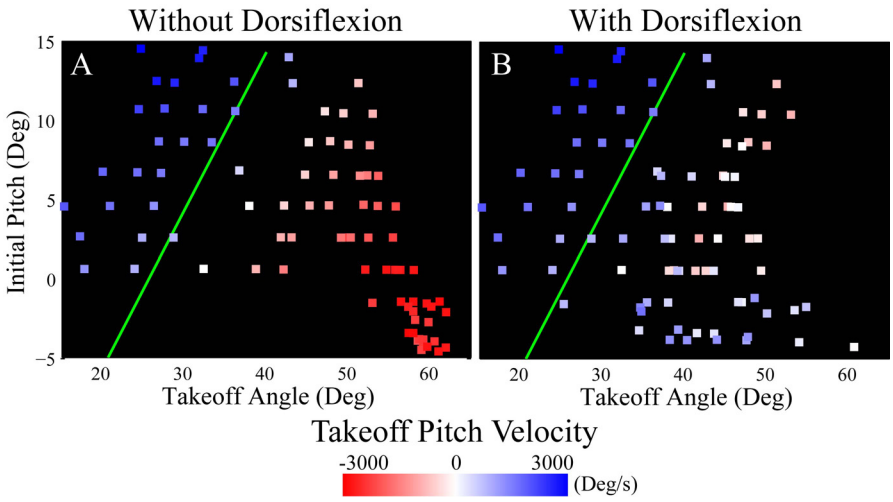

Figure 4.8. Simulated initial pitch vs. takeoff angle. Simulations were done with (right column) and without (left column) dorsal flexions. For both cases their was a small amount of shifting for jumps with positive pitch velocities, but the results were similar to what was seen when plotted against the *beta angle*. **(A)** Without dorsal flexions the jumps with negative *pitch velocities* were skewed towards *takeoff angles* of 60°. The more negative the velocity the greater the skew that resulted. **(B)** When dorsal flexions were included this skewing was greatly reduced, and the data points were pushed back towards the values seen when plotted against the beta angle.

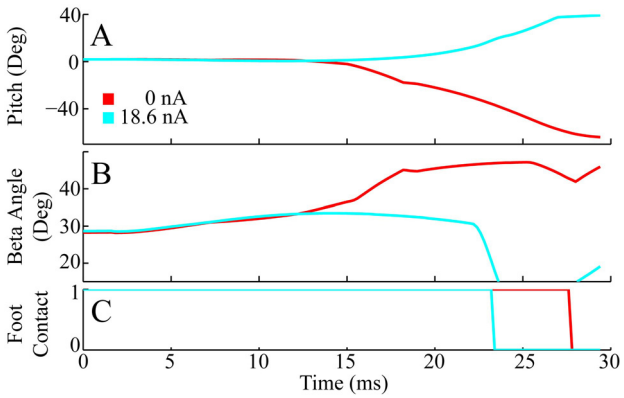

Figure 4.9. Beta angle shifts when no dorsal flexion is present. Simulations were the same as seen in fig. 4.5. **(A)** The *pitch* remained constant when a dorsal flexion was present, but decreased without one. **(B)** When the *pitch* began decreasing the *beta angle* increased to over 50^{o}, but when a dorsal flexion was included the *beta angle* remained approximately constant throughout the jump. **(C)** *Beta angle* decreased just before the *feet* left the ground in both instances.

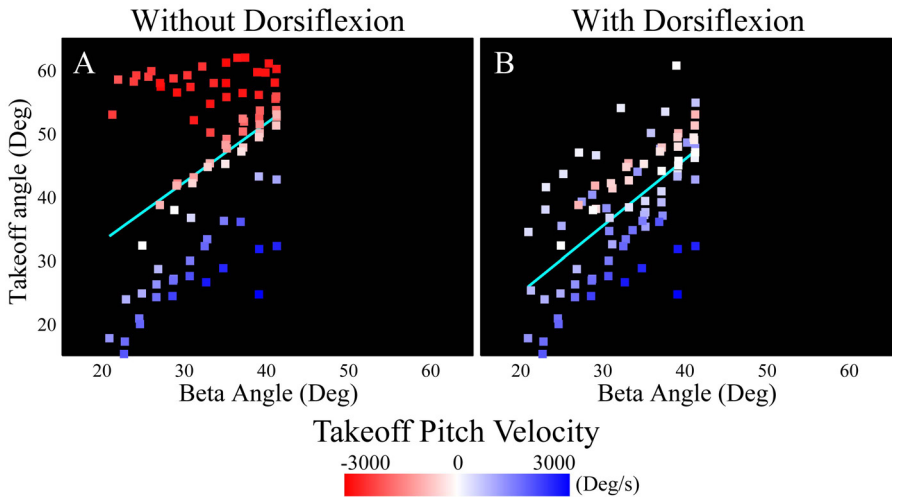

Figure 4.10. Simulated takeoff angle vs. beta angle. Simulations were performed with (right column) and without (left column) dorsal flexions. (A) Without dorsal flexions the jumps with negative velocities were skewed upwards towards 60^{o}. (B) When dorsal flexions are included this data was pushed back down and formed a linear relationship with a slope / intercept of 1.05 / 4.04^{o} ($R^2 = 0.427$). This re-established the relationship between the *beta* and *takeoff angles* seen previously (Sutton and Burrows 2008).

regardless of the initial *beta angle* (Fig. 4.10A). The correlation coefficient is low with $R^2 = 0.17$. When dorsal flexion was added the skewed points moved down to form a linear relationship with a slope / intercept of 1.05 / 4.04^o and correlation coefficient $R^2 = 0.427$ (Fig. 4.10B).

Discussion

Locusts adjusted their initial pitch relative to the beta angle prior to jumping, and simulations demonstrated that the *initial pitches* they adopted were close to the optimum value that minimized tumbling for the model *locust*. This suggests that control of tumbling is important during locust jumping, and supports the hypothesis that the locust deliberately adopts a posture that will keep tumbling velocities low. By changing the initial pitch of their bodies they can move the COM closer to the thrust vector, and thus reduce the torques acting on the body during the jump impulse.

Locusts rarely have negative tumbling pitch velocities. Regardless of the initial pitch or beta angle adopted by locust #2 it always had a positive pitch velocity, and only 5 of 126 jumps from all locusts had negative velocities. It would be difficult to explain those results if adjusting its body pitch was the only mechanism it used to control tumbling. Errors are inevitable, and the odds of placing the COM above the thrust vector are as likely as placing it under. Therefore, velocities should be evenly distributed instead of being overwhelming positive. Furthermore, physics and the simulations both predict that the when the COM is not inline with the thrust vector it will rotate away from the thrust during the jump impulse, and the rotational velocity will be related to the size of the error between the COM and thrust vector. This all supports the notion that the locust must utilize another mechanism to overcome these problems and control tumbling.

Contraction of the abdominal muscles prior to, and during the jump impulse could function as a secondary mechanism to control tumbling. Simulations where the dorsal *abdominal muscle* was contracted in this manner were able to recreate results very similar to the behavior of the real locust, and when that contraction was absent it behaved in a manner predicted by physics. The thrust torque moved the COM further away from the thrust vector and caused the thorax to rotate downward very rapidly. Varying the magnitude of the contraction also allowed the *takeoff pitch velocity* to be controlled, and produced a dip in the angle between the *abdomen* and *thorax* during the thrust. This type of dip was often noticed in the high-speed videos, and it was immediately followed by an abdominal dorsal flexion after the feet left the ground. This can be easily explained in the model. It occurred because the tension in the *dorsal muscle* was not able to counter all of the downward torque caused by the thrust. Once the thrust was finished, that torque was over and only the torque due to the *abdominal muscle* remained, and this caused the *abdominal* flexion. Further supporting this hypothesis is the fact that by adding an *abdominal flexion* it was possible to convert all of the simulated *jumps* that had negative *tumbling pitch velocities* into positive velocities. As with the live locusts, this mechanism allowed the simulated *locust* to always have a positive *TUPV* regardless of the *initial pitch* or *beta angle* that it adopted.

The type and timing of the current that was used to produce the dorsal flexion was also important. Initially, a simple step current was attempted, but this proved ineffective. Thrust for the *jump* was not applied uniformly throughout the jump impulse. The magnitude of the thrust began at a low level and increased to a peak near the end of the *jump* impulse. When a step command current was used that was strong enough to counter the peak torque of the thrust, the dorsal flexion occurred too early and always ended with a strong upward tumbling instead of the

more controlled scenario seen in live animals. The linear current used in the simulations proved to be quite effective. It increased the tension in a manner that did not overwhelm the downward torque early on, but was high enough near the end of the jump to counter the downward torque.

Timing was also important. It was necessary to start the command current before the *jump* occurred so the *muscle* would have time to build up sufficient tension to act on the *body* throughout the *jump*. This implies that just as the locust adjusts its pitch prior to jumping; it must also approximate how much dorsal flexion will be required for a given postural setup. This is further supported by the rapidness of the jump, which occurred over a span of 20-35 ms with the bulk of the acceleration occurring in the last 10-15 ms (Bennet-Clark 1975, Brown 1967). The speed of the jump makes it unlikely that a reflex circuit alone could be responsible for controlling the dorsal flexion. However, it is possible that a reflex circuit assists an already active dorsal flexion when the TOPV becomes negative and causes the dorsal abdominal muscles to stretch.

Data from the high-speed video also suggests that there is a bias in the way locusts use abdominal muscles during control of tumbling. Very few abdominal extensions (negative abdominal flexion) were seen for the live animals, and the few that were visible had a much smaller magnitude than the flexions. If locusts used both flexions and extensions then this data should have been more uniform. Simulations of ventral extensions were effective in reducing the *pitch velocities* for postures that produced a *jump* that pitched rapidly upward. So it appears that ventral extensions could be used in a manner similar to dorsal flexions, but the locust chose not to do so. Negative tumbling was corrected, but positive tumbling was left alone.

This bias may be due to an evolutionary advantage related to flight. Locusts typically initiate flight by jumping (Bicker and Pearson 1983, Pond 1972). This allows them to attain a

sufficient velocity and height above the ground to begin flight. However, if they were to begin flying after tumbling downward their head would be pointed at the ground and they would quickly crash. It's possible that they could correct their orientation in the air using their legs, body momentum, and wings to reorient the body (Arbas 1983). Unfortunately, both of these conditions could prove disastrous when attempting to escape a predator. If they were to crash into the ground they would be easily caught and killed. Even if they could correct the body orientation this would waste valuable time and delay their escape, thus increasing the likelihood they would be caught. From an evolutionary perspective it makes much more sense for the locust to pitch up instead of down. So while using the ventral abdominal muscles could potentially reduce the tumbling pitch velocities for cases where it is extremely positive, it may make it more likely that a positive tumble would be converted into being negative, and that could have potentially fatal consequences.

A key assumption in the previous work that related beta angle with the control of jump elevation was that the COM was located directly at the site of force application. The COM is so close the CF joint that this was a justifiable simplification of the model that greatly reduced the mathematical complexity. However, the COM is not actually located there, and this adds some nuance to the previous analysis. First, it means that the COM can be out of the line of the thrust vector, and if it is then this will cause torques that produce tumbling. Second, the amount of torque, and thus the velocity of tumbling, will be related to how much error there is between the location of the COM and thrust vector. By altering their pitch prior to jumping the locust can minimize this error. Third, torques from the activation of abdominal muscles can counteract the tumbling torques and change negative tumbles into positive ones. Fourth, the control of jump elevation appears to be influenced by tumbling. The simulations presented here show that when

the *pitch velocity* of the *locust* during takeoff was held close to zero the *beta angle* was a good predictor of the *takeoff angle*, and that they varied in linear fashion with a slope of one. When the velocity became strongly negative, as happens without dorsal flexions, the *beta angle* shifted during the *jump* due to shifting of the *body mass*, and this skewed the *takeoff angle* upwards. So it appears that control of tumbling is not only important for maintaining the orientation of the body during the jump, but also for ensuring that the elevation is correct.

It is unclear whether the mechanisms outlined here will be generally applicable to other jumping animals. For instance, while the bush cricket (*Pholidoptera griseoptera*) has a number of differences in its jumping mechanism compared to the locust, it shares enough similarities that one would expect it to face similar challenges regarding control of tumbling. However, recordings of their jumping failed to find examples of tumbling (Burrows and Morris 2003). This suggests that its COM is directly located at the point of thrust application. Other animals like the flee-beetle (Alticinae) appear to use their wings to prevent tumbling (Brackenbury and Wang 1995). This differs from locusts in that they typically open their wings before takeoff, while locusts open their wings 20-30 ms after the tarsi have left the ground (Pond 1972).

CHAPTER 5.

GENERAL DISCUSSION

The dynamics of sensori-motor feedback loops are affected by the linear, nonlinear, and time-varying properties of each of the many elements that compose them, including the location and transduction properties of sensory receptors, the responses of sensory neurons, the membrane properties, shapes, and synaptic interconnection of central neurons, synaptic transmission, the pattern of muscular innervation, excitation-contraction coupling of muscle, and the biomechanics of movement. Movement itself changes both the pattern of external and proprioceptive sensory input and the biomechanical context of the body's function. Both decisions about what to do next and how to do it must be continually adjusted in the face of these changes in order to respond adaptively.

To understand the operation of these feedback loops, we must first understand the function of the individual elements in isolation, and then determine how they operate in the context of the feedback loop. Finally, we must create a theoretical scheme, or model, that captures the essential properties of the elements as they work in a feedback loop in a specific neuromechanical context. Analysis of the model will help identify the elements, properties, and interactions that are most critical for it to function, as well as those to which it is indifferent, and thereby help us understand how the feedback loop in the animal works in this context.

Neuromechanical simulations have been constructed to study the behavior of humans (Cheng, et al. 2000, Delp and Loan 2000, Taga 1995a, Taga 1995b) and other animals, including insects (Cruse, et al. 1998, Ekeberg, et al. 2004, Spragna, et al. 2007), lamprey (Ekeberg and Grillner 1999), cat (Ekeberg and Pearson 2005, Lockhart and Ting 2007), and leech (Skierczynski, et al. 1996). Some of these models are more biologically realistic (Ekeberg, et al.

2004, Ekeberg and Grillner 1999, Ekeberg and Pearson 2005, Skierczynski, et al. 1996, Taga 1995a), while other models are more abstract (Cruse, et al. 1995a, Cruse, et al. 1995b, Cruse, et al. 1998, Reichler and Delcomyn 2000). These studies have shown that the simulation of the interacting neuronal and biomechanical components of the virtual animal can lead to valuable insights into the mechanisms that govern behavior. However, the modeling approaches are each tailored to a particular example of an animal's behavior and the neural structures that mediate it. At present, no general simulator for neuromechanical mechanisms exists to provide the function that NEURON (Hines and Carnevale 2001) and GENESIS (Bower and Beerman 2007) provide as general neural simulators, or that OPENSIM (Delp, et al. 2007) provides for biomechanical simulations. AnimatLab was created to provide such a general neuromechanical simulator.

AnimatLab was designed to provide a framework in which neuromechanical models can be assembled and analyzed in a specific context. AnimatLab models constitute a record of what is known or hypothesized about the individual elements, including the receptors, neurons, synapses, muscles, joints, and body segments that compose a neuromechanical system. Annotations in the 'description' cell of each 'properties table' allow the sources of all the parameter values in that table to be identified, so that what is known from measurement and what is guessed can be clearly distinguished. AnimatLab models also reflect hypotheses about how the system works and how it depends on the properties of individual elements. Simulations can test the hypotheses by showing the circumstances under which the model behaves like the animal, identifying the model parameters that have more or less influence on that behavior, and revealing patterns of activity and parameter change that are not normally observable in the freely moving animal. Moreover, the sensitivities of the model to its organization and to its parameters provide another set of testable hypotheses about the corresponding organization and parameters

of the real animal. This iterative dialog between experiment and simulation may allow the daunting complexity of dynamic neuromechanical systems to be analyzed and understood. Beyond a single investigator's efforts to build and test models of a neuromechanical system is the need to share the results of those efforts with others. The lack of a common platform for building, testing and editing neuromechanical models has meant that they are not readily shared, and so has prevented them from being tested, modified or extended by others. By providing a common, flexible and broadly applicable modeling framework, AnimatLab enables investigators to share models, test them and modify them.

Locust Jumping

The similarity between jumping and kicking in the locust has led to the assumption that both of these behaviors arise from the same neural motor program. Technical limitations currently prevent us from testing this physiologically, but neuromechanical simulation provided a method to verify that the motor program for the kick was able to reproduce jumping behavior. It demonstrated that the jump could be reproduced over a wide range of parameters, and that variables like the jump velocity could be controlled by setting the extensor tension.

Neuromechanical simulation also allowed us to do a detailed analysis of the role of the SLP in jump dynamics by comparing the jumps of virtual locusts when it was intact and when it was removed. As predicted, the SLP had a major impact on the jump distance, increasing it between 25% and 40% when the SLP was present. Analysis of the SLP during the jump also showed that there was a delay in its unfurling caused by the geometry of the joint and the torques that this produced. The geometry of the joint produces a flexion torque at the beginning of the

jump that helps assist the flexor muscle and allows that muscle to be weaker so that more of the volume in the leg can be devoted to the extensor muscle to increase the power for the jump.

These findings highlight some of the areas where neuromechanical simulation is very useful. First, simulation allows you to easily perform tests that are difficult or impossible on the living animal. This is tremendously useful because it gives the researcher the power to investigate questions that were previously out of reach.

Second, we have complete access to all the variables throughout the simulation. This makes it easy to see the relationships between components in the system and obtain a global perspective of what is happening. Typically, in physiological studies the researcher must focus on one small piece at a time, and then later try and fit all of those pieces back together like a jig-saw puzzle to produce a coherent story of how the system works. When data for all parts of the system are readily available this is no longer necessary.

Locust Tumbling

Two hypotheses on the control of tumbling in locusts were explored here. The first hypothesis was that locusts adjust the pitch of their body prior to jumping to move the COM closer to the thrust vector. This was supported by a strong correlation which showed that as the takeoff angle increases the initial pitch of the body also increased. Simulations extended this by finding that along the regression line of this correlation the pitch velocity was small, and that above the line it increasingly pitched upward, and below the line it pitched downward. The live locust was adopting a posture prior to jumping, and within the simulations that same *posture* moved the *COM* closer to the thrust vector, and thus minimized tumbling.

The second hypothesis was that contraction of the abdominal muscles during the jump creates a counter-torque that maintains the COM near the thrust vector and prevents rapid rotations of the body during the jump impulse. Comparison between an example jump from a live locust and simulated *jumps* with and without dorsal flexion stimuli demonstrated the profound effect that contraction of the *abdominal muscles* had on *jump* behavior. When no flexion was included the torque from thrust caused the *body* to rotate downward rapidly, but when dorsal *abdominal muscle* contraction was included the behavior was very similar to the live locust. Altering the amount of tension in the *muscle* also allowed the behavior to be controlled and produced tumbling over a wide range of velocities, and it was able to convert all the instances of negative tumbling into positive ones.

The simulations also produced some unexpected results. When strong negative tumbling occurred the relationship between the *beta* and *takeoff angle* was skewed upward. Further analysis showed that this occurred because the mass of the *body* shifted and caused the *beta angle* to increase during the *jump* impulse. When dorsal flexions were used to prevent negative tumbling the relationship between *beta* and *takeoff angle* was restored.

These results also demonstrated several important ways in which neuromechanical simulation is beneficial. It made it very easy to vary several parameters in a systematic manner and compare the resulting behavior. Getting live locust to vary their beta angle and initial pitch over such a wide range of values would be infeasible. Doing the same thing with the simulated *locust* was trivial. This ability allows researchers to explore the parameter space in more detail and allows them to gain a better understanding of how the real animal would behave in situations that would be difficult to replicate in an experimental setting. It also makes it easy to test which

parameters are most important to the behavior, and how sensitive they are to variation. This is something that is extremely difficult to do in living physiological preparations.

The fact that the relationship between the *beta* and *takeoff angles* was skewed for negative pitch velocities was surprising and unexpected. The neurobiology and biomechanics of behavior is such a complex and dynamic processes that it is almost impossible to keep track of all of the possible ways that variables can interact and influence each other. This is what makes relying on "just-so" stories of the causes of behavior so perilous. There may be unforeseen interactions that are difficult to grasp by just thinking about the hypothesis and the relationship between the proposed mechanism and the resulting behavior. This is where neuromechanical simulations are most useful. They make the hypothesis explicit and make these types of interactions readily apparent. This can then lead to new insights about how the system works, and generate ideas for tests of both the simulation and of physiological experiments.

General Conclusions

This work demonstrates the utility of neuromechanical simulation for analysis of the interaction of neural mechanisms of control with biomechanical processes that mediate and constrain the animal's movement. Neuromechanical simulation enables the operation of neural circuits that have been described in dissected, restrained and anesthetized animals to be understood in something like their natural context, where the consequences for movement of their activity become readily apparent.

REFERENCES

Albrecht, F. O. (1953). The anatomy of the migratory locust. London: The Athlone press.

Alexander, R. M. (1968). Animal Mechanics. Seattle: University of Washington Press.

Arbas, E. (1983). Aerial manoeuvring reflexes in flightless grasshoppers. J. exp. Biol. 107: 509-513.

Beer, R. D. (1990). Intelligence as Adaptive Behavior: An Experiment in Computational Neuroethology. San Diego, Ca: Acadmic Press, Inc.

Beer, R. D., Quinn, R. D., Chiel, H. J. and Ritzmann, R. E. (1997). Biologically Inspired Approaches to Robotics: What can we learn from instects? Communications of the ACM 40: 31-38.

Bennet-Clark, H. C. (1975). The energetics of the jump of the locust Schistocerca gregaria. J Exp Biol 63: 53-83.

Bennet-Clark, H. C. and Lucey, E. C. (1967). The jump of the flea: a study of the energetics and a model of the mechanism. J Exp Biol 47: 59-67.

Bicker, G. and Pearson, K. (1983). Initiation of flight by an identified wind sensitive neurone (TCG) in the locust. J Exp Biol 104: 289-293.

Bower, J. and Beerman, D. (2007). Constructing realistic neural simulations with GENESIS. Methods Mol Biol 401: 103-125.

Brackenbury, J. and Wang, R. (1995). Ballistics and visual targeting in flea-beetles (Alticinae). J Exp Biol 198: 1931-42.

Brown, R. H. J. (1967). The mechanism of locust jumping. Nature 214: 939.

Burrows, M. (1995). Motor patterns during kicking movements in the locust. J Comp Physiol [A] 176: 289-305.

Burrows, M. (1996). The Neurobiology of an Insect Brain. Oxford: Oxford University Press.

Burrows, M. (2003). Biomechanics: froghopper insects leap to new heights. Nature 424: 509.

Burrows, M. (2006). Morphology and action of the hind leg joints controlling jumping in froghopper insects. J Exp Biol 209: 4622-37.

Burrows, M. and Morris, G. (2001). The kinematics and neural control of high-speed kicking movements in the locust. J Exp Biol 204: 3471-81.

Burrows, M. and Morris, O. (2003). Jumping and kicking in bush crickets. J Exp Biol 206: 1035-49.

Burrows, M., Shaw, S. R. and Sutton, G. P. (2008). Resilin and chitinous cuticle form a composite structure for energy storage in jumping by froghopper insects. BMC Biol 6: 41.

Cameron, J., Skofronick, J. and Grant, R. (1999). Physics of the Body. Madison, WI: Medical Physics Publishing.

Camhi, J. M. (1970a). Sensory control of abdominal posture in flying locusts. J Exp Biol 52: 533-537.

Camhi, J. M. (1970b). Yaw-correcting postural changes in locust. J Exp Biol 52: 519-531.

Cheng, E. J., Brown, I. E. and Loeb, G. E. (2000). Virtual muscle: a computational approach to understanding the effects of muscle properties on motor control. J Neurosci Methods 101: 117-30.

Chiel, H. J. and Beer, R. D. (1997). The brain has a body: adaptive behavior emerges from interactions of nervous system, body and environment. Trends Neurosci 20: 553-7.

Clauser, C. R., McConville, J. T. and Young, J. W. (1969). Weight, Volume, and Center of Mass of Segments of the Human Body. Aerospace Medical Research Laboratory document AMRL-TR-69-70.

Cofer, D., Cymbalyuk, G., Heitler, W. and Edwards, D. (2008). Biomechanical analysis of locust jumping in a physically realistic virtual environment. Abstr Amer Physical Soc Meeting Prog. No. Y36.00002.

Cofer, D., Reid, J., Zhu, Y., Cymbalyuk, G., Heitler, W. and Edwards, D. (2006). Biomechanical simulation of the escape response in crayfish and locust. Neuroscience Meeting Planner, Online program No. 836.9/KK10.

Cruse, H., Bartling, C., Cymbalyuk, G., Dean, J. and Dreifert, M. (1995a). A modular artificial neural net for controlling a six-legged walking system. Biol Cybern 72: 421-30.

Cruse, H., Brunn, D., Bartling, C., Dean, J., Dreifert, M., Kindermann, T. and Schmitz, J. (1995b). Walking: a complex behavior controlled by simple networks. Adaptive Behavior 3: 385-418.

Cruse, H., Kindermann, T., Schumm, M., Dean, J. and Schmitz, J. (1998). Walknet—a biologically inspired network to control six-legged walking. Neural Networks 11: 1435–1447.

Davoodi, R. and Loeb, G. (2002). A software tool for faster development of complex models of musculoskeletal system and sensorimotor controllers in Simulink. - Journal of Applied Biomechanics. 357-365.

Davoodi, R., Urata, C., Todorov, E. and Loeb, G. (2004). Development of clinician-friendly software for musculoskeletal modeling and control. Proceedings of the 26th Annual International Conference of the IEEE EMBS.

Delp, S. and Loan, P. (2000). A computational framework for simulating and analyzing human and animal movement. IEEE Comput in Sci and Eng September/October. 46-55.

Delp, S. L., Anderson, F. C., Arnold, A. S., Loan, P., Habib, A., John, C. T., Guendelman, E. and Thelen, D. G. (2007). OpenSim: open-source software to create and analyze dynamic simulations of movement. IEEE Trans Biomed Eng 54: 1940-50.

Di Prisco, G. V., Wallen, P. and Grillner, S. (1990). Synaptic effects of intraspinal stretch receptor neurons mediating movement-related feedback during locomotion. Brain Res 530: 161-6.

Duysens, J. and Pearson, K. G. (1980). Inhibition of flexor burst generation by loading ankle extensor muscles in walking cats. Brain Res 187: 321-32.

Ekeberg, O. (1993). A Combined neuronal and mechanical model of fish swimming. Biol Cybern 69: 363-374.

Ekeberg, O., Blumel, M. and Buschges, A. (2004). Dynamic simulation of insect walking. Arthropod Structure & Development 33: 287–300.

Ekeberg, O. and Grillner, S. (1999). Simulations of neuromuscular control in lamprey swimming. Philos Trans R Soc Lond B Biol Sci 354: 895-902.

Ekeberg, O. and Pearson, K. (2005). Computer simulation of stepping in the hind legs of the cat: an examination of mechanisms regulating the stance-to-swing transition. J Neurophysiol 94: 4256-68.

Eriksson, E. (1980). Movement Parallax and distance perception in the grasshopper (Phaulacridium Vittatum). J Exp Biol 86: 337-340.

Gettrup, E. and Wilson, D. M. (1964). The Lift-Control Reaction of Flying Locusts. J Exp Biol 41: 183-90.

Grillner, S., Deliagina, T., Ekeberg, O., el Manira, A., Hill, R. H., Lansner, A., Orlovsky, G. N. and Wallen, P. (1995). Neural networks that co-ordinate locomotion and body orientation in lamprey. Trends Neurosci 18: 270-9.

Grillner, S., Parker, D. and el Manira, A. (1998). Vertebrate locomotion--a lamprey perspective. Ann N Y Acad Sci 860: 1-18.

Grillner, S. and Rossignol, S. (1978). On the initiation of the swing phase of locomotion in chronic spinal cats. Brain Res 146: 269-77.

Grillner, S. and Wallen, P. (2002). Cellular bases of a vertebrate locomotor system-steering, intersegmental and segmental co-ordination and sensory control. Brain Res Rev 40: 92-106.

Gronenberg, W. (1995a). The fast mandible strike in the trap-jaw ant *Odontomachus* I. Temporal properties and morphological characteristics. J comp physiol A 176: 391-398.

Gronenberg, W. (1995b). The fast mandible strike in the trap-jaw ant Odontomachus II. Motor control. J comp physiol A 176: 399-408.

Heitler, W. (1974). The locust jump. Specialisations of the metathoracic femoral-tibia joint J Comp Physiol 89: 93-104.

Heitler, W. (1977). The Locust Jump III: Structural specializations of the metathoracic tibiae. J Exp Biol 67: 29-36.

Heitler, W. (1988). The role of fast extensor motor activity in the locust kick reconsidered. J Exp Biol 136: 289-309.

Heitler, W. J. and Burrows, M. (1977a). The locust jump. I. The motor programme. J Exp Biol 66: 203-19.

Heitler, W. J. and Burrows, M. (1977b). The locust jump. II. Neural circuits of the motor programme. J Exp Biol 66: 221-41.

Hiebert, G. W., Whelan, P. J., Prochazka, A. and Pearson, K. G. (1996). Contribution of hind limb flexor muscle afferents to the timing of phase transitions in the cat step cycle. J Neurophysiol 75: 1126-37.

Hill, A. V. (1970). First and last experiments in muscle mechanics. Cambridge: Cambridge University Press.

Hines, M. L. and Carnevale, N. T. (2001). NEURON: a tool for neuroscientists. Neuroscientist 7: 123-35.

Klishko, A., Cofer, D., Edwards, D. and Prilutsky, B. (2008). Extremely high paw acceleration during paw shake in the cat: A mechanism revealed by computer simulations. Abstr Amer Physical Soc Meeting A38.00007.

Lockhart, D. B. and Ting, L. H. (2007). Optimal sensorimotor transformations for balance. Nat Neurosci 10: 1329-36.

MacGregor, R. and Lewis, E. (1977). Neural Modeling: Electrical Signal Processing in the Nervous System. New York: Plenum Press.

Massone, L. E. and Myers, J. D. (1996). The role of plant properties in arm trajectory formation: a neural network study. IEEE Trans Syst Man Cybern B Cybern 26: 719-32.

McMahon, T. A. (1984). Muscles, reflexes, and locomotion. New Jersey: Princeton University Press.

Myers, J. D. and Massone, L. L. (1997). The role of the plant properties in point-to-point arm movements: a robustness study. Biol Cybern 76: 173-80.

Pearson, K., Ekeberg, O. and Buschges, A. (2006). Assessing sensory function in locomotor systems using neuro-mechanical simulations. Trends Neurosci 29: 625-31.

Pearson, K. and Gordon, J. (2000). Spinal Reflexes. In: Principles of Neural Science (Kandel, E. R., Schwartz, A. B. and Jessell, T. M., eds.), pp. 713-736. New York: McGraw-Hill.

Pearson, K. G., Heitler, W. J. and Steeves, J. D. (1980). Triggering of locust jump by multimodal inhibitory interneurons. J Neurophysiol 43: 257-78.

Pond, C. (1972). The initiation of flight in unrestrained locust Schistocerca gregaria. J Comp Physiol 80: 163-178.

Reichler, J. and Delcomyn, F. (2000). Dynamics simulation and controller interfacing for legged robots. Int'l Robotics Res 19: 41-57.

Ridderstrom, C. (2003). Legged Locomotion: Balance, control and tools - from equation to action. PhD Thesis, Department of Machine Design. - Royal Institute of Technology.

Ritzmann, R. (1973). Snapping Behavior of the Shrimp Alpheus californiensis. Science 181: 459-460.

Santer, R. D., Yamawaki, Y., Rind, F. C. and Simmons, P. J. (2005). Motor activity and trajectory control during escape jumping in the locust Locusta migratoria. J Comp Physiol A Neuroethol Sens Neural Behav Physiol 191: 965-75.

Shadmehr, R. and Wise, S. (2005a). Computational Neurobiology of Reaching and Pointing: A Foundation for Motor Learning. Cambridge, MA: MIT Press.

Shadmehr, R. and Wise, S. (2005b). A simple muscle model. Supplement to (Shadmehr and Wise, 2005a). Online at http://www.shadmehrlab.org/book/musclemodel.pdf.

Skierczynski, B. A., Wilson, R. J., Kristan, W. B., Jr. and Skalak, R. (1996). A model of the hydrostatic skeleton of the leech. J Theor Biol 181: 329-42.

Sobel, E. C. (1990). The locust's use of motion parallax to measure distance. J Comp Physiol 167: 579-588.

Spragna, J., Goldman, D., Lin, P., Koditschek, D. and Full, R. (2007). Distributed mechanical feedback in arthropods and robots simplifies control of rapid running on challenging terrain. Bioinspir Biomim 2: 9-18.

Sutton, G. P. and Burrows, M. (2008). The mechanics of elevation control in locust jumping. J Comp Physiol A Neuroethol Sens Neural Behav Physiol 194: 557-563.

Taga, G. (1995a). A model of the neuro-musculo-skeletal system for human locomotion. I. Emergence of basic gait. Biol Cybern 73: 97-111.

Taga, G. (1995b). A model of the neuro-musculo-skeletal system for human locomotion. II Real-time adaptability under various constraints. Biol Cybern 73: 113-21.

Uuml, Schges, A., Schmitz, J., Auml and Ssler, U. (1995). Rhythmic patterns in the thoracic nerve cord of the stick insect induced by pilocarpine. J Exp Biol 198: 435-56.

Whelan, P. J., Hiebert, G. W. and Pearson, K. G. (1995). Stimulation of the group I extensor afferents prolongs the stance phase in walking cats. Exp Brain Res 103: 20-30.

Windhorst, U. (2007). Muscle proprioceptive feedback and spinal networks. Brain Res Bull 73: 155-202.