Computational Motor Control: Muscle

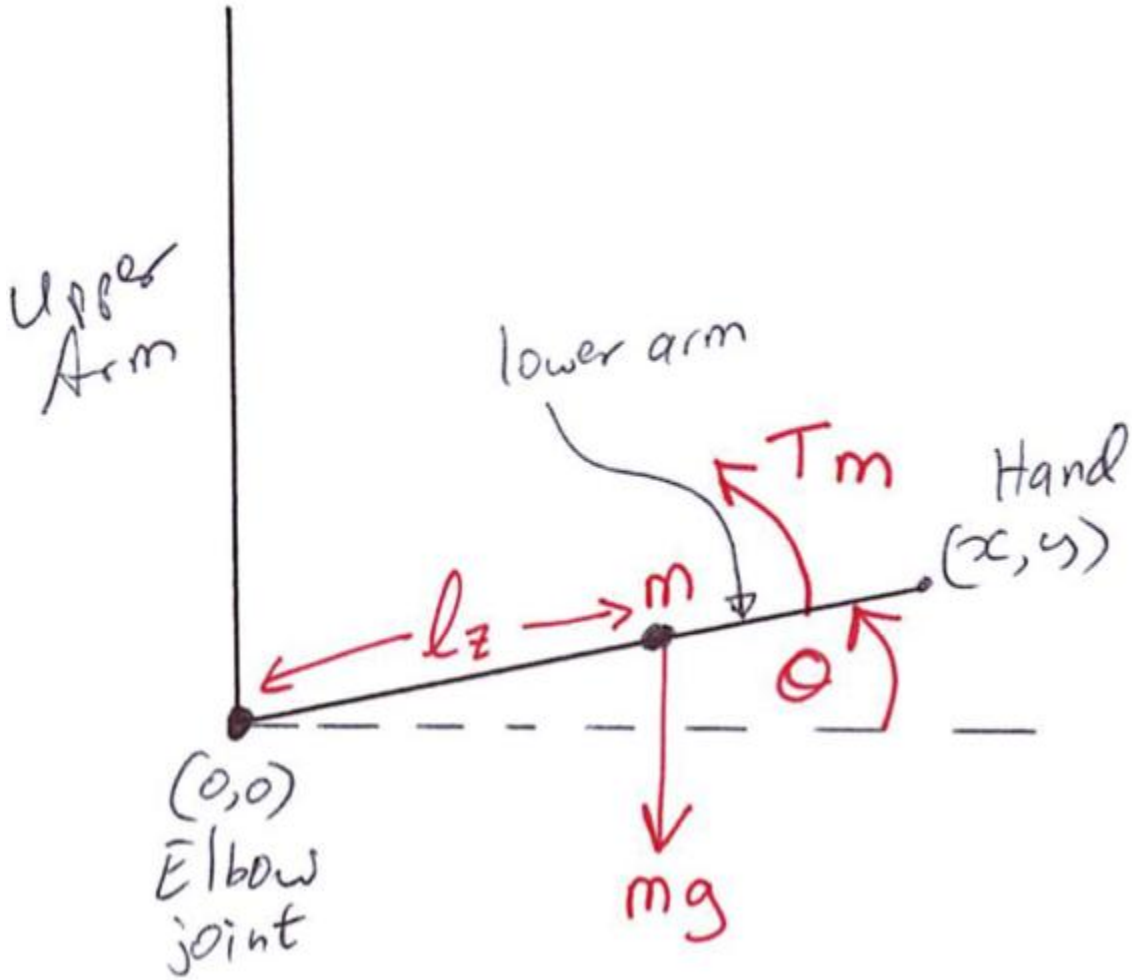
Introduction

Here we will add muscles with neural efferent inputs and afferent feedback, to our arm model. We will start with very simplified models of muscle force and work up to descriptions of more physiologically realistic models.

Modelling simple torque muscles

Constant Torque Muscle

In the previous section we built a model of the skeletal dynamics a forearm. As a first step to implement "torque muscles", we will use a simple dynamic equation of motion of a single joint arm in a vertical plane, with gravity:



Schematic of a simple one-joint arm in a vertical plane with one muscle.

$I\ddot{\theta} = mglz\cos\theta + Tm$

where m is the mass of the segment, g is the gravitational constant (9.81 m/s/s), lz is the distance of the centre of mass from the joint, and θ and $\ddot{\theta}$ are joint angular position and acceleration, respectively. The new term we are now adding, Tm, is the net joint torque due to muscle contraction.

Here is what a simple Python function would look like for simulating these dynamics:

```
def onejointmuscle(state,t,Tm):
  m = 1.65    # kg
  g = -9.81   # m/s/s
  l = 0.179   # metres
  I = 0.0779  # kg m**2
  a = state[0]
  ad = state[1]
  add = (m*g*lz*cos(a) + Tm) / I
  return [ad,add]
```

You can download a full program here that implements a simulation and animation of this version of the arm:

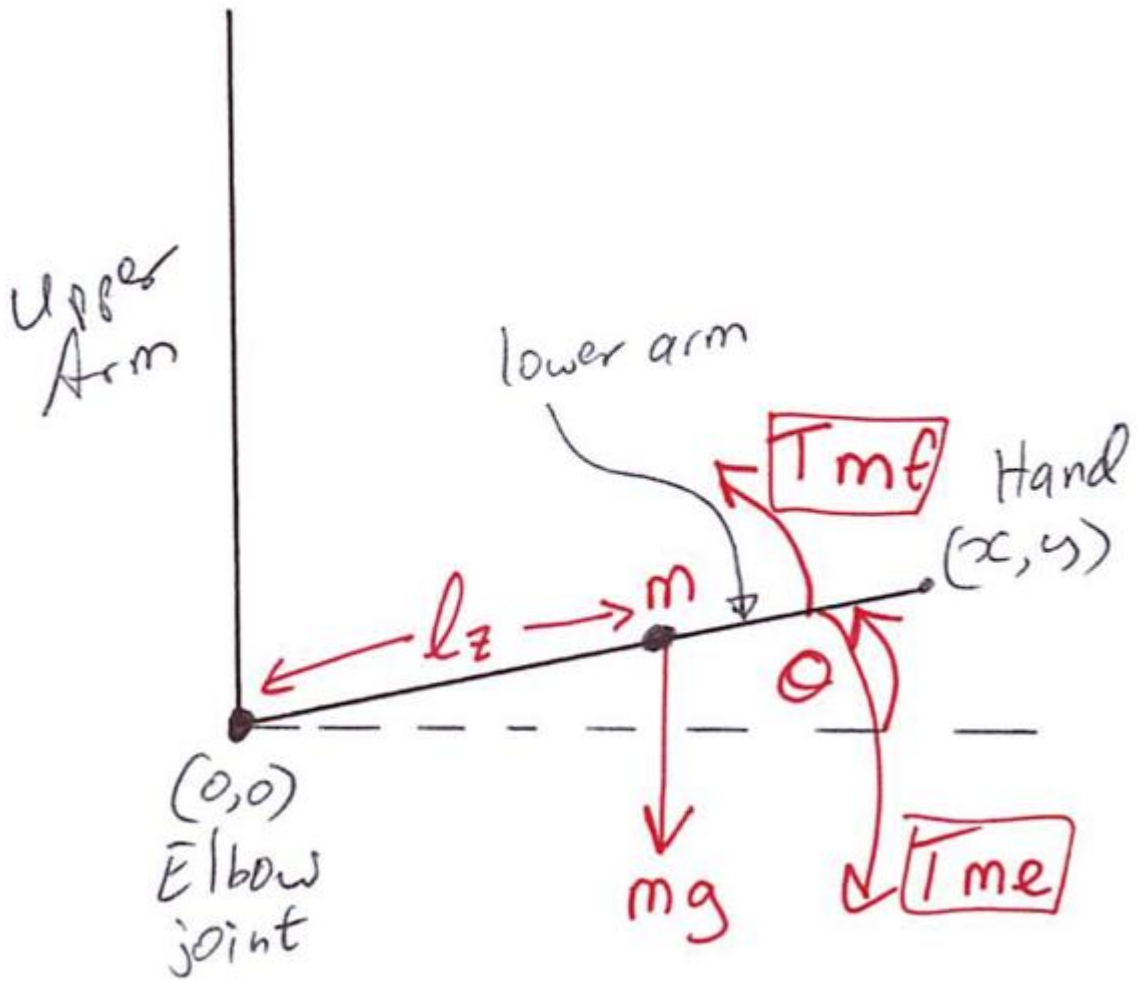
onejointmuscle_1.py

Questions

1. Play with the Tm parameter until you arrive at a value that maintains the arm at an angle of 30 degrees.
2. What value of muscle torque Tm is required to maintain the arm at 45 degrees?

Two Antagonistic Torque Muscles

Now we add a second "muscle" so we have Tmf from a "flexor" muscle (like biceps brachii) and Tme from an "extensor" muscle (like triceps brachii).



Schematic of a simple one-joint arm in a vertical plane with two antagonistic muscles.

Now our dynamics equation looks like this:

$I\ddot{\theta} = mglz\cos\theta + Tmf - Tme$

Note how the two muscles act to produce torque in opposite directions.

Here is a new version of the Python function:

```
def onejointmuscle(state,t,Tmf,Tme):
  m = 1.65    # kg
  g = -9.81   # m/s/s
  lz = 0.179  # metres
  I = 0.0779  # kg m**2
```

```
a = state[0]
ad = state[1]
add = (m*g*lz*cos(a) + Tmf - Tme) / I
return [ad,add]
```

You can download a full program here that implements a simulation and animation of this version of the arm:

onejointmuscle_2.py

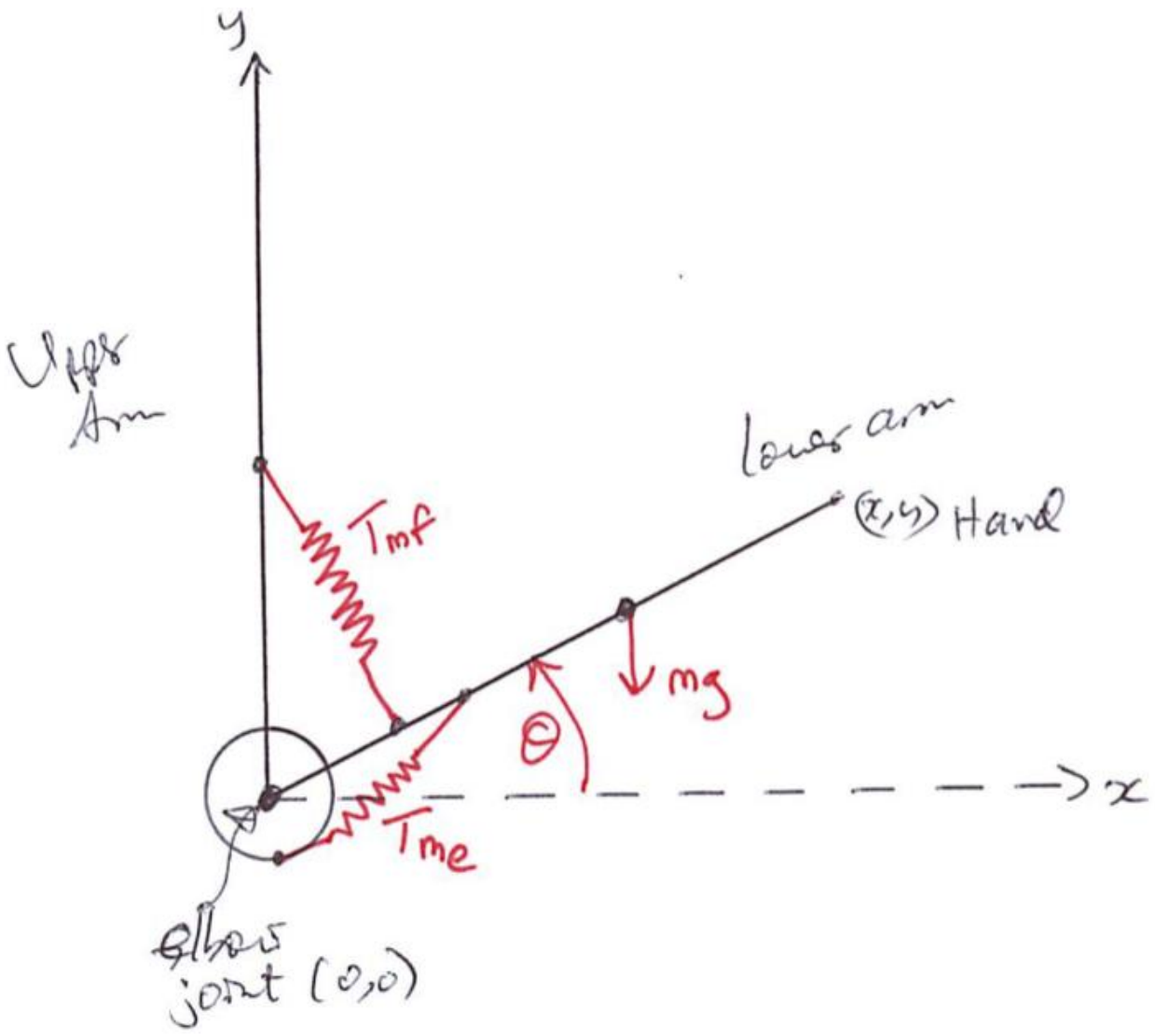
Questions

1. Say the initial position of the forearm is $\pi/4$ radians with zero initial velocity. What static muscle torques Tmf, Tme are needed to make the arm stay put?
2. Using your answer from Question 1 as a starting point, what happens if you increase Tmf and Tme by the same amount (e.g. by 1.0 Nm)?

Muscles as Linear Springs

A very coarse approximation of muscle behaviour is that the muscles deliver a torque that depends linearly on joint angle. This is essentially the same as a spring:

$Tm = -k(\theta - \theta 0)$



Schematic of a simple one-joint arm in a vertical plane with two antagonistic spring-like muscles.

Note that for simplicity we are ignoring muscle moment arms and we are assuming they are constant with respect to joint angle.

The $\theta 0$ parameter is the "rest length" angle for the elbow joint angle. This is the angle at which the muscles produce no force.

You may have noticed that our spring muscles can "push" as well as pull. To fix this we adjust the muscle equation a bit:

$TflexText = = [-k(\theta - \theta 0)] + [-k(\theta - \theta 0)] -$

The notation $y=[x]+$ and $y=[x]-$ means that y is x when $x>0$ (and zero otherwise) or $-x$ when $x<0$ (or zero otherwise), respectively.

So the equation of motion for linear spring-like muscles is:

$$I\ddot{\theta} = mglz\cos\theta + [-k(\theta - \theta 0)] + + [-k(\theta - \theta 0)] -$$

Here is a Python function that implements this:

```
def onejointmuscle(state,t,a0):
    m = 1.65   # kg
    g = -9.81  # m/s/s
    lz = 0.179 # metres
    I = 0.0779 # kg m**2
    k = -10.0  # Nm/rad
    a = state[0]
    ad = state[1]
    Tmf = max(k*(a-a0),0)
    Tme = min(k*(a-a0),0)
    add = (m*g*lz*cos(a) + Tmf + Tme) / I
    return [ad,add]
```

You can download a full program here that implements a simulation and animation of this version of the arm:

onejointmuscle_3.py

Questions

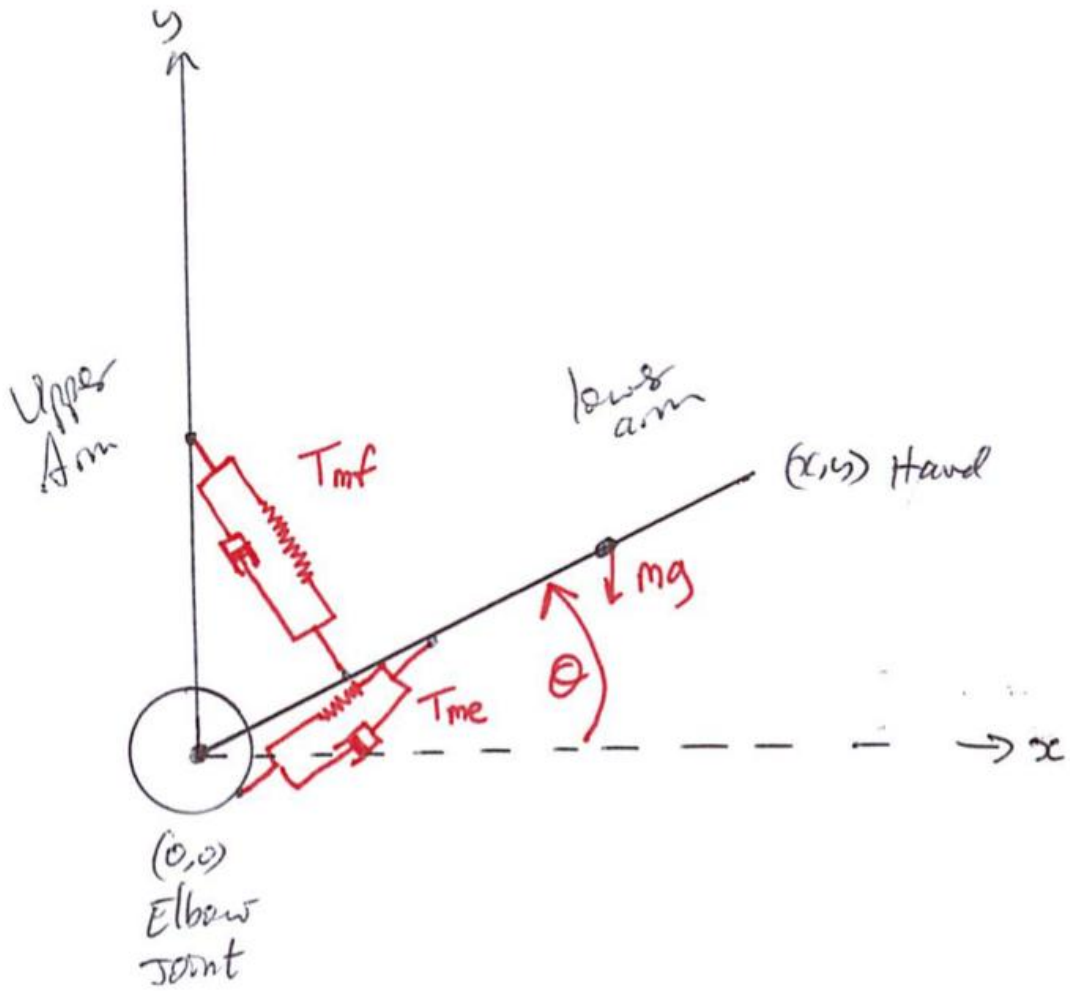
1. Play with the simulation found in onejointmuscle_3.py. Vary the value of k. What happens to the motion of the arm?
2. Vary the value of a0. What happens?
3. How would you change the code so that each muscle had its own stiffness parameter k and its own rest angle a0?

Muscles as Linear Spring-Dampers

A better approximation (yet still too coarse for good predictions about realistic muscle behaviour) is to incorporate a damper in the muscle. A damper is a mechanical unit

that delivers an opposing force that is proportional to velocity. It acts to "damp out" oscillations. Muscle force in a real muscle is also proportional to velocity, but in a more complex way, which we will see later. For now let's assume linear damping:

$TflexText==[-k(\theta-\theta 0)+b\dot{\theta}]+[-k(\theta-\theta 0)+b\dot{\theta}]-$



Schematic of a simple one-joint arm in a vertical plane with two antagonistic spring-like muscles with dampers.

```
def onejointmuscle(state,t,a0):
    m = 1.65    # kg
    g = -9.81   # m/s/s
    lz = 0.179  # metres
```

```
    I = 0.0779  # kg m**2
    k = -10.0   # Nm/rad
    b =  0.5    # Nms/rad
    a = state[0]
    ad = state[1]
    Tmf = max((k*(a-a0)) - (b*ad),0)
    Tme = min((k*(a-a0)) - (b*ad),0)
    add = (m*g*lz*cos(a) + Tmf + Tme) / I
    return [ad,add]
```

You can download a full program here that implements a simulation and animation of this version of the arm:

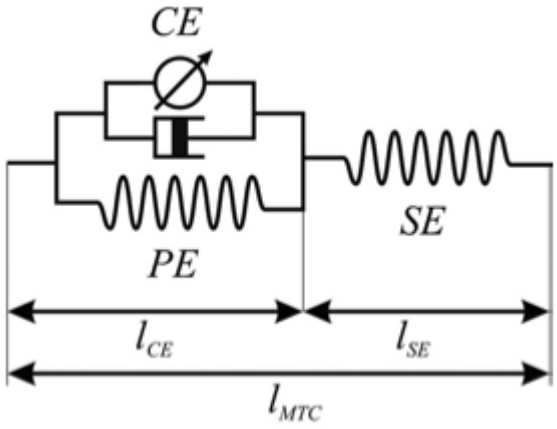
onejointmuscle_4.py

Questions

1. Play with the simulation found in onejointmuscle_4.py. Vary the value of b. What happens to the motion of the arm?
2. Can you find a combination of k, b and a0 that makes the arm rest in a stable equilibrium at $\theta = -\pi/6$ radians?

Introduction to Hill-type muscle models

In reality, muscles show way more complicated behaviour then depicted in the last section. Over the last decades, several different types of muscle models have been proposed in the literature. Most commonly used are the Huxley model (Distrubuted Moments Model) and the Hill-type muscle model. The main difference between the two models is that the Huxley model is a structural model, whereas the Hill-type muscle model is strictly phenomenological. Although both models have their own specific strengths and weaknesses, in general these models both describe the dynamic behaviour of real muscles fairly well. Hill-type muscle models are more frequently used, mainly because they are easier to implement.



Schematic of a Hill-type muscle model.

The Hill-type muscle model is build around three elements: a parallel elastic element (PE) a contractile element (CE) and a serial element (SE). The PE represents the collagen tissue that is present in the muscle belly. The CE represents the actual contractile properties of the muscle and the SE represents all the tendinous tissue in series with the CE. Below, a short simplified description of the elements is provided.

The SE

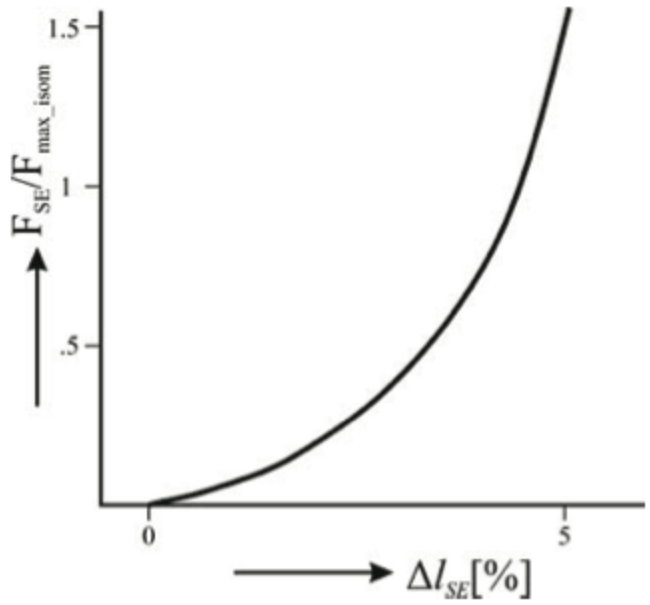
Collagen tissue is a passive material that behaves like a non-linear spring. Although the behavior of a tendon (or aponeurosis, etc) is quite complex, the "normal" working range is well described by a quadratic spring:

$$FSE=([kSE(lSE0-lSE)]+)2$$

Here, *kSE* is the tendon stiffness, *lSE* the tendon length and *lSE*0 the tendon slack length. *kSE* is usually measured in vivo using very fast perturbations, for example using the so-called "quick-release experiment":

- Hof, AL. In vivo measurement of the series elasticity release curve of human triceps surae muscle. J Biomech 1998 Sep;31(9):793-800

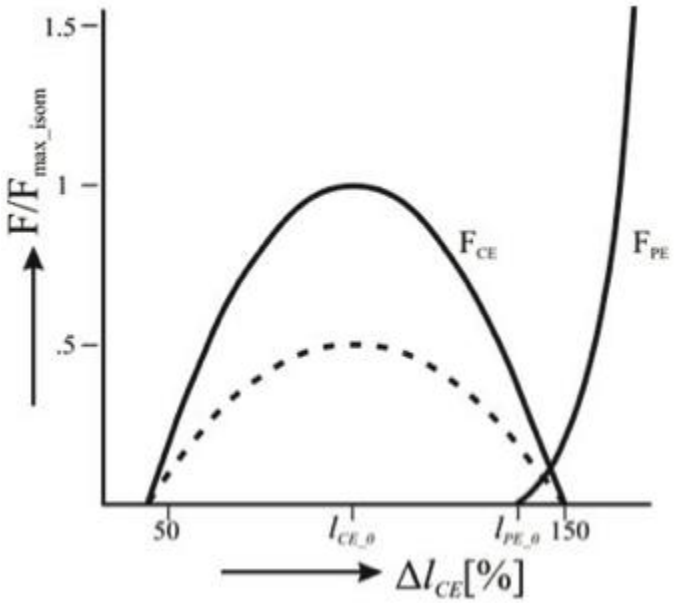
The slack length of tendons are muscle specific and are in general measured in cadaver studies. Here is an example of the (relative) force-length curve of a SE:



The force-length curve of a SE. Force is plotted relative to the maximal isometric force of the muscle.

The CE: the force-length relationship

Due to their contractile proteins (actin and myosin, etc.), muscles are capable of actively generating force. In contrast with the Huxley model, the Hill-type muscle model does not model the interactions of the proteins themselves, but rather the experimentally observed mechanical behaviour of these interactions. Two salient mechanical phenomena are typically observed in muscles: the force-length relationship and the force-velocity relationship. To keep things computationally simple, without harming the descriptive power of the model too much, the isometric force-length relationship is often described by a parabola:



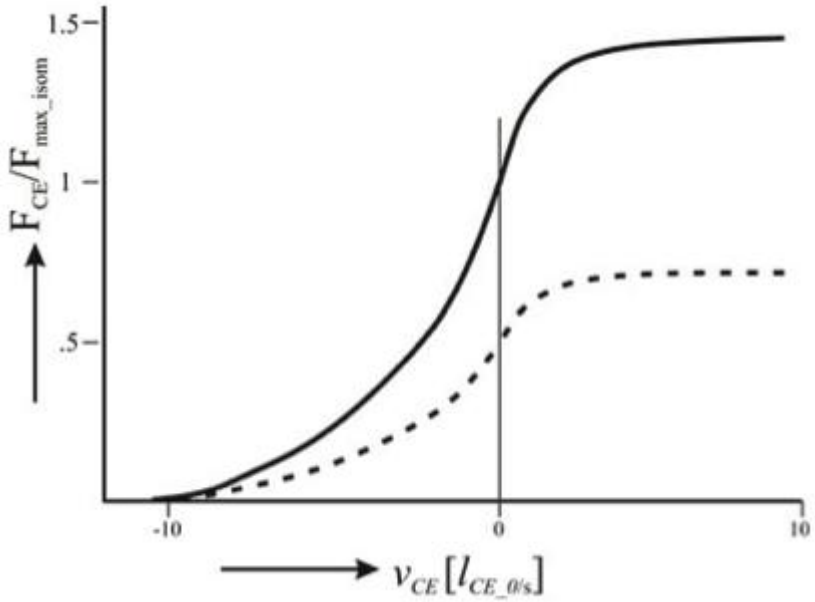
The isometric CE force (relative to the maximum isometric force) as a function of the length of the CE. The dashed line represents the isometric force when the muscle is stimulated half the maximum. Depicted is also the PE force. Note that the l_{PE} is equal to l_{CE}. The total force-length relationship is the sum of the active CE and passive PE force-length relationship. Also shown is the passive force of the PE: a quadratic spring that is similar to that of the SE.

The maximal isometric force of a real muscle can be estimated from cadaver studies by counting the amount of sarcomeres in parallel (or measuring the physiological cross-sectional area). The optimum length of a muscle can be estimated by counting the amount of sarcomeres in series, see for example:

- Murray WM, Buchanan TS, and Delp SL. The isometric functional capacity of muscles that cross the elbow. J Biomech 33: 943-952, 200

The CE: the force-velocity relationship

The force-velocity relationship is a bit more difficult that the force-length relationship. This relationship describes the phenomenon that the (maximal) force that muscles can deliver depends on the speed with which they contract. If the muscle shortens (concentric contraction), the maximal force decreases and if the muscle lengthens (eccentric contraction) the maximal force increases:



The CE force as a function of the contraction velocity. The dashed line represents the force-velocity curve when the muscle is stimulated half the maximal value.

Modeling the interaction between muscle and skeleton

In the previous section we have implicitly assumed that the "length" of the torque muscles change linearly with joint angle. In reality, the length of the whole muscle (i.e. the muscle tendon complex length l_{MTC}), depends also its origin and insertion and on the anatomy of the structures it is crossing. The relationship between l_{MTC} and joint angle is obtained in cadaver studies using a very elegant technique proposed by Grieve et al. in 1978 (Biomechanics VI-A, International series on Biomechanics, University Park Press, Baltimore). First, at a reference position, the l_{MTC} of a muscle is measured. Then, a piece of tendon is cut out and removed, and the distance between the two remaining parts is measured as a function of joint angle. The obtained data gives the change of l_{MTC} as a function of the angle. Together with the reference length, the kinematic relationship between l_{MTC} and joint angle is known. But what about the mechanical interaction?

The muscles deliver a torque on the skeleton. The torque delivered by a muscle equals the force they deliver multiplied with the moment arm (lever arm) of that muscle. But what about the moment arms of real muscles? The nice thing about measuring l_{MTC} as a function of joint angle, is that one gets the moment arm as a function of joint angle for free! Using the principle of "virtual energy", one can easily deduce that the moment

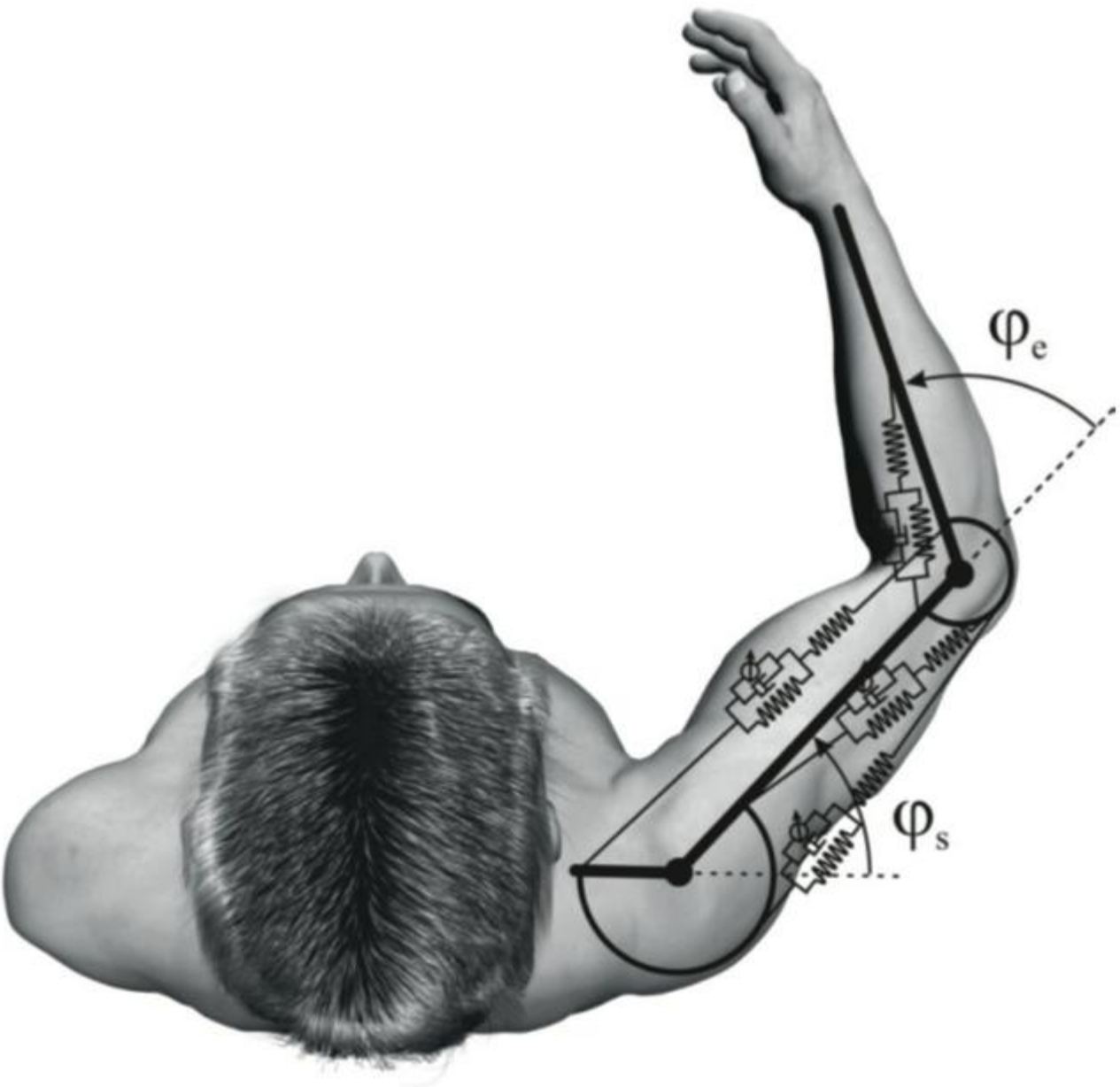
arm equals the change in muscle length divided by the change in joint angle (thus, the derivative of l_{MTC} with respect to joint angle:

$$momentarm = \partial lMTC \partial \theta$$

Simulations with a musculoskeletal model

Over the last decades several changes have been suggested to improve the predictive capacities of the Hill-type muscle model. For the interested reader, a detailed overview of a "full-blown" musculoskeletal model and its mathematical description can be found in:

- Kistemaker DA, Wong JD, Gribble PL (2010) The Central Nervous System does not minimize energy cost in arm movements. J Neurophysiol, 104, 2985-94



Schematic of "full-blown" musculoskeletal model described in Kistemaker et al. (2010).

In addition to the Hill-type muscle model, a model of activation dynamics is added. Activation dynamics is the process that takes place when an action potential arrives at a muscle. This AP causes the release of Ca2+ from the sarcoplasmatic reticulum in the intercellular which leads to free binding places for cross-bridges between actin and myosin.

There are many examples in the literature of using physiologically realistic (to varying degrees) musculoskeletal models to investigate questions of neural control of movement, sensory-motor learning, etc, in all sorts of model systems like arm movements, speech production, locomotion, posture and balance, jumping, etc. Here are a few:

- Kuo, A. D. (1995). An optimal control model for analyzing human postural balance. Biomedical Engineering, IEEE Transactions on, 42(1), 87-101.
- Gribble, P. L., & Ostry, D. J. (1996). Origins of the power law relation between movement velocity and curvature: modeling the effects of muscle mechanics and limb dynamics. Journal of Neurophysiology, 76(5), 2853-2860.
- Bobbert, M. F., Gerritsen, K. G., Litjens, M. C., & Van Soest, A. J. (1996). Why is countermovement jump height greater than squat jump height?. Medicine and Science in Sports and Exercise, 28, 1402-1412.
- Gribble, P. L., Ostry, D. J., Sanguineti, V., & Laboissière, R. (1998). Are complex control signals required for human arm movement?. Journal of Neurophysiology, 79(3), 1409-1424.
- Sanguineti, V., Laboissiere, R., & Ostry, D. J. (1998). A dynamic biomechanical model for neural control of speech production. The Journal of the Acoustical Society of America, 103, 1615.
- Todorov, E. (2000). Direct cortical control of muscle activation in voluntary arm movements: a model. Nature Neuroscience, 3, 391-398.
- Cheng, E. J., Brown, I. E., & Loeb, G. E. (2000). Virtual muscle: a computational approach to understanding the effects of muscle properties on motor control. Journal of neuroscience methods, 101(2), 117-130.
- Srinivasan, M., & Ruina, A. (2005). Computer optimization of a minimal biped model discovers walking and running. Nature, 439(7072), 72-75.
- Guigon, E., Baraduc, P., & Desmurget, M. (2007). Coding of movement-and force-related information in primate primary motor cortex: a computational approach. European Journal of Neuroscience, 26(1), 250-260.
- Raphael, G., Tsianos, G. A., & Loeb, G. E. (2010). Spinal-like regulator facilitates control of a two-degree-of-freedom wrist. The Journal of Neuroscience, 30(28), 9431-9444.

How are muscles controlled?

Once we have a physiologically realistic model of muscle force generation, the question arises, how does the brain control movement? How does the brain determine what

time-varying patterns of stimulaton to send down to muscles, in order to generate a desired movement?

A key insight is that there are many complex, non-linear relationships that sit in between neural control signals to muscles, and the arm movement that ultimately results. These include muscle mechanical properties that we have seen like force-length and force-velocity relationships, joint angle-dependent muscle moment arms, and there are many others including gradual development of muscle force over time (due to calcium kinetics), and dynamics introduced by series elastic elements (e.g. tendons).

Something we haven't talked about much yet is that the activation of spinal motoneurones controlling a muscle is determined not only by central efferent neural control signals but is also influence by afferent signals such as those from muscle spindles, golgi tendon organs, and signals from spinal interneurons. Thus the activation signal that gets ultimately sent to activate a muscle is determined by a very complex interplay between central efferent "control" signals, properties of the neuromuscular "plant", and afferent feedback signals.

Once you start to build up computational models that include all of these features in a realistic way, it's practically (and in some cases theoretically) impossible to "invert the system", and derive equations that give the neural control signals for a given desired movement. One influential model of motor control that proposes a physiologically motivated solution to the problem of how the brain controls movement, given these complexities, is the Equilibrium-Point Hypothesis.

Equilibrium-Point Control

Something you may have noticed about the simplified arm models presented above, is that when you have opposing muscles around a joint, the balance of forces between the two muscles defines an equilibrium joint angle - an angle for which the joint torques balance. According to the **Equilibrium Point Hypothesis**, movement arises by neurally controlled shifts in the equilibrium position of the limb from one static posture to another. Descending neural control signals define an equilibrium position for the limb, and movement is produced by gradually shifting this centrally specified equilibrium position over time.

Patterns of time varying muscle forces (and hence muscle activation patterns as measured by EMG) are not explicitly planned by the nervous system but rather unfold

as a natural consequence of the shifting equilibrium position of the limb and the spring-like properties of neurally activated muscle. This arrangement greatly simplifies the task of movement planning, eliminating the need to perform the often complex, nonlinear "inversions" required in order to explicitly specify the dynamics of multi-joint movement.

Two main variants of the EP hypothesis have been offered, and are known as the α and λ versions. Original proposals have centered around the control of movements at a single joint such as the elbow. According to the α version of the model proposed by Bizzi and colleagues, descending control signals specify levels of α motoneurone (MN) activity for antagonist muscles about a joint. For a given level of flexor and extensor MN activity, an equilibrium joint angle is specified as the angle at which flexor and extensor forces balance. Movement is produced from one joint angle to another through reciprocal changes to the balance of flexor and extensor MN activity. As the balance of forces change, the limb moves to a new position at which muscle forces, as defined by the new levels of flexor and extensor MN activation, balance. In addition joint stiffness may be controlled independent of movement through simultaneous increases or decreases in flexor and extensor MN activation, leading to muscle cocontraction around a joint.

In the α version of the EP hypothesis proprioception and afferent feedback play a limited role in MN activation, which is determined wholly by descending control signals. This assumption is based on studies of head movement and single joint elbow movement in monkeys showing that when unexpected loads are applied during movement, final position is not achieved until the loads are removed (Bizzi, Polit, & Morasso, 1976; Bizzi, Dev, Morasso, & Polit, 1978). In other words, load dependent changes in proprioceptive input did not alter neural input to muscles. Additional studies on deafferented animals showed that motor performance was relatively unaffected when proprioceptive input was eliminated altogether (Polit & Bizzi, 1978; Polit & Bizzi, 1979), supporting the idea that afferent information plays a minimal role in the achievement of final limb position. It should be noted however that normal execution of movement in the deafferented preparation was dependent on the animal having knowledge (through vision) of the initial position of their limb relative to the body. This suggests that even under these conditions afferent information plays some role in programming limb movement. The α version is a force control model since it posits that the centrally controlled variable is MN activation.

In contrast to the α version, in the λ version of the EP hypothesis proprioception plays a central role in movement, and in particular in determining MN activation levels. Consistent with physiological studies (see Houk & Rymer, 1981 and Rothwell, 1994 for reviews), in the λ version MN activation is determined by both central drive and afferent feedback from muscle spindles (Feldman, Adamovich, Ostry, & Flanagan, 1990). Descending control signals specify a threshold joint angle for MN recruitment (λ), such that as the difference between the current joint angle and λ increases, MN activation and muscle force increase. The physiological mechanism underlying this increase is assumed to be the tonic stretch reflex (Feldman, Adamovich, Ostry, & Flanagan, 1990). Thus in contrast to the α version, in the λ model EMG activity is a consequence of the shifting equilibrium and afferent reflexes, rather than a quantity under direct central control.

An appealing property of EP models is that at their heart they include the interaction between central efferent control signals and afferent signals, and their dual influence on muscle forces and movement.

Here are some papers on the EP hypothesis. There are many others.

- Bizzi, E., Accornero, N., Chapple, W., & Hogan, N. (1982). Arm trajectory formation in monkeys. Experimental Brain Research, 46(1), 139-143.
- Feldman, A. G. (1986). Once more on the equilibrium-point hypothesis (lambda model) for motor control. Journal of motor behavior, 18(1), 17.
- Bizzi, E., Hogan, N., Mussa-Ivaldi, F. A., & Giszter, S. (1992). Does the nervous system use equilibrium-point control to guide single and multiple joint movements?. Behavioral and Brain sciences, 15(04), 603-613.
- Gomi, H., & Kawato, M. (1996). Equilibrium-point control hypothesis examined by measured arm stiffness during multijoint movement. Science, 272(5258), 117-120.
- Gribble, P. L., Ostry, D. J., Sanguineti, V., & Laboissière, R. (1998). Are complex control signals required for human arm movement?. Journal of Neurophysiology, 79(3), 1409-1424.
- Shadmehr, R. (1998). The equilibrium point hypothesis for control of movement. Dept. of Bio. Eng. Johns Hopkins University.
- Gribble, P. L., & Ostry, D. J. (2000). Compensation for loads during arm movements using equilibrium-point control. Experimental Brain Research, 135(4), 474-482.

Force Control Models

In force control models, the hypothesis is that the brain is able to compute open-loop (no feedback) neural control signals to muscles for a given movement by using so-called "internal models" of the neuromuscular plant. This is an old concept from engineering and robotics that has been recently applied to the neural control of movement by the brain. The idea is that through experience, the brain learns an accurate "input-output" map of the motor system: the relationship between control inputs and motor outputs. This is known as a *forward model*. In these models the brain also has a neural representation of the *inverse model*, where given a desired movement the brain uses the inverse model to compute required control signals that produce that movement. The proposal also involves the idea that the brain can use the forward model to *predict* the consequences of a given motor command, and use that prediction as a proxy for feedback of the actual movement (which is delayed due to neural feedback delays, which can be quite long), to update the command, and hence achieve motor learning.

A major challenge to this class of models, in particular the idea of an inverse model, is centered around the criticism that a full characterization of the complete dynamics of the full-blown neuromuscular system, and a neural representation of its inverse, is wildly unrealistic. There is much empirical evidence that the motor system engages in "prediction" of the consquences of motor commands, and hence the idea of a forward model, at least in principle, is less controversial.

- Miall, R. C., & Wolpert, D. M. (1996). Forward models for physiological motor control. Neural networks, 9(8), 1265-1279.
- Wolpert, D. M., & Kawato, M. (1998). Multiple paired forward and inverse models for motor control. Neural Networks, 11(7), 1317-1329.
- Wolpert, D. M., Miall, R. C., & Kawato, M. (1998). Internal models in the cerebellum. Trends in cognitive sciences, 2(9), 338-347.
- Kawato, M. (1999). Internal models for motor control and trajectory planning. Current opinion in neurobiology, 9(6), 718-727.
- Desmurget, M., & Grafton, S. (2000). Forward modeling allows feedback control for fast reaching movements. Trends in cognitive sciences, 4(11), 423-431.
- Ostry, D., & Feldman, A. (2003). A critical evaluation of the force control hypothesis in motor control. Experimental Brain Research, 153(3), 275-288.
- Pasalar, S., Roitman, A. V., Durfee, W. K., & Ebner, T. J. (2006). Force field effects on cerebellar Purkinje cell discharge with implications for internal models. Nature neuroscience, 9(11), 1404-1411.

Hybrid Models

There is debate about the extent to which "pure" EP models can account for features of motor control and motor learning such as compensation for interaction torques, and motor learning. Thus hybrid models have been proposed in which there is a combination of "adjustable" open-loop control signals and EP-style feedback-control signals. This however does not solve the question of how the brain learns to shape the open-loop control signals.

Some review articles that cover these topics:

- Wolpert, D. M., & Ghahramani, Z. (2000). Computational principles of movement neuroscience. nature neuroscience, 3, 1212-1217.
- Wolpert, D. M., Ghahramani, Z., & Flanagan, J. R. (2001). Perspectives and problems in motor learning. Trends in cognitive sciences, 5(11), 487-494.
- Shadmehr, R., Smith, M. A., & Krakauer, J. W. (2010). Error correction, sensory prediction, and adaptation in motor control. Annual review of neuroscience, 33, 89-108.
- Krakauer, J. W., & Mazzoni, P. (2011). Human sensorimotor learning: adaptation, skill, and beyond. Current opinion in neurobiology, 21(4), 636-644.

Learning using feedback signals to train feedforward controllers

One of the challenges in the motor system when we start to think about learning from errors, is that movement errors are in a different coordinate frame than the control signals that lead to those errors. When our brain "guesses" at the control signals necessary to stimulate our arm muscles so that we throw a basketball towards a net, and we miss the net to the right, the motor error is in cartesian coordinates (the ball was 10 cm too high). How do we transform that "distal" error into an appropriate change in neural control signal to arm muscles?

A Neural Network model using combined Forward and Inverse Models

Michael Jordan (no, not that one, I'm talking about the statistician from MIT/Berkeley) and David Rumelhart proposed a solution to this that involves a clever combination of forward and inverse internal models, arrange in a multi-layer neural network. The basic idea is that first, a forward model is trained, for example by "motor babbling". The brain sends out (initially incorrect) motor commands, but monitors the input/output relationships, and models that, in the form of a forward model. Once

trained even to a partial extent, the forward model is able to predict (at least partially) the motor output given a motor command input. Once the forward model is trained at least to be partially accurate, it can be used to transform motor errors into required changes in motor commands.

- Jordan, M. I., & Rumelhart, D. E. (1992). Forward models: Supervised learning with a distal teacher. Cognitive science, 16(3), 307-354.

We haven't talked about neural networks yet, but it is our next topic, so we will return to this idea again alter.

Feedback Error Learning

Another interesting theory called *feedback error learning* proposes that the CNS uses afferent feedback signals from a simplistic (and only partially accurate) feedback controller (e.g. at the spinal cord) as a "teaching signal" to train an accurate central neural controller.

- Miyamoto, H., Kawato, M., Setoyama, T., & Suzuki, R. (1988). Feedback-error-learning neural network for trajectory control of a robotic manipulator. Neural Networks, 1(3), 251-265.
- Kawato, M., & Gomi, H. (1992). A computational model of four regions of the cerebellum based on feedback-error learning. Biological cybernetics, 68(2), 95-103.
- Predictive and feedback performance errors are signaled in the simple spike discharge of individual purkinje cells. Popa LS, Hewitt AL, Ebner TJ. J Neurosci. 2012 Oct 31;32(44):15345-58.

We will return to the question of motor learning after we talk about neural network models and learning more generally.

Source:
http://www.gribblelab.org/compneuro2012/6_Computational_Motor_Control_Muscle_Models.html