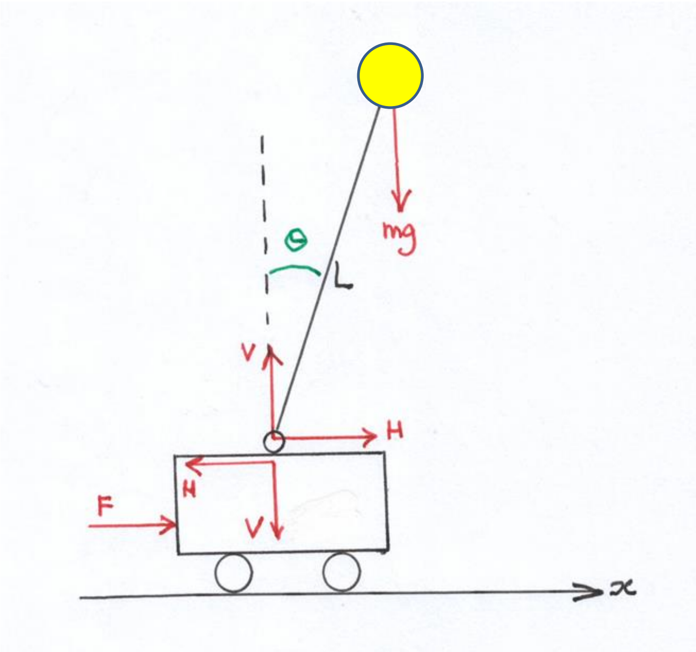
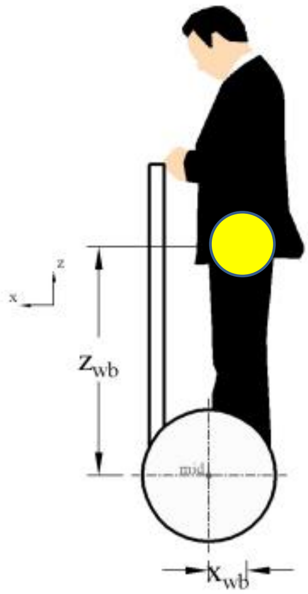
The Inverted Pendulum and Segway Funporter

CBPrice March 2021 Rev 03-05-21

Outline of the System





On the left is a picture of an inverted pendulum on a cart and on the right is a guy on a Segway. These two systems are identical where the yellow pendulum bob corresponds to the yellow centre of mass of the guy on the right.

Such systems are unstable unless they are controlled, they will fall over; the pendulum will rotate clockwise and settle with the bob underneath the cart. The guy will rotate either forwards or backwards and will end up on the ground if he is daft.

How are these systems made stable? Think about the inverted pendulum on the left. It will rotate clockwise unless the cart moves to the right and gets underneath it, then it is vertical. If it falls to the left then the cart moves to the left to get underneath it again. So, in general, the cart must move in the direction the bob is falling to get underneath it. The same idea applies to the guy on the Segway.

We must now develop the mathematical model of the inverted pendulum. Once we have that then we can code a simulation. We must derive two equations (i) for the rotation of the pendulum about its centre of mass (the bob at the top) and (ii) for the horizontal displacement of the cart. We know the routine; we have to get expressions for accelerations which give us velocities which give us displacements which we use to render our objects in UDK.

First we give the solution, then we explain how it is derived. For small motions around the vertical position $\theta = 0$ we obtain the two expressions which we must code. The first is the angular acceleration of the pendulum bob which we use to calculate its angular velocity, and hence the angle. In these expressions M is the mass of the cart and m is the mass of the pendulum bob.

$$\alpha = \frac{(M+m)g\theta - F}{ML}$$

Then we have the acceleration of the cart which we use to calculate its linear velocity and so its displacement.

$$a = \frac{-mg\theta + F}{M}$$

Remember our “coding chain”, knowing the acceleration allows us to update the velocity, and knowing the velocity allows us to update the displacement. When we have the displacement then we know how the object moves. In this case we have two displacements, the linear displacement of the cart, and the angular displacement of the pendulum on the cart.

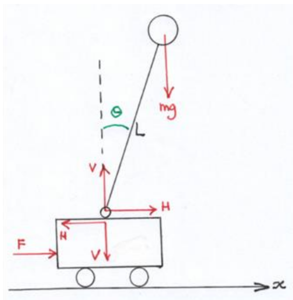
So from the angular acceleration we can calculate the rotation of the pendulum

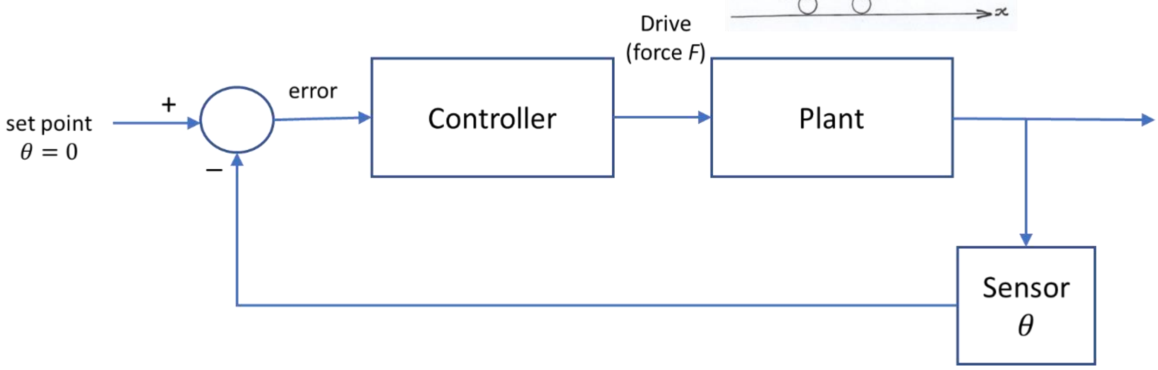
$$\Delta\omega = \alpha\Delta t$$
$$\Delta\theta = \omega\Delta t$$

and from the linear acceleration we can calculate the movement of the cart

$$\Delta v_x = a_x\Delta t$$
$$\Delta x = v_x\Delta t$$

Now let’s think about the control loop, here’s the diagram





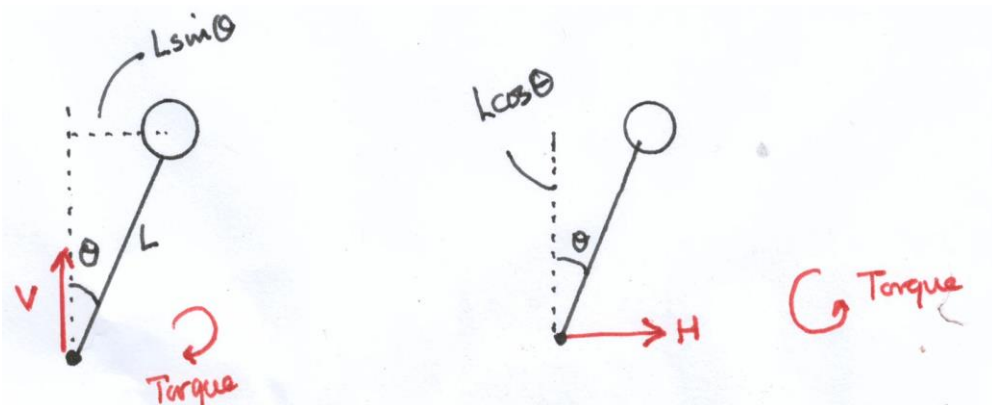
The set point angle is zero, we want the pendulum to be vertical. The sensor measures the angle of the pendulum and the controller takes the error and applies a force to the cart (by moving its wheels). It is interesting (and vital) to see how this force *F* appears in the above expression for linear acceleration (we expect this since the cart is being forced) but it also appears in the expression for the angular acceleration (unexpected perhaps, but reasonable, since the cart is joined to the pendulum).

Some Detail

First a reminder about the notation for translational and rotational movement:

Translation		Rotation	
Force	F	Torque	τ
Acceleration	a	Angular Acceleration	α
Velocity	v	Angular Velocity	ω
Displacement	z	Angle	θ
Mass	m	Moment of Inertia	I

Let's derive the equations we presented above. Consider the diagram below.



First the rotation about the centre of mass of the pendulum. The torque on the pendulum due to force V in the clockwise sense is calculated as force times the perpendicular distance to the centre of rotation.

$$VLsin\theta$$

The torque in the anticlockwise sense due to the horizontal force from the cart is

$$HLcos\theta$$

The total torque is the sum of these and produces angular acceleration α,

$$I\alpha = VLsin\theta - HLcos\theta$$

where I is the moment of inertia of the pendulum. When the pendulum is almost vertical (where it should be) then θ is small therefore $sin\theta \approx \theta$ and $cos\theta \approx 1$ so the above equation can be approximated to

$$I\alpha = VL\theta - HL$$

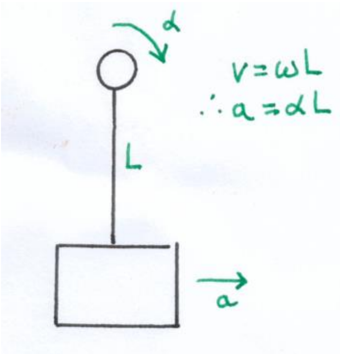
Since the mass it concentrated at the top of the pendulum, the centre of mass is located here and for this case we can assume that $I \approx 0$ so the above equation becomes

$$V\theta = H \qquad (1)$$

Now we need to consider the horizontal motion of the centre of mass of the pendulum. If the pendulum is almost vertical then we have

$$ma + mL\alpha = H \qquad (2)$$

The first term on the left is just the usual linear acceleration, but we must add on an extra acceleration because of the pendulum's rotation, that is the second term. Angular and linear acceleration are related by the expression $a = \alpha L$.



There is very little vertical motion since the pendulum is almost vertical, the vertical force is due to gravity

$$V = mg \qquad (3)$$

Finally we consider the horizontal motion of the cart

$$Ma = F - H \quad (4)$$

Now for a little algebra. From (1) and (3) we find that

$$H = mg\theta$$

and putting this into (4) we find the equation for linear acceleration is

$$Ma = F - mg\theta$$

Putting this into (2) we end up with

$$ML\alpha = (M + m)g\theta - F$$

To understand how these equations work, we solve them for the angular acceleration of the bob and the linear acceleration of the cart which result in the equations we code for the system dynamics.

$$\alpha = \frac{(M + m)}{Ml} g\theta - \frac{1}{Ml} F$$
$$a = -\frac{m}{M} g\theta + \frac{1}{M} F$$

Consider the first expression. The first term in this expression shows how gravity causes θ to increase, proportional to the value of θ, in other words the pendulum rotates away from its desired vertical position. On its own, it is unstable. The second term involving *F* is negative, in other words the applied force *F* causes θ to decrease, moving it to the desired vertical position, which is what we want. So our controller must provide *F* in this term.

Consider the second expression. The first term shows that the pendulum reduces the displacement *x* of the cart, but the force *F* in the second term causes *x* to increase, which is as expected. Together, these terms can reduce the acceleration of the cart to zero, so it achieves stability.

Enter the Controller

Now let's see what happens when we use a proportional controller to provide the force *F* based on the error between desired and actual angles. The proportional controller's control signal is defined by the usual expression

$$u = K_p(\theta_{des} - \theta)$$

which, since the desired angle is zero reduces to

$$u = -K_p\theta$$

If we used this control signal as the force, then the second term in the expression for angular acceleration would become positive, rendering the motion of the pendulum even more unstable, since this extra force would cause θ to increase even more. Also the expression for the cart's acceleration could never become zero, so the motion of the cart would be unstable. We therefore conclude that the applied force *F* should be the opposite of the control signal *u*.

Therefore, for the general PD controller described by the expression

$$u = K_p(\theta_{des} - \theta) + \ -K_d\omega$$

the applied force in the above equations should be

$$F = -u$$

Some Interesting (and important) Results.

1. Pendulum Stability for the Proportional Controller

If we put the expression for the proportional control into the expression for angular acceleration then we find

$$\alpha = \frac{(M+m)}{Ml}g\theta + \frac{1}{Ml}K_p(\theta_{des} - \theta)$$

Now let us consider a small change in angle $\Delta\theta$ and the corresponding change in angular acceleration $\Delta\alpha$

$$\Delta\alpha = \frac{(M+m)}{Ml} g\Delta\theta - \frac{1}{Ml} K_p \Delta\theta$$

Now assume the pendulum is at the top, and it starts to rotate, so $\Delta\theta > 0$. We want $\Delta\alpha < 0$ ie the pendulum accelerates in the opposite direction to cancel out the rotation. How can this happen? Well the first term is positive, so the angular acceleration increases, but the second term is negative, it is this term which restores the pendulum to the top.

So for stability, the second term must be larger than the first term, in other words

$$K_p > (M+m)g$$

This is an important result since it fixes the minimum value of K_p.

2. Relation between desired angle and the cart movement.

Here we consider the case where the desired angle $\theta_{des} > 0$, in other words we are forcing the pendulum to move away from the vertical. If the controller is working to make the pendulum stable at this angle then its angular acceleration is zero. Then taking the expression for the angular acceleration of the cart

$$a = -\frac{m}{M} g\theta + \frac{1}{M} K_p (\theta_{des} - \theta)$$

and combining this with the expression for zero angular acceleration

$$0 = \frac{(M+m)}{Ml} g\theta + \frac{1}{Ml} K_p (\theta_{des} - \theta)$$

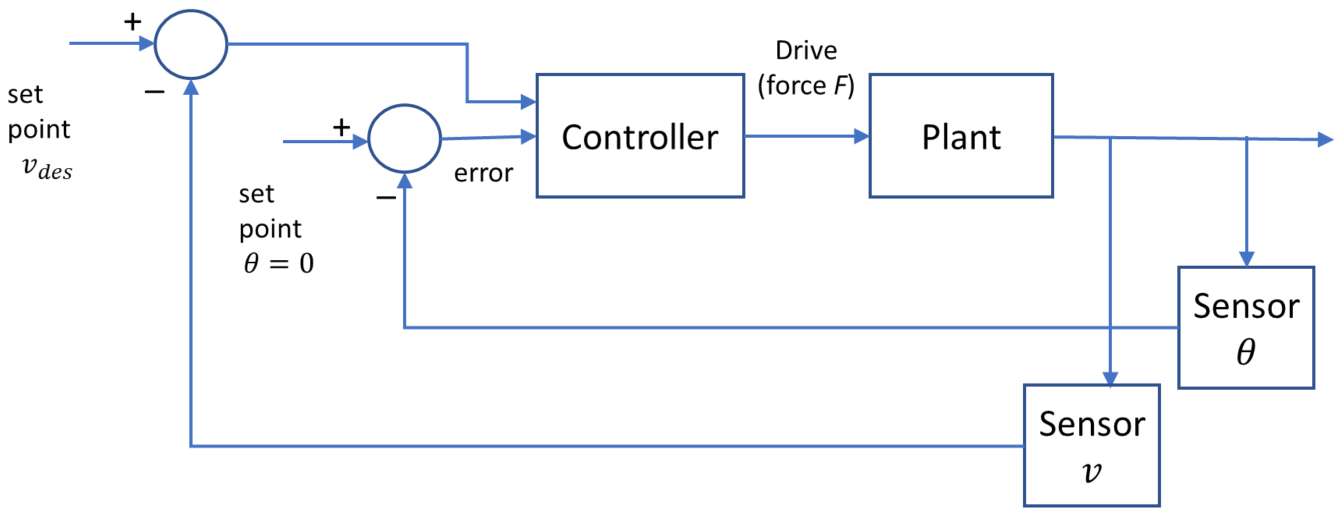
we find that the linear acceleration is $a = g\,\theta$ in general. If the controller is working quickly then $\theta = \theta_{des}$ so we have

$$a = g\theta_{des}$$

which is a useful expression which tells us that the linear acceleration is proportional to the amount we tilt the Segway. And here is a hidden devil ☢ Note that is no expression which relates the pendulum angle to the *speed* of the cart. So the cart can get any speed and, together with the pendulum, it may move off into the infinities of the universe, and we shall have to chase it there!

How to Control the Segway's velocity?

Well, here's the idea. We already have a controller which keeps the Segway upright. So we need a second controller to manage our desired speed. So we could 'wrap' the upright controller in the speed controller like this.



So we generate a second error signal for the controller from a velocity sensor attached to the Segway, and the set-point desired velocity.

Impulse Response

This is the response of the Segway to an input impulse to the controller. It can be obtained by setting the initial and desired thetas to zero and adding the following line of code to the function **computation()**

```
function computation(float dT) {
  if(time < dT) u = 1.0;
```

The theoretical response can be obtained by running the Octave script **Segway_Spectral.** The best way to compare the UDK and theoretical results is

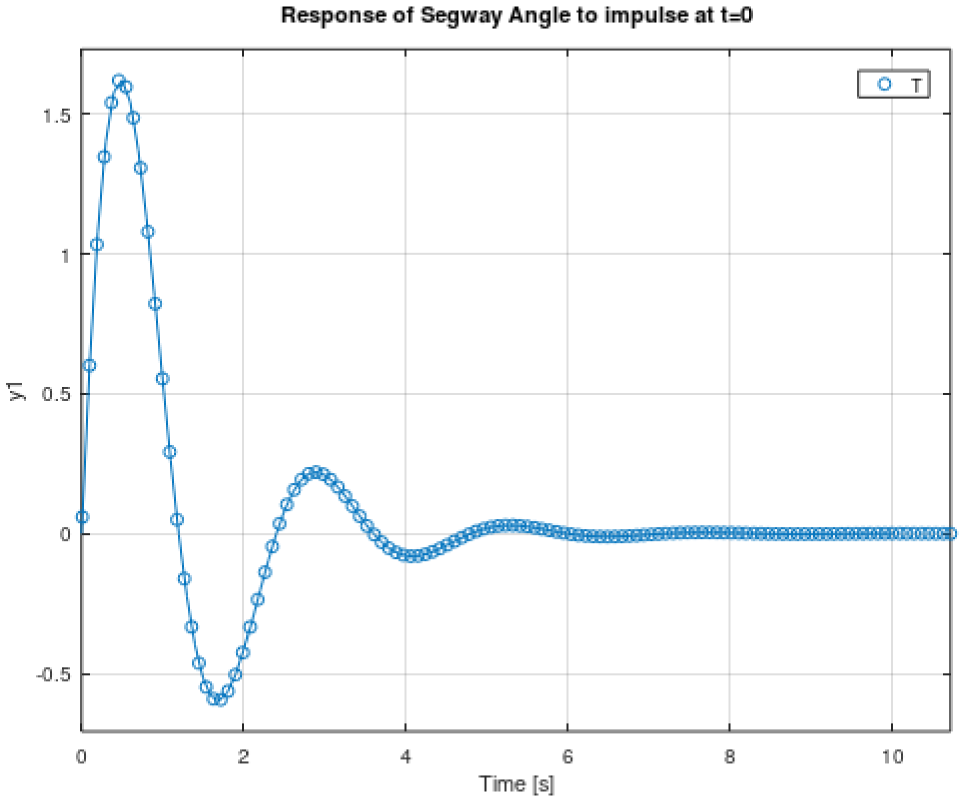
1) Run the UDK simulation and get a plot with 5 graphs.
2) Kill the plot window
3) Subsample your data like this

```
time1 = downsample(time,10);
omega1 = downsample(omega,10);
```

4) Plot your downsampled data

```
plot(time1,omega1,'o');
```

5) type **hold on**
6) Run the script **Segway_Special** inputting your values for Kp and Kd and a suitable end time. Here's what I got



You see the controller responds by driving the angular velocity of the pole which then makes the angle theta approach zero.

A couple of notes: We plot **omega** not **theta** (the graph title is incorrect) since this is what Octave returns to us. Also the cart will zoom off into the distance. This is normal when we gather impulse response data. Here's the Octave script

```
M = 0.5;
m = 0.2;
b = 0.0;
I = 0.000;
g = 9.81;
l = 0.3;

q = (M+m)*(I+m*l^2)-(m*l)^2;
s = tf('s');
P_pend = (m*l*s/q)/(s^3 + (b*(I + m*l^2))*s^2/q - ((M + m)*m*g*l)*s/q - b*m*g*l/q);

Ki = 0;
Kp = input('Input Kp ');
Kd = input('Input Kd ');
C = pid(Kp,Ki,Kd);
T = feedback(P_pend,C);
tEnd = input('End time ');
t=0:0.1:tEnd;
impulse(T,t);
```