2D Muscle-Driven Gait Model (gait2d)
Reference Manual

Ton van den Bogert

February 3, 2010

Contents

1 Introduction

This document is the reference manual for the gait2d model, and its implementation in the Matlab MEX function gait2d.mexw32.

2 Model Reference

2.1 Body Segments and Coordinate Systems

The 2D gait model has seven body segments: trunk, and thigh, shank, foot in each leg. The model is defined as a tree structure with parent-child hierarchy:

GROUND

- Trunk
  - RThigh
    - RShank
      - RFoot
  - LThigh
    - LShank
      - LFoot

So, GROUND is the root segment. Each segment, except GROUND, has exactly one parent segment. For instance, the parent of RThigh is Trunk, and the Parent of Lthigh is also Trunk.

On each body segment we define a coordinate system. The origin of the segment is located at the joint where the segment is attached to the parent. Trunk is not jointed to the GROUND, so we can choose an arbitrary point, and we choose the hip joint as origin. The origin of thigh segments is at the hip, origin of shank segments is the knee, and origin of foot segments is the ankle. The X-axis of the body segment is defined as pointing anterior during neutral standing, and the Y-axis of the segment is defined as pointing upward during neutral standing. By neutral standing, we mean a posture where the long axes of the bones and trunk are perfectly vertical and the plantar surface of the foot is perfectly horizontal. The GROUND coordinate system has the X axis horizontal, and Y pointing upward. The origin of GROUND has no special meaning, other than as a reference for the ground surface which is at Y=0. The GROUND coordinate system is also known as the "global coordinate system".

The coordinate systems are visualized in Figure 1.

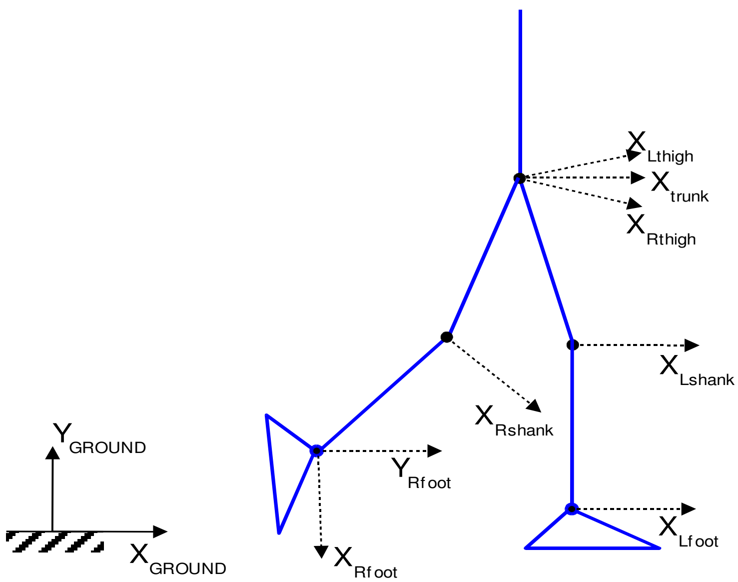


Figure 1: Body segments and reference frames. Y-axes were only drawn when not overlapping with stick figure.

The body segment coordinate systems are important when we need to change something in the model. For instance, the geometry of the undeformed foot shape, i.e. the contact points, is defined as X,Y coordinates in the foot coordinate system. The coordinate systems also help define the joint angles, as described in the next section.

2.2 Model parameters (gait2d_par.xls)

Model parameters are stored in the file gait2d_par.xls. This file will be read when the model is initialized.

In this Excel file, segment lengths and mass properties are calculated from subject mass and height according to Winter (2005). The only additional assumptions that had to be made was: heel-ankle horizontal distance is 6/180 times the subject height, i.e. 6 cm in a subject of 1.8 m height.

The gait2d MEX function has some functions to alter the body segment properties, for instance to add weight to the foot segment to represent a shoe. This will be described in section Error: Reference source not found

Only the white cells in gait2d_par.xls are modifiable by the user. Each parameter is defined within the file, but you may want to read the following sections of the documentation to understand what each parameter does.

2.3 Degrees of Freedom, Joints, Joint Angles, and Joint Moments

The model has nine kinematic degrees of freedom. Nine generalized coordinates (**q**) are defined:

1. Global X coordinate of trunk origin (hip)
2. Global Y coordinate of trunk origin (hip)
3. Global orientation of trunk
4. Right hip angle
5. Right knee angle
6. Right ankle angle
7. Left hip angle
8. Left knee angle
9. Left ankle angle

Units are meters (for 1 and 2) and radians (for the others).

Global coordinates are defined relative to GROUND. Global orientation is the angle between segment X axis and GROUND X axis, and is positive then segment is rotated counterclockwise relative to ground.

Joint angle is the angle between the segment and its parent segment, and is positive when the child segment is rotated counterclockwise relative to the parent. This means that a hip flexion angle is positive, a knee flexion angle is negative, and an ankle dorsiflexion angle is positive. You may need to apply some sign changes or constant offsets if you want to convert these joint angles to other (e.g. clinical) definitions.

When all generalized coordinates are zero, the model is standing with foot flat and all other segments vertical. Also, the hip will be positioned at GROUND origin. Usually, GROUND origin is placed on the ground contact surface. Therefore, this “zero” posture is not realistic until the second generalized coordinate is increased by one meter or so, to raise the model above the ground surface.

Nine generalized velocities ($\dot{\mathbf{q}}$) are defined, which are the time derivatives of the generalized coordiantes. The last six generalized velocities are the joint angular velocities in radians per second. Similarly, there are nine generalized accelerations $\ddot{\mathbf{q}}$

We also define six joint moments (stored in a vector $\boldsymbol{\tau}$). Units are Newton-meters (N m) and a joint moment is positive when it accelerates the child segment in a counterclockwise direction. This means that hip flexor moment is positive, knee extensor moment is positive, and ankle dorsiflexor moment is positive. In the model, joint moments will be generated by muscles and passive joint structures.

2.4 Air Drag

The model currently has no air drag, and is therefore a model of treadmill gait rather than overground gait. Air drag will be added in a future version because it is important for fast overground running.

2.5 Passive joint moments

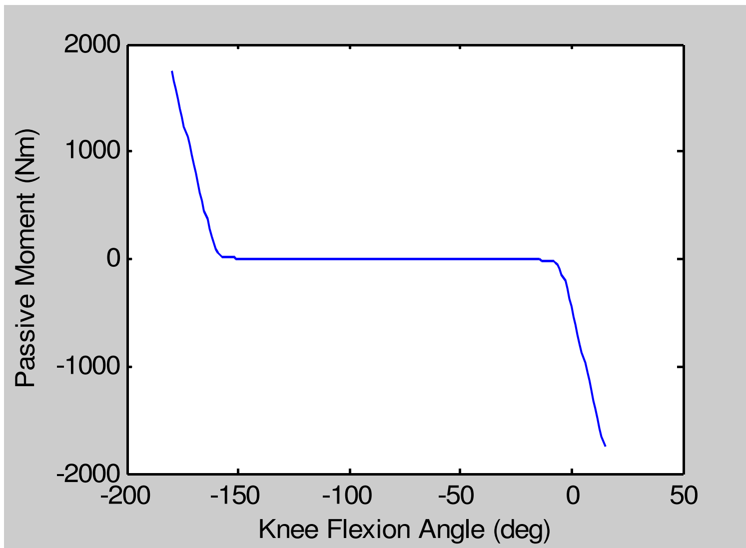
First, the muscles, when not activated, will introduce some passive joint moments when joint angles are far away from neutral. These can be seen in the passive isometric tests (section 4).

Second, we also have additional passive moment based on the range of motion in the joints. This is modeled as follows:

$$M = -k\left[f^+(\varphi - \varphi_{\max};\varphi_0) - f^+(\varphi_{\min} - \varphi;\varphi_0)\right] - B\dot{\varphi} \qquad (1)$$

The nonlinear function f^+ is defined in Appendix A. The range of motion parameters ($\varphi_{\min}$ and $\varphi_{\max}$) and stiffness parameters (k and φ_0) are set in the model parameter file gait2d_par.xls.

Here is the static passive moment-angle relationship for the knee, where the range of motion is from 160 degrees flexion to 5 degrees flexion. We have chosen $\varphi_0 = 2$ degrees and k = 5000 Nm/rad.



We have added a small amount of damping, this helps the numerical stability of simulation and optimization. A damping coefficient $B = 0.1$ N m s rad^{-1} is high enough and will have a negligible effect on joint moments, compared to muscle moments.

This passive moment model is quite simple and mainly intended to stop the movement from going outside this range. If we run simulations in which these passive moments have functional importance, we will need to make a more sophisticated model, based on literature data or our own measurements.

Relevant parameters in gait2d_par.xls, and their "normal" values:

	A	B	C	D	E	F	G
65							
66		5 JOINT RANGE OF MOTION PARAMETERS					
67		MinAngle	MaxAngle	JointK	JointPhi0	JointB	
68	R.Hip	-30	160	5000.0	2.0	0.1	
69	R.Knee	-160	5	5000.0	2.0	0.1	
70	R.Ankle	-60	60	5000.0	2.0	0.1	
71	L.Hip	-30	160	5000.0	2.0	0.1	
72	L.Knee	-160	5	5000.0	2.0	0.1	
73	L.Ankle	-60	60	5000.0	2.0	0.1	
74							
75		Definition of range of motion parameters					
76		MinAngle	[deg]	minimum joint angle			
77		MaxAngle	[deg]	maximum joint angle			
78		JointK	[Nm/rad]	Joint range of motion stiffness			
79		JointPhi0	[deg]	ROM transition region			
80		JointB	[Nms/rad]	Joint damping			
81							

Sheet1 Sheet2 Sheet3

2.6 Muscles

The 2D gait model contains 16 muscles, eight in each lower extremity. They are:

1. R.Iliopsoas
2. R.Glutei
3. R.Hamstrings
4. R.Rectus
5. R.Vasti
6. R.Gastroc
7. R.Soleus
8. R.TibialisAnt
9. L.Iliopsoas
10. L.Glutei
11. L.Hamstrings
12. L.Rectus
13. L.Vasti
14. L.Gastroc
15. L.Soleus
16. L.TibialisAnt

The muscle model is described in the following sections.

Activation dynamics

Active state of a muscle (denoted by the symbol *a*) is a number between 0 and 1 which represents the muscle's potential to produce force. At optimal fiber length, isometric force will be exactly *a**Fmax. Active state is controlled by neural excitation *u*, and this process is modeled as a first order differential equation according to He et al. (1991):

$$\frac{da}{dt} = (u - a)(c_1 u + c_2) \tag{2}$$

We use rate constants c1 = 3.3 s-1 and c2 = 16.7 s-1, which gives time constants of 50 and 60 ms for activation and decactivation, respectively, which is consistent with values given in Winters and Stark (1987).

NOTE: What we really want is to vary the rate constants depending on (u-a), not u. For instance, if u=0.5 and a=0, we are increasing active state via neural excitation and we should have a faster response than when u=0.5 and a=1 (deactivation). But for now, we just use He's equation. It does not make much difference.

Contraction dynamics

Each muscle is modeled as a 3-element Hill model.

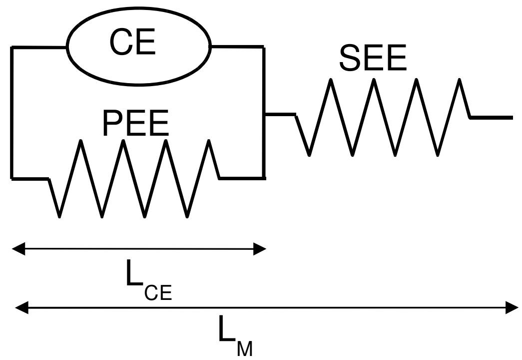


Figure 2: Arrangement of the three elements in the muscle model.

The contractile element (CE) represent contractile tissue, or muscle fibers, and the parallel elastic element represents passive properties of muscle fibers and surrounding tissue. The series elastic element represents the tendon and any elastic tissue in the muscle itself that is arranged in series with the muscle fibers.

The muscle length Lm depends on skeletal joint angles. The dynamics of length change in the CE are modeled by the following implicit differential equation which represents the force balance between the three elements of the muscle model:

$$aF_1(L_{CE})F_2(\dot{L}_{CE},a) + F_3(L_{CE}) - F_4(L_M - L_{CE}) = 0 \tag{3}$$

The components of this equation will now be presented.

F_1 is the isometric force-length relationship of the CE, approximated by a Gaussian curve:

$$F_1(L_{CE}) = F_{\max} e^{-\left(\frac{L_{CE}-L_{CEopt}}{WL_{CEopt}}\right)^2} \tag{4}$$

W is the width parameter of the force-length curve. We use the width values from Gerritsen et al. (1996), where a quadratic model was used rather than a Gaussian. However, the second derivative (curvature) of the force-length relationship, at its peak, is equal for both models. The Gaussian curve, however, will be wider at low forces. We choose the Gaussian in this project because it is a differentiable function. The quadratic had to be cut off below zero force, causing a discontinuity in first derivative. The general shape of F_1 is shown in Figure 3.

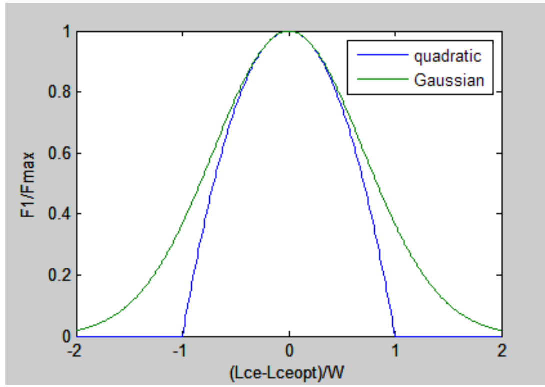


Figure 3: Normalized force-length relationship of the contractile element (CE).

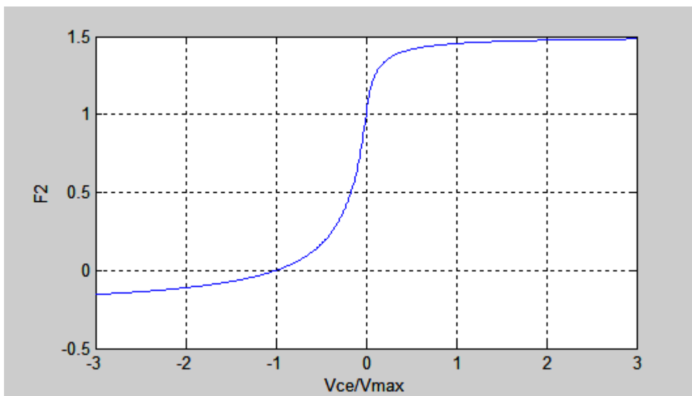


Figure 4: Normalized force-velocity relationship of the contractile element. Vce is negative for shortening and positive for lengthening.

F_2 is the normalized force-velocity relationship of the CE (Figure 4). We use the classical Hill-Katz relationship, using the two hyperbolic equations from McLean et al. (2003):

$$F_2(V_{CE}) = \begin{cases} \dfrac{\lambda(a)V_{\max} + V_{CE}}{\lambda(a)V_{\max} - V_{CE} / A} + \beta V_{CE} & \text{if } V_{CE} \le 0 \quad \text{(shortening)} \\ \dfrac{g_{\max} V_{CE} + c_3}{V_{CE} + c_3} + \beta V_{CE} & \text{if } 0 < V_{CE} \quad \text{(lengthening)} \end{cases} \tag{5}$$

The Hill curve parameter A was assumed to have a value of 0.25, and the maximal shortening velocity at full activation ($V_{\max}$) was assumed to have a value of $10 \cdot L_{CEopt}$ per second (Herzog, 2007). A scaling factor λ was introduced to account for the influence of activation level a on the maximal shortening velocity:

$$\lambda(a) = 1 - e^{-3.82a} + a \cdot e^{-3.82} \tag{6}$$

This relationship fits the experimental data of Chow and Darling (1999). In the present version of the software, we neglect this effect and assume:

$$\lambda(a) = 1 \tag{7}$$

For lengthening of the CE (eccentric contractions), the second hyperbolic relationship is used. The parameter $g_{\max}$, the maximal normalized eccentric muscle force, is usually assumed to be 1.5. The constant c3 is set to a value that produces a continuous first derivative at Vce = 0:

$$c_3 = \frac{V_{\max} A(g_{\max} - 1)}{A + 1} \tag{8}$$

NOTE: When the muscle fibers shorten faster than Vmax, a negative (compressive) force is produced by this model. This reaches a value of F_2=-A at infinite shortening velocity. This is acceptable because the series elastic element prevents such forces from being generated. This is also desirable because it keeps the derivative of F_2 continuous which is needed for optimization.

Finally, there is a small linear term βV_{CE} to ensure that we can always solve V_{CE} from the force-velocity equation even when the CE is not activated. This is needed when conventional explicit methods are used for simulation of the model. In the current version, we have coded this with $\beta = 0.001 F_{\max} L_{CEopt}^{-1}$ s.

The passive elastic elements (PEE and SEE) are modeled as nonlinear springs:

$$F(L) = \begin{cases} k_1(L - L_{\text{slack}}) & \text{if } L \le L_{\text{slack}} \\ k_1(L - L_{\text{slack}}) + k_2(L - L_{\text{slack}})^2 & \text{if } L > L_{\text{slack}} \end{cases} \tag{9}$$

This is a continuously differentiable function. The original quadratic model in McLean et al. (2003) did not have the k_1 term. This term was added to ensure that stiffness is never zero, which could cause singularity in implicit solution methods. Typically, k_1 is very small, to approximate zero force when length is below slack length. The value of k_1 is 10 Fmax per meter in the model, and this can presently not be altered by the user.

The stiffnesses k_2 and slack lengths of the PEE and SEE can be set by the user in the Excel file gait2d_par.xls. This is done indirectly, via the dimensionless parameters kPEE and umax:

$$k_2(PEE) = \frac{F_{\max} \cdot kPEE}{L_{CEopt}^2}$$

$$k_2(SEE) = \frac{F_{\max}}{\left(u_{\max} L_{CEopt}\right)^2}$$

Because kPEE and umax are dimensionless, we can make them the same for all muscles in this model. The normal values are kPEE = 1 and umax = 0.04.

Relevant parameters in gait2d_par.xls:

Section 1, General parameters

HillA	0.250	Normalized Hill constant "a" for muscles
Gmax	1.500	Maximal eccentric muscle force (normalized to Fmax)

Section 3, Muscle parameters

Fmax	[N]	Maximal isometric muscle force
Lceopt	[m]	Length at which contractile element (CE) can produce its highest force
Width		Half width of the CE force-length relationship, relative to Lceopt
PEEslack		Slack length of the parallel elastic element (PEE), relative to Lceopt
SEEslack	[m]	Slack length of the series elastic element (SEE), in meters
kPEE		Stiffness parameter of PEE, normalized to Fmax and Lceopt
umax		Strain in SEE when loaded by Fmax of muscle
Vmax	[s^-1]	Maximum shortening velocity, normalised to Lceopt
Tact,Tdeact	[s]	Activation and deactivation time constants of muscle

These parameters can be set separately for each of the 16 muscles.

2.7 Coupling between muscles and skeleton

In the 2D model, we assume that muscles have constant moment arms with respect to each joint. The moment arm is also the ratio between muscle-tendon length change and change in joint angle (An et al., 1984).

Therefore in each muscle, total muscle+tendon length is a linear function of the joint angles (q):

$$L_m = L_0 - \sum_i d_i q_i \tag{10}$$

Here, L_0 is the length of the muscle-tendon complex when all joint angles are zero, and d_i is the moment arm of the muscle with respect to joint *i*. A muscle that crosses one joint has only one moment arm. A two-joint muscle has two moment arms. If the muscle does not cross joint i, that moment arm is zero. The negative sign arises from the fact that a muscle shortens when the joint moves in the direction of the moment arm. This is the kinematic coupling between muscles and skeleton.

The dynamic coupling between muscles and skeleton is the relationship between the set of muscle forces **F** and the joint moments $\boldsymbol{\tau}$:

$$\boldsymbol{\tau} = \mathbf{D} \cdot \mathbf{F} \tag{11}$$

where **D** is a 6x16 matrix containing the moment arms. Element D_{ij} is the moment arm of muscle *j* at joint *i*. Most elements of the matrix **D** will be zero, because muscles typically only span one or two joints.

Relevant parameters in gait2d_par.xls:

L0	[m]	Muscle+tendon length when all joint angles are zero
dXXX	[m]	Moment arm of muscle with respect to joint XXX. Positive when muscle causes

		anterior swing of distal segment.

These parameters can be set separately for each muscle in the model.

2.8 Foot-Ground Interaction

The ground is at the line $Y_{global} = 0$.

For a given body posture, deformation *d* of a point on the foot is simply the amount of penetration into the ground that would have occurred if the foot had been rigid. Fur numerical reasons, we approximate this as a continuous function of the vertical coordinate *y* of the contact point:

$$d = \tfrac{1}{2}\left(\sqrt{y^2 + y_0^2} - y\right) \tag{12}$$

The parameter y_0 is the size of the transition region between contact and no contact. This parameter can be set in the parameter file gait2d_par.xls. We recommend $y_0 = 0.002$ m.

The model has two contact points on each foot: heel and toe. The heel is placed 6/180 cm behind the ankle, and the toe is placed (FootLength – 5 cm) in front of the heel. FootLength is calculated from body height using Winter (2005). The 5 cm subtraction for toe length is an arbitrary assumption. The vertical distance between ankle and heel is calculated from body height using Winter (2005). These calculations are done within gait2d_par.xls.

For body height of 180 cm, the contact point coordinates are (with respect to the foot coordinate system):

Heel: X = -0.0600 Y = -0.0702

Toe: X = 0.1636 Y = -0.0702

During movement, the global coordinates (x, y) of each contact point, and the corresponding velocities $(\dot{x}, \dot{y})$ are known as a function of system state.

At each point, the vertical force is calculated with the following equation:

$$F_y(d) = kd(1 - b\dot{y}) \tag{13}$$

The stiffness *k* and damping *b* can be set in the parameter file gait2d_par.xls. Gerritsen et al. (1995) used values $b = 1.0$ s m^{-1}. The stiffness is not critical as long as it is in a realistic range. We suggest k = 1e5 N m^{-1} which leads to 10 mm deformation when loaded with a typical ground reaction force of 1000 N.

NOTE: When the foot is pulled off the ground very quickly, this contact model can generate a negative (sticking) force. This has not been a problem in normal movements, but if it is a problem, alternatives are possible.

The horizontal force is modeled as a continuous approximation to Coulomb friction, in which force depends on the normal force F_y and the sliding velocity $\dot{x}$:

$$F_x(F_y, \dot{x}) = -\gamma F_y \frac{\dot{x}}{\sqrt{\dot{x}^2 + v_0^2}} \tag{14}$$

Where γ is the friction coefficient (a value of 1.0 was used in Gerritsen et al., 1995). The parameter v_0 must be small enough to approximate Coulomb friction, i.e. a change in sign of the force when there is a change in direction of sliding. We recommend a value of 0.1 m/s. At sliding speeds below v_0, the friction is more viscous (proportional to sliding speed) while at higher speeds the friction is more like a "dry" Coulomb friction model.

The parameters of the contact model are not critical for the model unless we are studying the effect of footwear or surface on movement and forces.

Relevant parameters in gait2d_par.xls:

Stiffness	1.00E+05	[N/m]	Stiffness of ground contact (nominal value 1e5 N/m)	k
Damping	1.000	[s/m]	Damping parameter for ground contact force	b
ContactY0	0.002	[m]	Deformation transition parameter in normal force model	y_0
ContactFric	1.000		friction coefficient	γ
ContactV0	0.100	[m/s]	Transition speed for friction model	v_0

3 System dynamics

3.1 Equations of motion

The multibody model has 18 state variables, the nine generalized coordinates $\mathbf{q}$ and the nine generalized velocities $\dot{\mathbf{q}}$.

Equations of motion for the model were derived using Autolev (Online Dynamics Inc., Sunnyvale, CA) in the following mathematical form:

$$\mathbf{M(q)} \cdot \ddot{\mathbf{q}} + \mathbf{B(q,\dot{q})} + \mathbf{C(q)} \cdot \boldsymbol{\tau} = \mathbf{0} \tag{15}$$

The first term are inertial effects, with $\mathbf{M}$ being a 9x9 mass matrix. The second term includes effects of centrifugal and coriolis forces, gravity, and ground contact forces. All of these are functions of generalized coordinates and velocities of the model. The last term is the effect of joint moments via a 9x6 coefficient matrix $\mathbf{C}$. One can easily see that this formulation is not unique. An equivalent system of equations van be obtained by multiplying this equation by any non-singular 9x9 matrix. Autolev has chosen the mass matrix such that the coefficient matrix $\mathbf{C}$ is a constant and sparse upper diagonal matrix, which is actually a form convenient for inverse dynamics where joint moments must be solved. In forward dynamics, one usually chooses to multiply by the inverse of the mass matrix, so that the generalized accelerations are solved explicitly and can be integrated into simulated motion. We will, however, leave the equation in this implicit form and solve it directly.

The equations of motion are converted by Autolev into C code which can be found in the file gait2d_al.c. The output vector “zero” contains the left hand side of the equation. The C code also calculates derivatives (Jacobian matrices) of the equation of motion with respect to **q**, $\dot{\mathbf{q}}$, $\ddot{\mathbf{q}}$, and $\boldsymbol{\tau}$. These derivatives are needed for the solution algorithms that are used for simulation and optimization of movement.

3.2 Muscle dynamics

The muscle model has two state variables for each muscle: the active state *a* and the contractile element length L_{CE}. There are 16 muscles and therefore 32 state variables associated with the muscle models.

The state equations for each muscle are the differential equations (2) and (3), which were provided above. There are a total of 32 state equations.

From these equations, one usually solves the derivatives of the state variables, but we will leave the equations as they are in their implicit form because we will use implicit methods anyway to solve the equations.

3.3 System dynamics

The combined (multibody and muscle) dynamics of the model is now described with an implicit differential equation:

$$\mathbf{f}(\mathbf{x},\dot{\mathbf{x}},\mathbf{u})=0 \tag{16}$$

The state vector **x**, for this model, contains 50 state variables: 9 generalized coordinates (**q**), nine generalized velocities ($\dot{\mathbf{q}}$), 16 muscle contractile element lengths ($\mathbf{L}_{CE}$), and 16 muscle active states (**a**).

$$\mathbf{x}=\begin{bmatrix}\mathbf{q}\\ \dot{\mathbf{q}}\\ \mathbf{L}_{CE}\\ \mathbf{a}\end{bmatrix} \tag{17}$$

The function **f**() has 50 equations, and consists of four parts:

- 9 identities d/dt(x(1:9) = xdot(1:9)
- 9 multibody equations of motion, eq. (15)
- 16 muscle activation dynamics equations (2)
- 16 muscle contraction dynamics equations (1)

One important feature of the model as we have constructed it, is that the function **f** is twice differentiable with respect to **x**, **x**dot, and **u**. This makes it possible to use implicit integration methods for simulation (Volino and Magnenat-Thalmann, 2005) and gradient-based optimal control methods for optimization (Betts, 2001).

We provide a MEX function gait2d.mexw32 which calculates the function **f**() and its derivatives (Jacobians) with respect to state **x**, state derivative **x**dot, and controls **u** .

4 Model testing

The program gait2d_test.m contains several tests to verify that the model and the software are working correctly.

Execute the program from the Matlab command console:

>> gait2d_test(command)

"command" is s string that tells the program which tests to run.

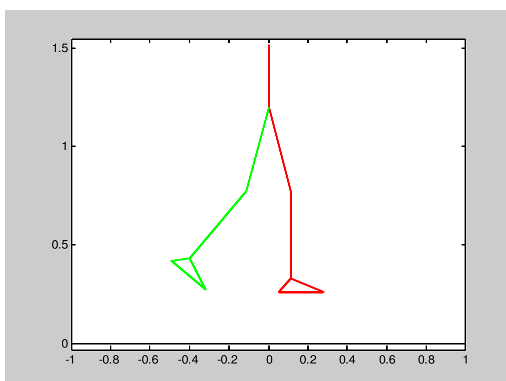
Please note that not everything in the gait2d_test.m program is useful as an example of how to use the MEX function. The derivatives test is designed to find programming errors and of no practical use. In the isometric and isokinetic tests, the MEX function outputs are used only partially so that the muscle contraction and muscle-skeleton coupling in the model can be tested seperately. The stick figure and simulate tests are simpler and give a better example of how the MEX function is typically used.

The following tests are currently available, and we present them in the order that they would normally be used by the developer of the model.

Stick figure test

>> gait2d_test('stick')

This draws a stick figure of the model in a typical state. This tests the forward kinematics part of the model, and the stick figure output. The Figure window should look like this:



Speed test

```
>> gait2d_test('speed')
```

This tests the execution time of the MEX function, with or without the multibody dynamics part. The MEX function is executed 10000 times with random inputs, and the resulting time is divided by 1000. The output will look like this:

```
Computation time for each model evaluation:  0.03236 ms
Computation time without Autolev code:       0.00908 ms
```

The first measurement is the time it takes to evaluate the dynamics residuals and all the Jacobians. The second measurement only includes the muscle models, not the multibody dynamics. This shows that the multibody dynamics (C code generated by Autolev) takes up most of the execution time.

Free fall dynamics test

>> gait2d_test('freefall')

This places the model in a the same state as during the stick figure test. We also deactivate the muscles and set the CE length long enough that the tendons are slack. Since the model does not touch the ground, it should be in freefall, i.e. vertical acceleration is -9.81. Since there are no muscle forces, all other accelerations in the model should be close to zero.

We set vertical acceleration to -9.81 m/s, all other state derivatives to zero, and then execute the MEX function to see the dynamics residuals. These should be close to zero. Not exactly zero, because the ground contact force is not exactly zero even above ground (section 2.8). Also the joint moments are not exactly zero, there is a small passive torque and there are small passive muscle forces.

Output in the Matlab command window should be as follows:

```
--------------------------------------------------
velocity    multibody    contraction    activation
--------------------------------------------------
   0.0000      0.0000      -0.0405        0.0000
   0.0000      0.3417      -0.0406        0.0000
   0.0000     -0.0263      -0.0404        0.0000
   0.0000      0.2844      -0.0003        0.0000
   0.0000     -0.8036      -0.0001        0.0000
   0.0000      0.0193      -0.0401        0.0000
   0.0000      0.6640      -0.0401        0.0000
   0.0000     -0.3885      -0.0403        0.0000
   0.0000      0.0131      -0.0403        0.0000
                           -0.0407        0.0000
                           -0.0405        0.0000
                           -0.0001        0.0000
                           -0.0001        0.0000
                           -0.0401        0.0000
                           -0.0401        0.0000
                           -0.0403        0.0000
--------------------------------------------------
GRF_heel =   -0.000    0.192
GRF_toe  =   -0.000    0.150
```

The four columns are subsets of the vector f (equation (16)) that comes back from the MEX function. f is a measure of how much the dynamics equations of the system are violated with the given inputs. Since we constructed state and state derivatives such that dynamics should be nearly satisfied, all values should be nearly zero.

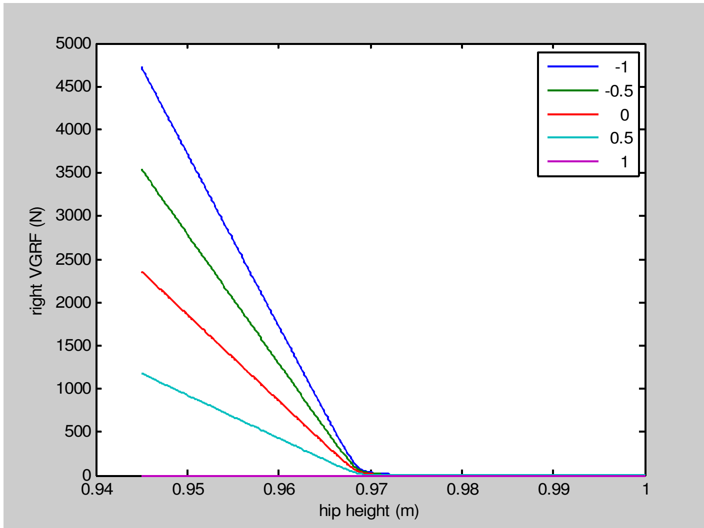
The first column with 9 values contains violations of the equality between generalized velocities and derivatives of generalized coordinates. The next 9 are related to multibody dynamics. Then 16 related to muscle contraction dynamics, and finally, 16 related to muscle activation dynamics.

If this test finds significant errors, the C and Autolev source code for the MEX function may have errors.

Ground reaction force test

>> gait2d_test('grf')

This tests the contact model. The model is put in a series of positions with different hip height. The vertical ground reaction force is plotted as a function of hip height and at five vertical hip velocities. This should show the force-deformation relationship of the contact model at each velocity. The legend shows the vertical velocities.



Derivatives test

>> gait2d_test('derivatives')

This tests the correctness of the analytical derivatives returned by the MEX function. The MEX function returns the dynamics violation vector f (equation (16)) as a function of state x, state derivatives xdot, and muscle stimulations u, and the derivatives (Jacobian matrices) df/dx, df/dxdot, and df/du.

The test generates random vectors x, xdot, and u. The MEX function is used to calculate the f and the Jacobians at these inputs. For the same inputs, the Jacobians are then also estimated via finite differences and repeated MEX function calls. The matrix elements with the highest difference between analytical and numerical result are reported.

An example of satisfactory output of the test:

```
Initializing model...
Checking dynamics derivatives...
Checking df/dx...
Max. difference:  0.01208 at 13 20 (90359.869867 vs. 90359.857786)
Checking df/dxdot...
Max. difference:  0.00001 at 13 15 (0.034972 vs. 0.034961)
Checking df/du...
Max. difference:  0.00000 at 49 15 (-16.473341 vs. -16.473341)
Checking dGRF/dx...
Max. difference:  0.00001 at 2 2 (0.000972 vs. 0.000980)
```

These differences are small enough to indicate success. If this test finds large errors, the most likely cause is errors in the C source code for the MEX function. When in doubt, run the test a few more times. Occasionally, with random inputs, there may be a relatively large error due to strong nonlinearities in the contact model, which are hard to test by finite differences.

Isometric strength test

This tests the muscle properties. Whenever muscle properties are changed as part of the research, it is important to run this test (and also the isokinetic test) to make sure that the desired functional changes were achieved.

```
>> gait2d_test('isometric')
```

This puts the model into various joint angle combinations and activates the muscle set three times for each joint. First, activating the muscles that contribute to positive moment. Second, activating the muscles that contribute to a negative moment. Third, no activation, to obtain the passive moment.

As explained in section 2.3, joint moments are defined as positive when causing a forward swing of the distal segments. Therefore, hip flexor moment, knee extensor moment, and ankle dorsiflexor moment are all positive. Negative moments are hip extensors, knee flexors, and ankle plantarflexors.

The Figure window will show the following result:

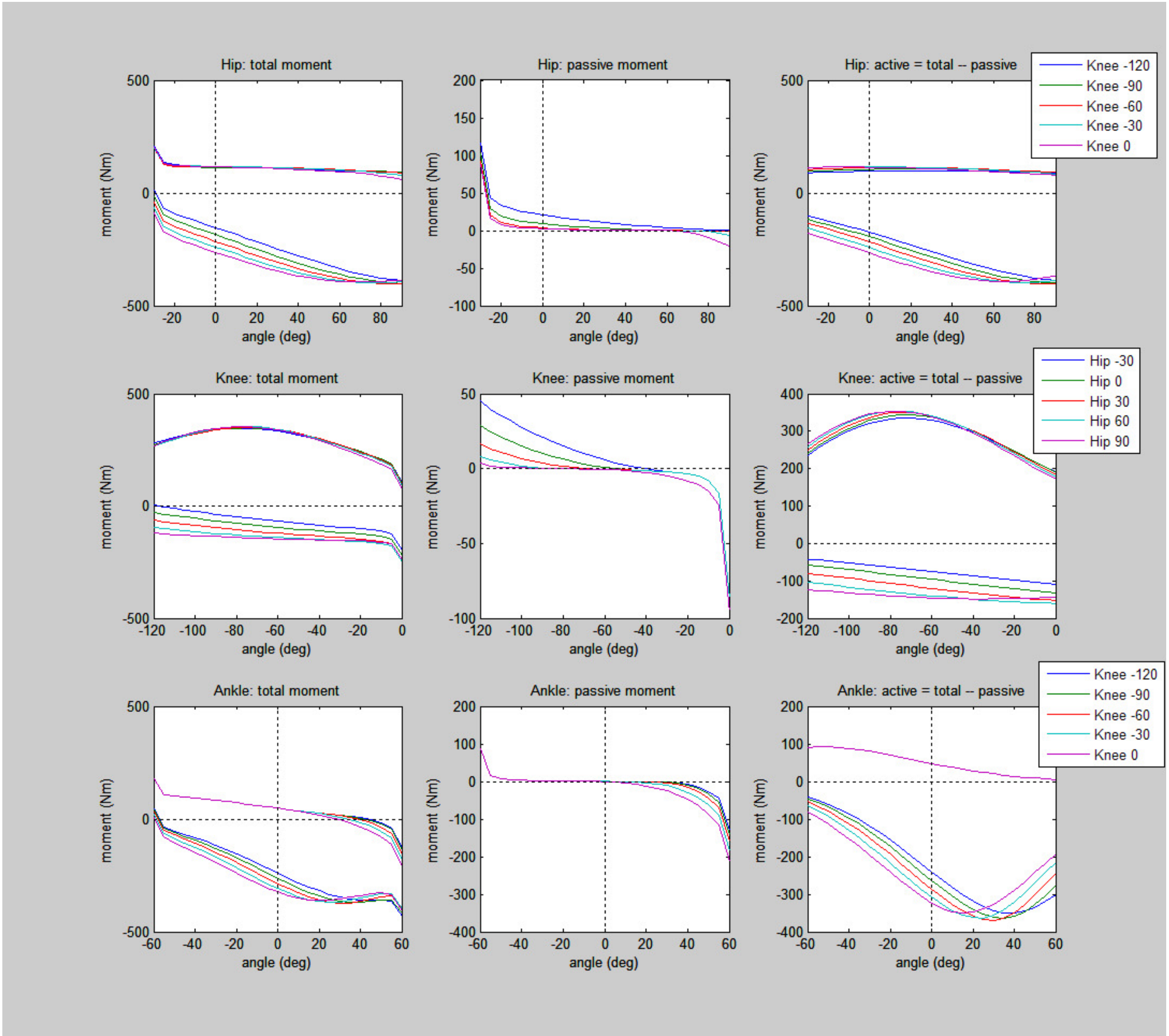


Figure 5: Simulated isometric joint moments.

These plots should be compared to corresponding human data, e.g. Kulig et al. (1984, active moments) or Riener and Edrich (1999, passive moments).

The leftmost column of this Figure shows the range of joint moments that can be achieved by the model. Muscular control can only vary the joint moment between the upper and lower curves.

When significant discrepancies between human and model are found, these should be eliminated by adjustment of muscle properties in the model (gait2d_par.xls).

Isokinetic test

>> gait2d_test('isokinetic')

For the isokinetic test, the model is placed in the neutral position (hips and ankles at zero degrees, knees at 5 degrees flexion). At each joint, angular velocity is then varied from -1000 to +1000 deg/s. At each angular velocity, the state of the system is found that produces steady state muscle forces (dF/dt = 0). The muscle forces are then converted to joint moments and plotted. This is done separately for the flexors and extensors.

The following graphs will be displayed:

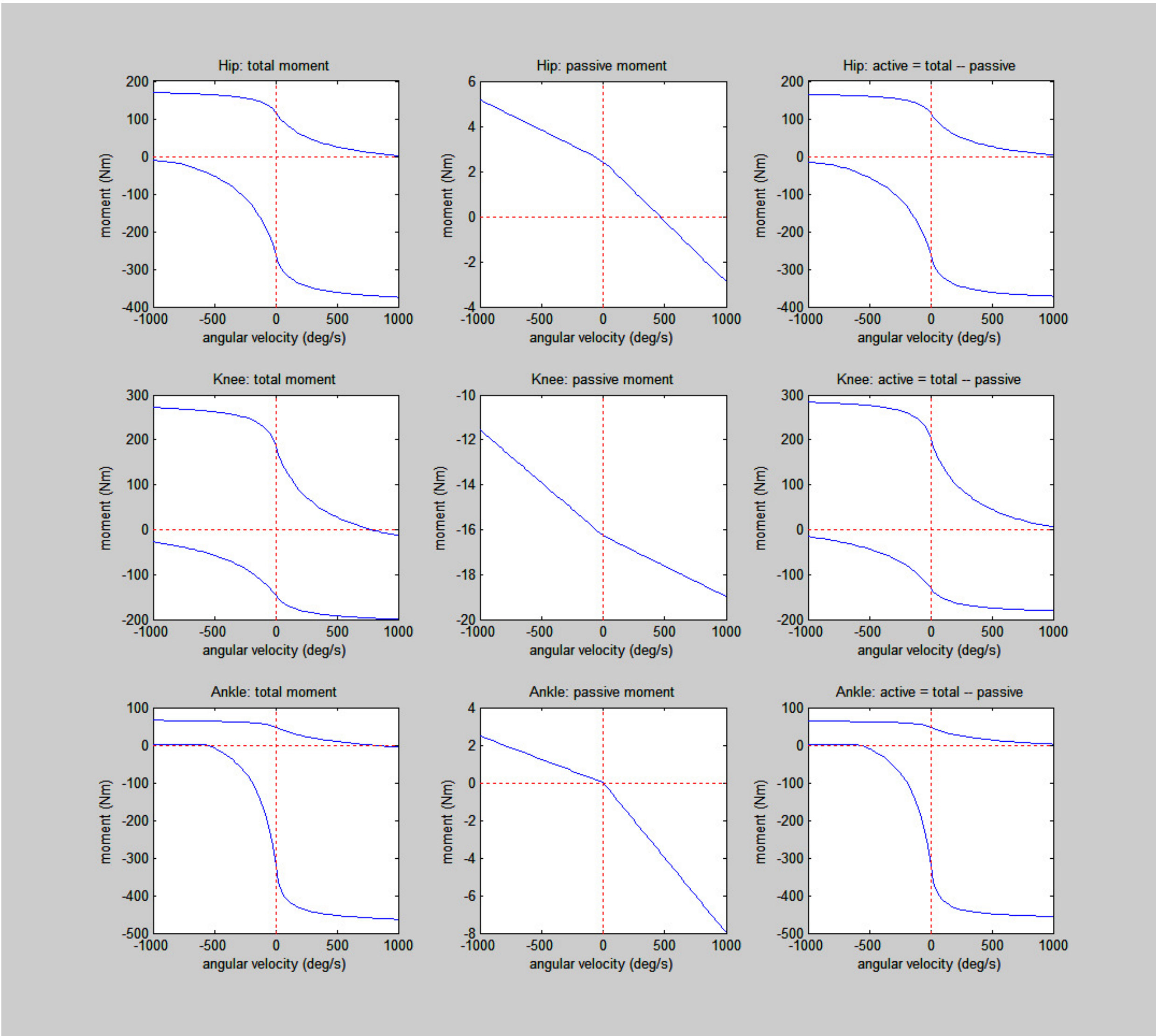


Figure 6: Simulated moment-angular velocity relationships at each joint. See text for details.

These graphs on the right show the typical hyperbolic force-velocity curves of active muscles:

Passive moments (middle column) are almost zero because we are at neutral position.

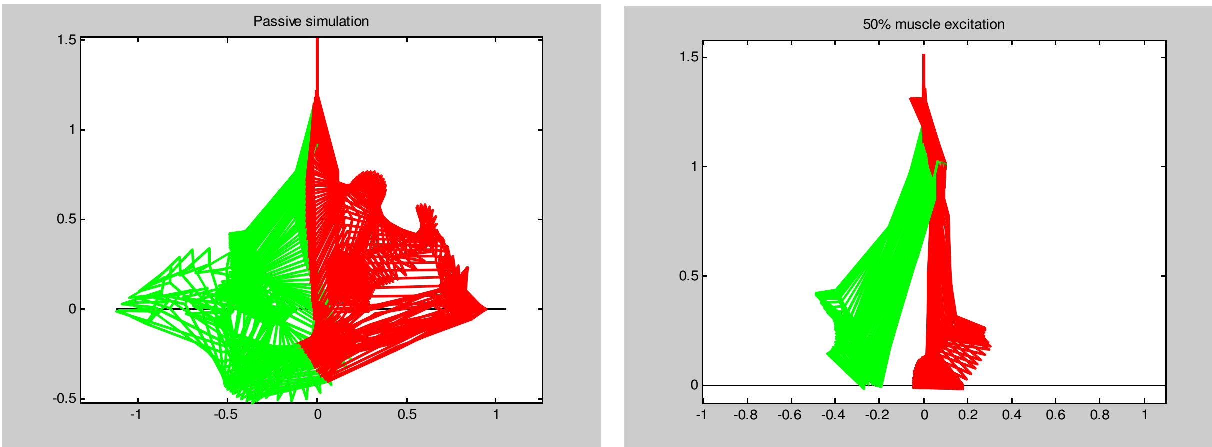
The leftmost column of this Figure shows the range of joint moments that can be achieved by the model. Muscular control can only vary the joint moment between the upper and lower curves.

Forward simulation test

>> gait2d_test('simulate')

The model is placed in the same posture as for the "stick" and "freefall" tests. We drop the model on the ground. This is done twice, once with passive muscles and once with active muscles. It uses the implicit midpoint method (Volino and Magnenat-Thalmann, 2005).

The results will be shown in two series of stick figures, as shown below:



The passive simulation shows the model collapsing (left). The active simulation shows stiff legs (right).

The Matlab console window will report the number of function evaluations. The code may be modified to investigate how the simulation accuracy depends on the number of time steps, but this is perhaps better done with more realistic simulations.

5 References

Ackermann M, van den Bogert AJ (2010) Optimality principles for model-based prediction of human gait. *J Biomech* **43**: 1055-1060.

An KN, Takahashi K, Harrigan TP, Chao EY (1984) Determination of muscle orientations and moment arms. *J Biomech Eng* **106**(3): 280-282.

Chow JW, Darling WG (1999) The maximum shortening velocity of muscle should be scaled with activation. *J Appl Physiol* **86**: 1025-1031.

Gerritsen KGM, van den Bogert AJ, Nigg BM (1995) Direct dynamics simulation of the impact phase in heel-toe running. *J Biomech* **28**: 661-668.

Gerritsen KGM, Nachbauer W, van den Bogert AJ (1996) Computer simulation of landing movement in downhill skiing: anterior cruciate ligament injuries. *J Biomech* **29**: 845-854.

He J, Levine WS, Loeb GE (1991) Feedback gains for correcting small perturbations to standing posture. *IEEE Trans Autom Control* **36**: 322-332.

Herzog W (2007) Muscle. In: *Biomechanics of the Musculoskeletal System*, 3rd Edition (ed: B.M. Nigg and W. Herzog). Wiley, New York, pp. 169-217.

Kulig K, Andrews JG, Hay JG (1984) Human strength curves. *Exerc Sport Sci Rev*. **12**: 417-66.

McLean SG, Su A, van den Bogert AJ (2003) Development and validation of a 3-D model to predict knee joint loading during dynamic movement. *J Biomech Eng* **125**: 864-874.

Riener R, Edrich T (1999) Identification of passive elastic joint moments in the lower extremities. *J Biomech* **32**(5): 539-544.

Volino P, Magnenat-Thalmann, N (2005) Implicit midpoint integration and adaptive damping for efficient cloth simulation: Collision Detection and Deformable Objects. *Comput. Animat. Virtual Worlds* **16**(3-4), 163-175.
Winter DA (2005) *Biomechanics and Motor Control of Human Movement*, 3rd Edition. Wiley.
Winters JM, Stark L (1987) Muscle models: What is gained and what is lost by varying model complexity. *Biol Cybern* **55**: 403-420.

APPENDIX A

In musculoskeletal models, we frequently see the following operation:

$$f^+(x) = \begin{cases} 0 & \text{if } x < 0 \\ x & \text{if } x \geq 0 \end{cases} \tag{18}$$

This is needed in contact models and in soft tissue models, where we have no force when there is no contact, or when the tissue is at negative strain.

For gradient-based methods to work, we need to approximate this by a twice differentiable function. It is also desirable if this can be coded without "if", because Autolev has no "if". Avoiding "if" also cause the function to be C-infinity, i.e. infinitely times differentiable (http://mathworld.wolfram.com/C-InfinityFunction.html).

The following function was found to work well:

$$f^+(x;x_0) = \tfrac{1}{2}\left(x + \sqrt{x^2 + x_0^2}\right) \tag{19}$$

where x_0 is the parameter that describes the size of the region where the transition between the two branches of the "if" occurs.

Below is a plot of this function for x_0 =0.1