

Biorobotic Laboratory EPFL



ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Modelling of stepping reflex and physical growth

Alexandre BELHASSEN

Supervision:
Amy WU
Prof. IJSPEERT Auke

Submitted on June 8, 2018

1 Introduction

Newborn and very young infants show growth at a very high rates. Not only they grow up but they also experience a considerable weight gain. These body changes have behavioral consequences and more specifically consequences on the locomotion capabilities of the infants. One theory that has been considered to explain this phenomenon is the maturation of the nervous system. This theory postulates that at the early stage of life the nervous system is not enough developed to generate stepping reflex and walking patterns. In the paper¹ *The Relationship between Physical Growth and a Newborn Reflex* by *Esther Thelen*, the authors develop another theory proposing that the stepping reflex exist in new born and very young infants but that the muscle gain is not proportional with the weight gain blocking the possibility to walk. The team developed 3 experiments to demonstrate the theory :

- Study 1 : Group of 40 infants selected, 20 boys and 20 girls with an average birth weight of 3,554 kg. Infants were observed at 2,4 and 6 weeks. The experiment consisted in an examiner holding infants under the armpits, to provide support since they infants can't stay up on their own, and making sure that both feet touched the ground. Stepping rate was measured
- Study 2 : In this experiment infant subjects,12 subjects, were overloaded with additional weights to observe what would be the consequences on stepping rate and on joint angles at maximum flexion.
- Study 3 : In order to support the idea that insufficient muscle development is at the origin of the absence of stepping the opposite experience to study 2 was developed. Measuring stepping rate in water in order to "diminish" the baby weight. 12 subjects were selected. Joint angles were also measured.

The aim of all these 3 experiments is to demonstrate the impact of mass gain and loss on stepping frequency and amplitude. The higher the mass gain the lower the stepping frequency and amplitude, the lower the mass gain (Archimede forces lowering the effect of gravity) the higher the stepping frequency and amplitude.

The goal here is to develop a model that can simulate a baby leg step in order to reproduce those three experiments and observe the effect on stepping amplitude. Frequency is not considered in the model. The model will allow to observe the effect of mass gain and loss on different parameters such as foot clearing height, step length and joint angles.

2 State of the art

Many solutions exist to model a leg and generate stepping. In this section we review some of this existing methods. The models are biomechanic models which are to analyze locomotion performance and understand locomotion principles.

2.1 Inverted pendulum

Inverted pendulum is commonly used to model walking. The leg is assimilated to a single axis with contact created on the ground with the foot and the mass at the top of the pendulum. The model is consistent with the energy exchange between kinetic and gravitational potential energy. The two energy fluctuations are out of phase. This model is the "simplest walking model" existing and presents limitations such as the absence of the knee joint.

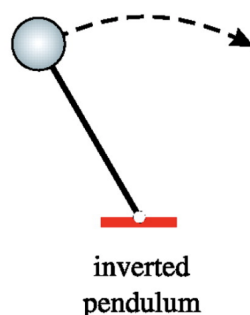


Figure 1: Inverted pendulum² to model walking pattern

2.2 SLIP Model

The spring loaded inverted pendulum (SLIP) is a strategy to model human locomotion and more precisely running. It is very efficient for predicting ground reaction forces and center of mass trajectories. Gait patterns from this model show self-stability if the leg stiffness k and the angle of attack α_0 (landing angle of spring) is adjusted properly. Walking can be modeled with the SLIP model by properly modeling the double support phase.

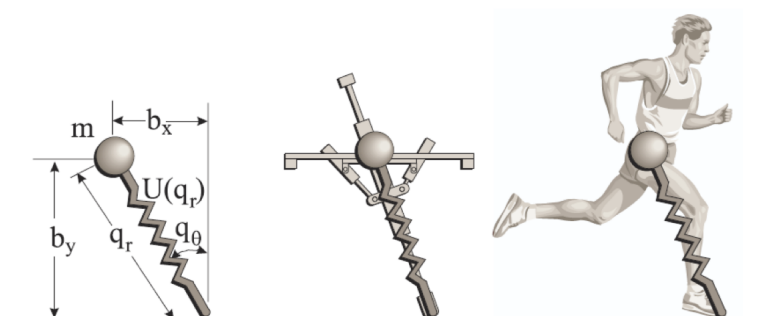


Figure 2: Inverted pendulum³ to model walking pattern

2.3 Double pendulum - Throwing model

One strategy to compensate for some of the limitation of the simple pendulum limitations is to use a double pendulum. the double pendulum allows to model the two segments of the leg and the knee and ankle joints.

The analogy between an arm throwing an object and a leg taking a step can be made. The model⁴ developed by R. Alexander presents a model simulating an object being thrown by an arm. The arm is modeled by a double pendulum where each segments represents a part of the arm. The model is composed of only 2 muscles each of which is either active or inactive. When

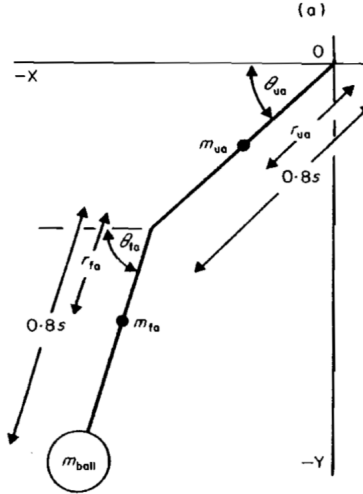


Figure 3: Arm modeling by R. Alexander

one muscle is activated at $t=0$ the other one is activated after a delay. The following equations rules this dynamic :

$$\left. \begin{array}{l} \text{if } \dot{\theta} < \dot{\theta}_{\max}, \quad T = T_0(\dot{\theta}_{\max} - \dot{\theta})/(\dot{\theta}_{\max} + 3\dot{\theta}) \\ \text{but if } \dot{\theta} \geq \dot{\theta}_{\max}, \quad T = 0, \end{array} \right\}$$

The models⁵ works the following way : At a time t when the muscle is activated a torque T is generated creating motion of the segment. Then the model looks for the optimum delay between activation proximal and distal muscle

2.4 Muscle-Reflex Model

The Muscle-Reflex Model is a model developed by *Geyer H1 and Herr H.*. The model developed relies on muscle reflexes, that allows to generate a stable walking pattern, walking dynamics, leg kinematics and self adaptation to slopes and ground disturbance. The study suggest that that mechanics and motor control are not separable, also it supports the idea that CPG in muscle activity may have a limited impact in locomotion. The absence of CPG in the model developed by the authors and the ability of generating a walking pattern without it suggest that no central input is required to achieve walking motion and muscle activity. The study supports that reflex inputs might be prevalent over the CPG, reflex inputs being the bridge between the nervous system and it's mechanical environment.

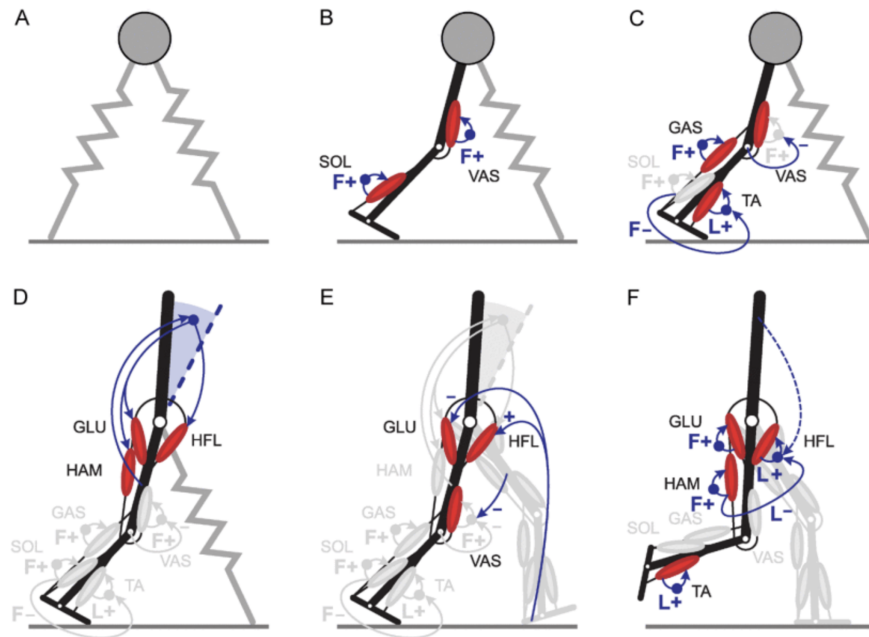


Figure 4: Evolution of the model developed by Geyer H1 and Herr H.. From a SLIP model to a segmented leg (two segments+foot). The point mass was replaced by a trunk and swing leg control was added.

2.5 Baby stepping model

No model for infant stepping seems to exist. Nevertheless previous models are relevant to the development of the baby stepping model. The model will resemble to the arm model developed by R. Alexandre since it will be a double pendulum but transposed into a leg.

3 Methods

3.1 Model

To model the infant leg a double pendulum was chosen since it allows to control the two segments of the leg and obtain a realistic movement of the leg. To do so torques are generated to create motion and the functioning of the simulation will be detailed in this section. The average infant

Variable	phi1	dtphi1	phi2	dtphi2	g	m1 (kg)	m2 (kg)	l1 (m)	l2 (m)
Initial Value	$-\pi/12$	0	$-\pi/12$	0	9.81	0.25	0.25	0.1	0.1

Table 1: Initial set value for the simulation, g,m1,m2,l1 and l2 are constants. Joint angles and angular velocities change over time.

weight when born is 3.5kg and size is comprised between 35 cm and 50 cm. From the table⁶ below masses of the baby leg were estimated. The model is as it follows, the pendulum is not an

Body Parts	Trunk	Thigh	Head	Lower leg	Upper arm	Forearm	Foot	Hand
Relative weight	50.80%	9.88%	7.30%	4.65%	2.7%	1.60%	1.45%	0.66%

Table 2: Average percentage of weight for each body part from the Human Body Dynamics: Classical Mechanics and Human Movement by Aydin Tozeren⁶

inverted pendulum since the infants are hold under the arms (at the waist in our case to simplify the problem). The leg start on the ground and then is moved until it touches back the ground.

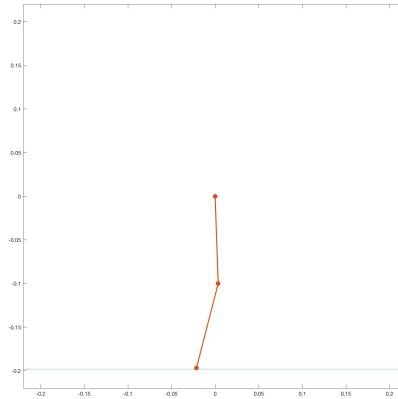


Figure 5: Matlab snapshot of the leg modeled as a double pendulum

3.1.1 Optimal solution search

A set of initial guess of torques u is given to the `fmincon` function that searches for the most optimal solution that validates the constrains applied. u is a vector of 10 values generated by the `rand` function.

3.1.2 Constraints

In order to generate a stepping pattern to the double pendulum multiple constraint were applied. The nonlinear constraints are applied through the `fmincon` function of Matlab. All the constrains are located within the `mycon` function.

- The first constraint insures that the second joint angle ϕ_2 never exceeds the first joint angle ϕ_1 . The knee joint allows flexion in one way only, this constrain validates this physiological requirement.
- The second constraint insures no excessive lifting of the leg from the ground to keep the movement to match best the walking pattern. The infant's leg measures 0.2 m, each segments is 0.1m, the condition prevents lifting of the end of the second segment (foot) above 0.04 m from the ground
- The third constraint can be compared to a stop condition. When the foot touches the ground the leg motion stops.

3.1.3 Torques

Torques Tfl and $Tfl2$ are implemented in the *doublependulumODE.m*. The torques are generated from the set of u generated by *fmincon* by the function *interp1*. Tfl is obtained by the *interp1* of the 5 first values of u and $Tfl2$ is obtained by the *interp1* of the 5 last values of u.

3.1.4 Boundaries

Lower boundaries and upper boundaries lb and ub respectively were determined from the from all the computed torques values in the normal situation. Maximum and minimum values for each of the 10 values of u were taken to obtain lb and ub . The constraints were then applied for the cases where weight is modified in order to "set the muscle strength", the idea is that without constraints the results of the simulation remain the same when weight is modified because the torques values are increased. Putting boundaries prevent that issue.

3.2 Equations

The following equations are for a double pendulum with an adjustable center of mass. Due to a lack of time those were not implemented in matlab. The equations used in the simulation are those for a classic double pendulum. We derive here the equations of motion for our double pendulum :

$$x_1 = l_1 a \sin \theta_1 \quad (1)$$

$$y_1 = -l_1 a \cos \theta_1 \quad (2)$$

$$x_2 = l_1 \sin \theta_1 + l_2 b \sin \theta_2 \quad (3)$$

$$y_2 = -l_1 \cos \theta_1 + l_2 b \cos \theta_2 \quad (4)$$

We obtain the following derivatives for those equations :

$$\dot{x}_1 = l_1 a \dot{\theta}_1 \cos \theta_1 \quad (5)$$

$$\dot{y}_1 = -l_1 a \dot{\theta}_1 \sin \theta_1 \quad (6)$$

$$\dot{x}_2 = l_1 \dot{\theta}_1 \cos \theta_1 + l_2 b \dot{\theta}_2 \cos \theta_2 \quad (7)$$

$$\dot{y}_2 = -l_1 \dot{\theta}_1 \sin \theta_1 + l_2 b \dot{\theta}_2 \sin \theta_2 \quad (8)$$

Now we can derive the Lagrangian

$$L = T - V \quad (9)$$

Where T is the kinetic energy and V is the potential energy.

$$L = \frac{1}{2} m_1 (\dot{x}_1^2 + \dot{y}_1^2) + \frac{1}{2} m_2 (\dot{x}_2^2 + \dot{y}_2^2) + \frac{1}{2} I (\dot{\theta}_1^2 + \dot{\theta}_2^2) - V \quad (10)$$

We obtain the following expressions for the kinetic energy and the potential energy :

$$T = \frac{1}{2}m_1(l_1^2 a^2 \dot{\theta}_1^2 \cos^2 \theta_1 + l_1^2 a^2 \dot{\theta}_1^2 \sin^2 \theta_1) + \frac{1}{2}m_2(l_1^2 \dot{\theta}_1^2 + l_2^2 b^2 \dot{\theta}_2^2 + 2l_1 l_2 \dot{\theta}_1 \dot{\theta}_2 \cos(\theta_1 - \theta_2)) + \frac{1}{2}I(\dot{\theta}_1^2 + \dot{\theta}_2^2) \quad (11)$$

$$V = gl_1 \cos \theta_1 (-m_1 a - m_2) - m_2 g_2 l b \cos \theta_2 \quad (12)$$

With the help of trigonometric identities we obtain the following final expression for the Lagrangian

$$L = \frac{1}{2}(m_1 a^2 + m_2)l_1^2 \dot{\theta}_1^2 + \frac{1}{2}m_2 l_2^2 b^2 \dot{\theta}_2^2 + m_2 l_1 l_2 b \dot{\theta}_1 \dot{\theta}_2 \cos(\theta_1 - \theta_2) + \frac{1}{2}I(\dot{\theta}_1^2 + \dot{\theta}_2^2) + gl_1 \cos \theta_1 (m_1 a + m_2) + m_2 g_2 l b \cos \theta_2 \quad (13)$$

$$\frac{d}{dt} \left(\frac{\delta L}{\delta \dot{\theta}_i} \right) - \frac{\delta L}{\delta \theta_i} = Q_i \quad (14)$$

We will first proceed with $\theta_i = \theta_1$

$$\frac{\delta L}{\delta \theta_1} = -m_2 l_1 l_2 b \dot{\theta}_1 \dot{\theta}_2 \sin(\theta_1 - \theta_2) - (m_1 a + m_2) g l_1 \sin \theta_1 \quad (15)$$

$$\frac{\delta L}{\delta \dot{\theta}_1} = (m_1 a^2 + m_2) l_1^2 \dot{\theta}_1 + m_2 l_1 l_2 b \dot{\theta}_2 \cos(\theta_1 - \theta_2) + I \dot{\theta}_1 \quad (16)$$

$$\frac{d}{dt} \left(\frac{\delta L}{\delta \dot{\theta}_1} \right) = (m_1 a^2 + m_2) l_1^2 \ddot{\theta}_1 + m_2 l_1 l_2 b \ddot{\theta}_2 \cos(\theta_1 - \theta_2) - m_2 l_1 l_2 b \dot{\theta}_2 \sin(\theta_1 - \theta_2) (\dot{\theta}_1 - \dot{\theta}_2) + I \ddot{\theta}_1 \quad (17)$$

Now with $\theta_i = \theta_2$

$$\frac{\delta L}{\delta \theta_2} = -m_2 l_1 l_2 b \dot{\theta}_1 \dot{\theta}_2 \sin(\theta_1 - \theta_2) - (m_1 a + m_2) g l_2 \sin \theta_2 \quad (18)$$

$$\frac{\delta L}{\delta \dot{\theta}_2} = m_2 l_2^2 b^2 \dot{\theta}_2 + m_2 l_1 l_2 b \dot{\theta}_1 \cos(\theta_1 - \theta_2) + I \dot{\theta}_2 \quad (19)$$

$$\frac{d}{dt} \left(\frac{\delta L}{\delta \dot{\theta}_2} \right) = m_2 l_2^2 b^2 \ddot{\theta}_2 + m_2 l_1 l_2 b \ddot{\theta}_1 \cos(\theta_1 - \theta_2) - m_2 l_1 l_2 b \dot{\theta}_1 \sin(\theta_1 - \theta_2) (\dot{\theta}_1 - \dot{\theta}_2) + I \ddot{\theta}_2 \quad (20)$$

So we now have the two following equation of motion

$$(m_1 a^2 + m_2) l_1^2 \ddot{\theta}_1 + m_2 l_1 l_2 b \ddot{\theta}_2 \cos(\theta_1 - \theta_2) - m_2 l_1 l_2 b \dot{\theta}_2 \sin(\theta_1 - \theta_2) (\dot{\theta}_1 - \dot{\theta}_2) + I \ddot{\theta}_1 + m_2 l_1 l_2 b \dot{\theta}_1 \dot{\theta}_2 \sin(\theta_1 - \theta_2) + (m_1 a + m_2) g l_1 \sin \theta_1 = \tau_1 \quad (21)$$

$$m_2 l_2^2 b^2 \ddot{\theta}_2 + m_2 l_1 l_2 b \ddot{\theta}_1 \cos(\theta_1 - \theta_2) - m_2 l_1 l_2 b \dot{\theta}_1 \sin(\theta_1 - \theta_2) (\dot{\theta}_1 - \dot{\theta}_2) + I \ddot{\theta}_2 + m_2 l_1 l_2 b \dot{\theta}_1 \dot{\theta}_2 \sin(\theta_1 - \theta_2) + (m_1 a + m_2) g l_2 \sin \theta_2 = \tau_2 \quad (22)$$

4 Results

4.1 Result tab

	Foot clearing height [m]	Step length [m]	Joint angles at max flexion phi1 and phi2 (rad)
Normal conditions	0.0342	0.0797	0.6173 and -0.7174
Weight added : 81.5g	0.0259	0.0759	0.4816 and -0.6674
Weight added : 163g	0.0193	0.0541	0.2959 and -0.6235
Weight added : 326g	0.0133	0.0304	0.1212 and -0.5529
Water conditions	0.0379	0.0794	0.6437 and -0.7479

Table 3: Result tab for 5 different cases

The foot clearing values show that the more mass we add the lower the foot clearing is observed, the same can be told about the step length and the two joint angles. The water situation shows an increase in values for the foot clearing and joint angles but not the step length.

4.2 Foot clearing height

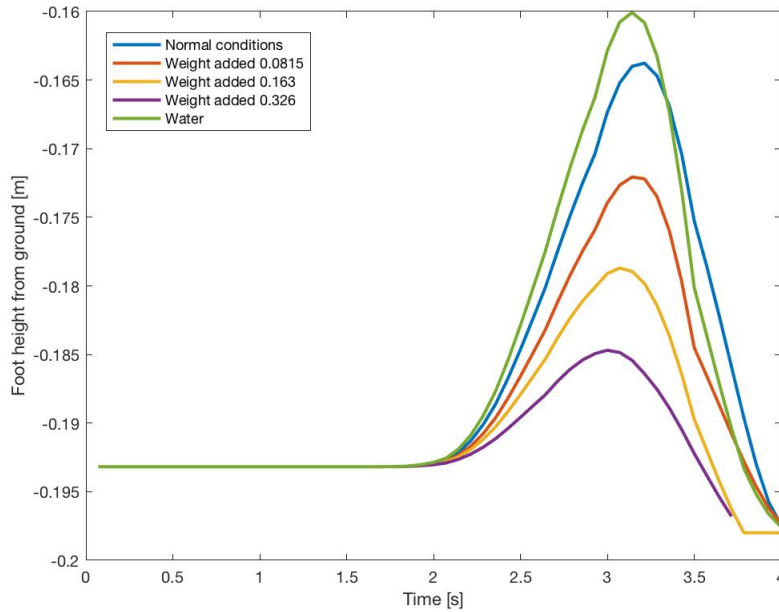


Figure 6: Foot clearing heights over time for different conditions. Simulation starts after approximately 2 seconds, no motion detected before that time.

To determine what would be the apparent weight for the new born leg in water we used the following formula :

$$\text{Apparent weight in water} = \text{weight} - \text{weight of the displaced fluid} \quad (23)$$

$$\text{Apparent weight in water} = 0.5\text{kg} - 0.15\text{kg} \quad (24)$$

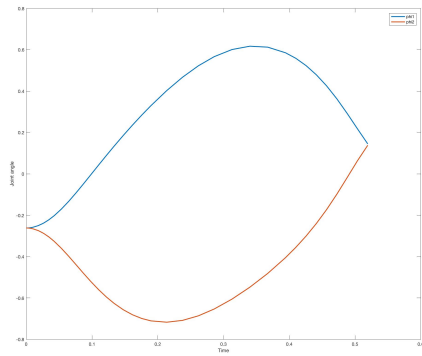
The previous graph shows the different foot clearing depending on the conditions. The foot clearing height is maximal in water and is minimal at the maximum weight added

(0.326kg) on one leg. Nevertheless the water curve should not be taken as completely accurate since the weight removed is removed unevenly on the two segments of the leg since we don't know how the buoyancy apply on the different segments.

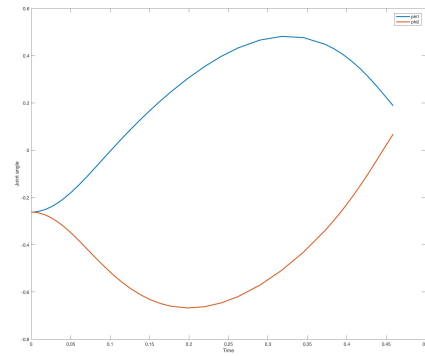
4.3 Step length

The result tab in section 4.1 shows that the step length is maximal in normal conditions and minimal when maximum weight is added. The step length in water is lower than the normal step length which should technically not be the case this might come from limitations of the modeling and from the fact that removing the weight on the different segments is done unevenly. Further calculation and force analysis should allow to determine precisely how to remove this weight in water, or simulate water.

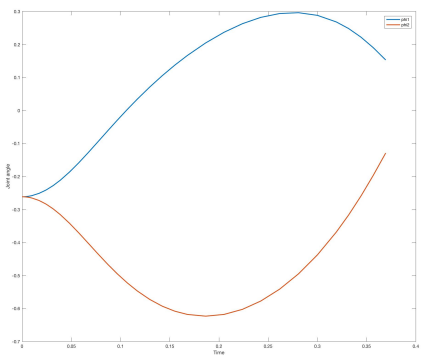
4.4 Joint Angles



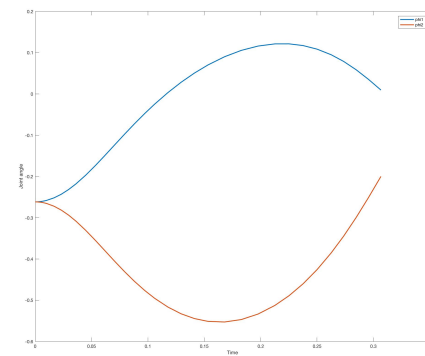
(a) Normal case $m_1=0.1$ $m_2=0.1$



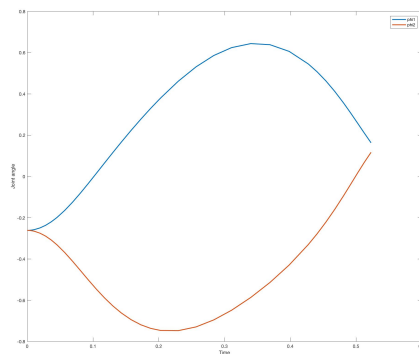
(b) Weight added first case +0.0815kg



(c) Weight added second case +0.163kg



(d) Weight added third case +0.326kg

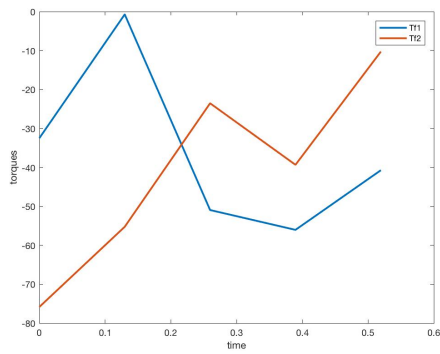


(e) Weight removed third case -0.15kg

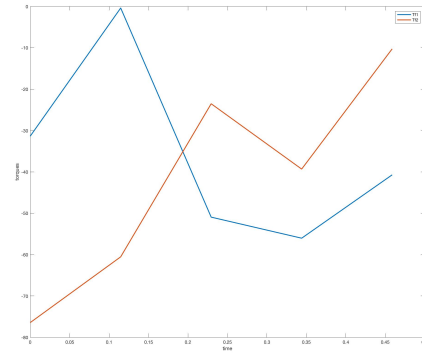
Figure 7: Plot of joint angle variation ϕ_1 and ϕ_2 in all different cases

The figure above show the variation of joint angles ϕ_1 and ϕ_2 compared to time. It can be observed that angles and their max values are lowered with the weight added. Angle values tend to increase a bit with water conditions.

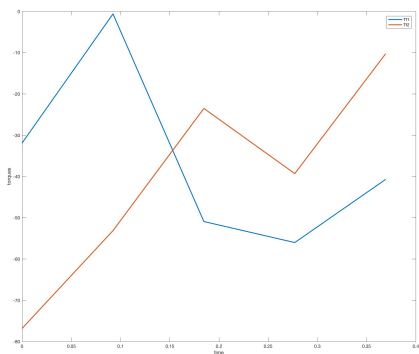
4.5 Torques



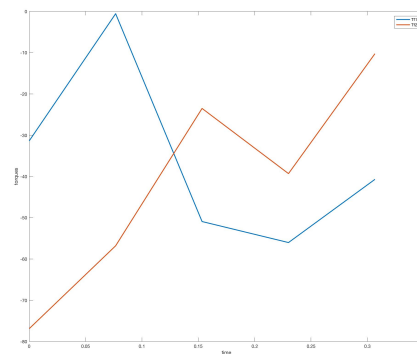
(a) Normal case $m_1=0.1$ $m_2=0.1$



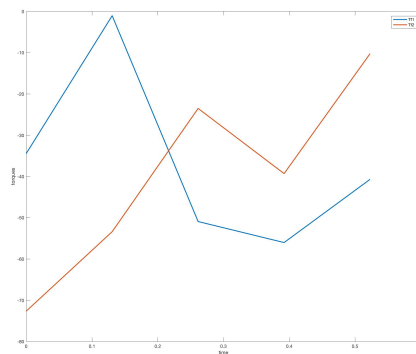
(b) Weight added first case $+0.0815\text{kg}$



(c) Weight added second case $+0.163\text{kg}$



(d) Weight added third case $+0.326\text{kg}$



(e) Weight removed third case -0.15kg

Figure 8: Plot of torques variation applied at the different joint angles. Each case corresponds an optimal set of torques determined by the `fmincon` function with constraints and boundaries applied (no boundaries in the first case)

The torques applied at the two joints seem to remain almost constant in all the different cases. Pattern and amplitude observed are the same for all five situations.

5 Discussion

In the paper¹ *The Relationship between Physical Growth and a Newborn Reflex* the authors measured multiple parameters. Among those they obtained data on the stepping rate of the infants. Our simulation does not provide data on the stepping rate. One of the problem with simulating the infants stepping rates is that we do not know the origin of it making it hard to implement it. Infant's locomotion at a very young age can be chaotic as can be seen in this [video](#). For these reasons we focused only on the taking of one step and extracted data on the following parameters : *Foot clearing height, Step length, Joint angles and Torques variation at joint angles*. In the paper's¹ discussion can be found the following sentence : *they did not lift their legs either as often or with as great an amplitude as when unweighted. In contrast, submerging infants' legs and thus reducing the effects of mass had a dramatic impact on infants' ease of stepping..* The simulation as can be seen in the results section lead us to the same conclusion. The foot clearing height is reduced when mass are added, the more consequent is the weight added the less is the height. The same phenomenon can be observed with the step length, which is reduced with the increase of the mass added. In both cases the results are in line with those obtained with the paper even though no numerical values are given on the paper for these parameters. We can however compare results obtained for the joint angles. In the paper a 8% decrease in hip joint angle is observed when weights are added, our results show a 20% decrease. For the knee the authors obtained a 4% decrease and our result show a 6%. For the hip the difference is a bit high but for the knee results are quite close. This might come from the limitations of the simulation, it can provide a step but not the most realistic step possible even though close. Finally the torques at the joint angles as can be seen on **Figure 7** are very similar on all situations which confirms that the same "muscle force" is applied in every situation making the results obtained plausible.

In water conditions the Archimedes forces have for goal to help the infant by "lowering their weight". Our results show amplitude increase for foot clearing and joint angles but not for the step length. The results from the paper for the joint angles show 8% hip angle increase, 15% knee increase, our results show 5% increase for the hip and 5% increase for the knee. The difference for the knee is considerable.

The model and the result obtained have some limitation. First we did not take into account the muscle gain over the weeks since there is no data for this, the problem is however limited since the paper's experience with mass added was performed only by 4 weeks old children. But modeling the muscle growth could be interesting to compare over the week the modification of the results by adding the increase of the torque possible to apply. Secondly for the experience in water, the "weight diminution" effect is not applied properly, simulating water and all the forces applied on the leg would be more accurate and could give more precise results.

Overall we managed to obtained results supporting the theory from the paper that the infant's inability to perform walking pattern originates at least (partially) from insufficient muscular development. As can be seen on the simulations at equivalent muscle force mass gain progressively reduced the ability to step and water situation helped stepping. Our model could benefit from further implementations such as seen with muscle-reflex model⁵ seen in the state of the art section, to match more realistic simulations. Finally the lack of data on infant's development (growth, weight gain, muscle gain) limits the accuracy of the data obtained.

References

- [1] Esther Thelen, Donna M Fisher, and Robyn Ridley-Johnson. The relationship between physical growth and a newborn reflex. *Infant Behavior and Development*, 7(4):479–493, 1984.
- [2] Hartmut Geyer, Andre Seyfarth, and Reinhard Blickhan. Compliant leg behaviour explains basic dynamics of walking and running. *Proceedings of the Royal Society of London B: Biological Sciences*, 273(1603):2861–2867, 2006.
- [3] William J Schwind and Daniel E Koditschek. Approximating the stance map of a 2-dof monopod runner. *Journal of Nonlinear Science*, 10(5):533–568, 2000.
- [4] R McN Alexander. Optimum timing of muscle activation for simple models of throwing. *Journal of Theoretical Biology*, 150(3):349–372, 1991.
- [5] Hartmut Geyer and Hugh Herr. A muscle-reflex model that encodes principles of legged mechanics produces human walking dynamics and muscle activities. *IEEE Transactions on neural systems and rehabilitation engineering*, 18(3):263–273, 2010.
- [6] Aydin Tözeren. *Human body dynamics: classical mechanics and human movement*. Springer Science & Business Media, 1999.

6 Appendix

```
function InfantStepping
% Optimization of an pendulum from resting position to swing up
% Finds optimal set of control inputs

%save optimal parmaeters

global N
N = 10;

%save u values
global torques
torques = [];

% global angle1
% global angle2

global timespaced
% initial guess
%u0 = -60*rand(N,1)
u0= [-31.9645
     -0.5069
     -50.9478
     -56.0396
     -40.7241
     -76.9249
     -53.1384
     -23.5336
     -39.3287
     -10.2712]
% u0 = [-0.0076 -4.1206 -10.3051 -0.3699 -7.6786 -3.9123 -7.6106 0.1235 -5.1291 -2.5615]
%torques

% options to see details of each iteration
options = optimset('Display','iter-detailed','PlotFcn',@optimplotfval);

%bound constrains
lb=[-35.3910 -1.6722 -50.9478 -56.0396 -40.7241 -76.9494 -80.9253 -23.5336 -39.3287 -10.2712];
ub=[-31.3782 -0.3847 -50.9478 -56.0396 -40.7241 -71.6137 -47.5505 -23.5336 -39.3287 -10.2712];

% run fmincon optimization
runopt=1;
if runopt
    [uopt,fval,exitflag,output] = fmincon(@myopt,u0,[],[],[],[],lb,ub,@mycon,options)
else
    uopt=u0;
end

% make plots to visual results
[ivp,duration] = init();
tspan = [0,duration];
optionsode = odeset('events',@eventstop);
[ts,xs] = ode45(@(t,x) double_pendulum_ODE(t,x,uopt),tspan,ivp,optionsode);

timespaced =[timespaced,ts];
save('timespaced.mat','timespaced');

cmap = hsv(length(ts));
```

```

% figure %replace
% plot([0 sin(xs(1,1))],[0 -cos(xs(1,1))],'k','LineWidth',2)
% xlim([-1 1]); ylim([-1 1])
% hold all
% for k = 1:length(ts)
%   plot([0 sin(xs(k,1))],[0 -cos(xs(k,1))],'Color',cmap(k,:))
%   pause(0.2)
% end
% plot([0 sin(xs(end,1))],[0 -cos(xs(end,1))],'k','LineWidth',2)
% hold off
%

phi1=xs(:,1)'; dtphi1=xs(:,2)';
phi2=xs(:,3)'; dtphi2=xs(:,4)';
l1=ivp(8); l2=ivp(9);
g=ivp(5); m1=ivp(6); m2=ivp(7);
movie = 0;
% ivp=[phi1; dtphi1; phi2; dtphi2; g; m1; m2; l1; l2];
figure(5);
plot(ts,phi1,'LineWidth',2);
hold on
plot(ts,phi2,'LineWidth',2);
xlabel('Time'); ylabel('Joint angle'); legend('phi1','phi2')
hold off;
% angle1 = [angle1,phi1];
% save('angle1.mat','angle1')
% angle2 = [angle2,phi2];
% save('angle2.mat','angle2')
figure(2);
%Ycoordground=[0,-l1*cos(ivp(1)),-l1*cos(ivp(1))-l2*cos(ivp(3))];
Ycoordground=-0.198;
plot([-1 1],[Ycoordground Ycoordground])
hold;
h=plot(0,0,'MarkerSize',30,'Marker','.', 'LineWidth',2);
range=1.1*(l1+l2); axis([-range range -range range]); axis square;
set(gca,'nextplot','replacechildren')

% global Max_foot_Clearing
% global Step_distance

    for i=1:length(phi1)-1
        if (ishandle(h)==1)
            Xcoord=[0,l1*sin(phi1(i)),l1*sin(phi1(i))+l2*sin(phi2(i))];
            Ycoord=[0,-l1*cos(phi1(i)),-l1*cos(phi1(i))-l2*cos(phi2(i))];
            Max_foot_Clearing = [Max_foot_Clearing, Ycoord(3)];
            save('foot_clearing.mat','Max_foot_Clearing');
            Step_distance = [Step_distance, Xcoord(3)];
            save('Step_distance.mat','Step_distance');
            set(h,'XData',Xcoord,'YData',Ycoord);
            drawnow;
            F(i) = getframe;
            if movie==false
                pause(ts(i+1)-ts(i));
            end
        end
    end
end
if movie==true
    movie2avi(F,'doublePendulumAnimation.avi','compression','Cinepak','fps',fps)
end

```



```

% constraints - final condition must be (pi,0)
function [c,ceq] = mycon(u)

% x = (theta,thetadot), which is initially at (0,0)
options = odeset('events',@eventstop);
[ivp,duration] = init();
tspan = [0,duration];
[ts,xs] = ode45(@(t,x) double_pendulum_ODE(t,x,u),tspan,ivp,options);
ts2=linspace(0,ts(end),10);
xs2=interp1(ts,xs,ts2);
c = zeros(2*length(xs2)+1,1);
c(1:length(ts2)) = xs2(:,3)-xs2(:,1);

% global torques
% torques = [torques, u];
% save('torques.mat','torques')
x = xs(end,:);
u;
phi1 = x(1); dtphi1 = x(2); phi2 = x(3); dtphi2 = x(4);
g=x(5); m1=x(6); m2=x(7); l1=x(8); l2=x(9);

Ycoordground=-0.198;
Ycoord=[0,-l1*cos(phi1),-l1*cos(phi1)-l2*cos(phi2)];
Xcoord=[0,l1*sin(phi1),l1*sin(phi1)+l2*sin(phi2)];

x0 = xs(1,:);
Ycoord0=[-l1*cos(x0(1))-l2*cos(x0(3))];
Xcoord0=[l1*sin(x0(1))+l2*sin(x0(3))];
Ycoord3=[-l1*cos(xs2(:,1))-l2*cos(xs2(:,3))];
c(end) = Xcoord0-Xcoord(3);
c(length(ts2)+1:end-1) = Ycoord3+0.16;
ceq = [Ycoordground-Ycoord(3)];
end

% cost/fitness function to minimize
function costFunction = myopt(u)

%savefunction in sstruct result.timestamp
[ivp,duration] = init();
tspan = [0,duration];
optionsode = odeset('events',@eventstop);
[ts,xs] = ode45(@(t,x) double_pendulum_ODE(t,x,u),tspan,ivp,optionsode);

x = xs(end,:);
phi1 = x(1); dtphi1 = x(2); phi2 = x(3); dtphi2 = x(4);
g=x(5); m1=x(6); m2=x(7); l1=x(8); l2=x(9);
Xcoord=[0,l1*sin(phi1),l1*sin(phi1)+l2*sin(phi2)];
costFunction = -Xcoord(3); %-Xcoord(3)/ts(end) speed is not the best
end

function [value, isterminal, direction] = eventstop(t, x)
phi1 = x(1); dtphi1 = x(2); phi2 = x(3); dtphi2 = x(4);
g=x(5); m1=x(6); m2=x(7); l1=x(8); l2=x(9);

[ivp,duration] = init();
Ycoordground=-0.198;
Xcoord=[0,l1*sin(phi1),l1*sin(phi1)+l2*sin(phi2)];
Ycoord=[0,-l1*cos(phi1),-l1*cos(phi1)-l2*cos(phi2)];

value = [Ycoordground-Ycoord(3)];
isterminal = [1]; % stop when any event occurs
direction = [0]; % regardless of slope

end

```