Segway Modelling and Control

D.A. Toapanta
University of Twente
S2074036

F.H. Alammal
University of Twente
s2058502

***Abstract*—This report is on the modelling and control of a Segway system. The system is first modelled using the bond graph method. A controller is then built to control the angle and velocity of the Segway body. Finally, the modelled system and controller behavior are simulated and uploaded into the Segway to test for performance.**

I. Introduction

The Segway is a modern mean of transportation centered upon the device's ability to balance. A smaller version of the Segway is used in this project, which is made by the RAM department of the University of Twente. This project is made up of two parts. The first part being the motor modelling and control of the wheel orientation. The second part addresses the rest of the Segway body.

This report is on the second part of the project which aims to model the rest of the Segway, namely the body and wheels. After modelling, the control part of the project is addressed, whereupon controllers for the Segway are designed to meet the specifications given and complete a few challenges. To achieve the goal intended, applications such as 20sim and its sub program 20sim 4C are used to model, simulate, and test the design, in conjunction with Matlab to assist in processing data.

II. Analysis

A. Motors

The first part of the project tackled the modelling of the motors in the Segway, and the relevant environmental contributions to the system. This included the electrical and mechanical domains. The final model of the motor can be found below.

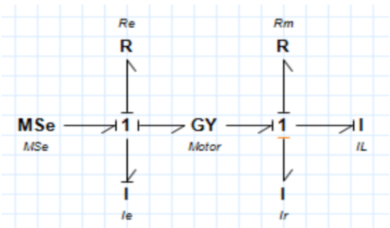

Figure 1 Final model in part 1 of the project

The elements modelled had unknown parameter values which needed to be characterized to obtain a realistic behavior of the physical system. These values were derived systematically through a series of experiments and processing of data.

The parameter values obtained are found in table I.

TABLE I. Obtained parameters

Parameters of motor model					
Re	***Rm***	***Ie***	***Km***	***Ir***	***I_L***
14.57 Ω	4.88E-15	4.29E-02 H	0.046 Nm/A	1.40E-05 kgm^2	8.25E-05 kgm^2

B. Modeling the Segway

This section discusses the modelling procedure of the Segway body and wheels using the bond graph method. Modelling the Segway will take place in the mechanical rotational and translational domain. The model will be described in three parts first, namely the body, the wheel, and the hinge point. These parts will then be joined in one bond graph to form the final model.

The Segway may be modelled as a 3D object, however for this project a 2D model makes the procedure easier while still having a sufficiently accurate representation of the real behavior of the Segway.

The entire Segway may be split into two separate parts joined through the hinge point. The grey spots represent the center of masses for the Segway body and wheel. The blue dot is the hinge point between the two objects.

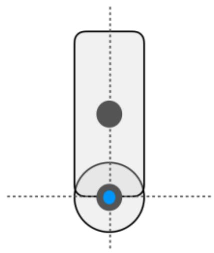

Figure 2 Side view of Segway

1) Segway Body

- Identifying Velocities

To model the body, an identification of the points of interest is required. In the body, the points of interest are the center of mass COM and the hinge point. The velocities to be described by this point are to be identified next. The velocities mentioned below can be seen in figure 3.

Starting in the rotational domain, it is important to note that the body will rotate when the Segway moves using its wheels, or when being pushed by hand for example. The rotation about the pivot point can be modelled. A 1 junction representing the angular velocity of the body is constructed, labelled by ω_body.

The Segway body will also undergo linear velocity, which can be viewed from a body fixed frame or an inertial frame. Therefore, the second velocity is a translational velocity in body fixed frame, identified as $v_{\_body\_BF}$. The third 1-junction represents the last velocity known as $v_{\_body\_IF}$. This velocity is acting in the inertial frame.

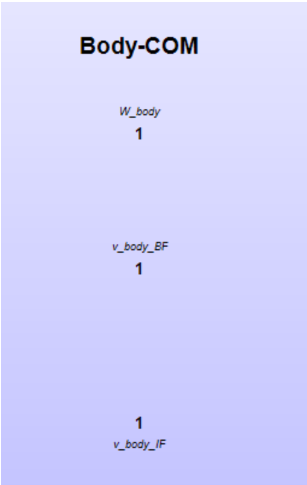

Figure 3 Identified velocities in the Segway body

- Connecting the Velocities

To connect $v_{\_body\_BF}$ and $v_{\_body\_IF}$, a rotational transformer RTF is used. The RTF is characterized by a rotational matrix $\boldsymbol{T}(\varphi)$ which is dependent on an angle φ to modulate the RTF.

$$\boldsymbol{T}(\varphi) = \begin{bmatrix} cos\varphi & -sin\varphi \\ sin\varphi & cos\varphi \end{bmatrix} \quad [1]$$

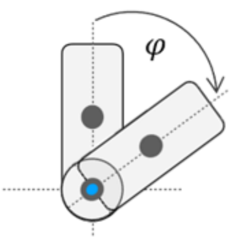

Figure 4 Rotational motion of the Segway body showing the angle used for the rotational matrix

This matrix maps the polar coordinates used to describe motion in the body fixed frame to cartesian coordinates in the inertial frame. This mapping uses the angle φ given by the Segway's rotation about the hinge point. This can be obtained by integrating the common flow of the angular velocity of the Segway's body, as is represented by figure 5.

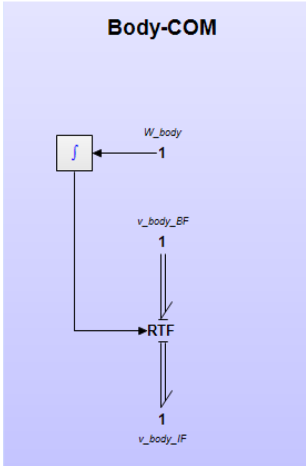

Figure 5 Connecting the Segway velocities to each other

- Connecting Elements

To proceed in modelling the body, some elements are identified and connected to the junctions in figure 6.

The first being an I-element representing the moment of inertia of the Segway's body. If the Segway is turned upside down and held only from the wheels, the body will remain at rest if no force is acting upon it. If a force is applied it will oscillate, thus encompassing a moment of inertia. This element will be acting in the rotational frame, and so is connected to $\omega_{\_body}$. The I element is named J_{body}.

To the same 1-junction, an R element will be connected. This R element represents the air resistance acting on the Segway's body. Although the contribution of air resistance in minimal, it is essential to include it due to its ability to model the damping of a Segway when considered as a pendulum. Without air resistance, a pendulum in harmonic motion will keep oscillating if no other forces act on it.

Another I element can be modeled in the Segway's body, and that is the mass of the Segway. This value will include the body's frame, motors, axle, and all the internal components. This mass is acting in the inertial frame and will be connected to $v_{\_body\_IF}$.

Furthermore, the effect of gravity is modelled in the inertial frame using a source of effort Se. Since gravity is acting in the y-axis only, the Se component will be characterized by a two-element vector with a zero value in the horizontal component and the mass multiplied by the acceleration of freefall ($g = 9.81 m/s^2$) in the negative vertical direction. This element will be connected to $v_{\_body\_IF}$.

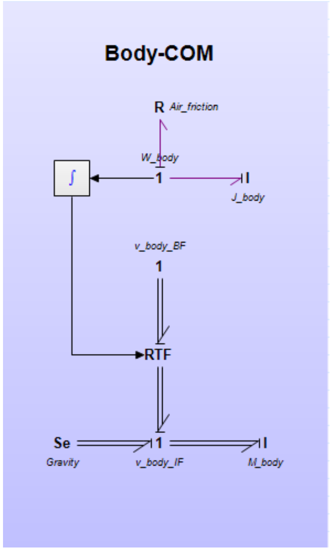

Figure 6 Full model of the Segway's body

The Segway body is now complete, and the hinge-point and wheel will be modelled in the next two sections.

2) Hinge-point

The Segway body is directly connected to the hinge point, which is the second point of interest. This point will be modelled by identifying the velocities acting on it.

The hinge point will have two velocities; one acting with respect to its own frame, called $v_{\_hinge\_BF}$, and another in the inertial frame, given by $v_{\_hinge\_IF}$.

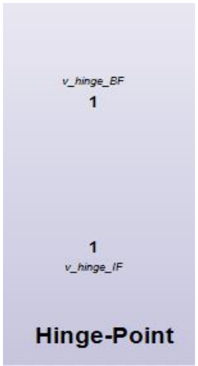

Figure 7 Hinge-point velocities

To relate the two velocities, an RTF with the same matrix $\boldsymbol{T}(\varphi)$ that is dependent on the same angle φ is used. The resulting sub-model can be found in figure 8.

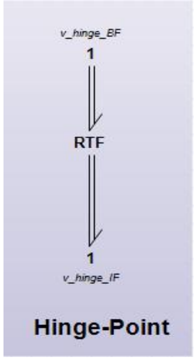

Figure 8 Full hinge-point sub-model

Before connecting the body to the hinge, the idea of a tangential velocity is discussed.

A relationship between the angular velocity and the tangential velocity of a rotating circular object, can be found by differentiating the formula for arc length with respect to time. Equation 2 is differentiated to obtain equation 3.

$$l = \theta r \quad [2]$$

$$v_{tan} = \omega r \quad [3]$$

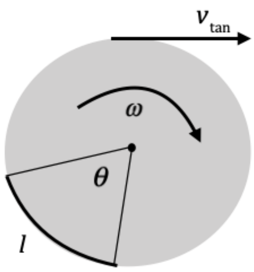

Figure 9 Reference diagram for equations 2 and 3

Accordingly, the tangential velocity of the Segway's body rotating about the COM is given by $v_{tan} = (d)(\omega_{body})$, where d is the distance between the pivot and the center of mass. This finding will be used later to connect the body to the hinge point.

When considering the hinge point as the point of interest, an expression for the translational velocity in body fixed frame $v_{\_hinge\_BF}$ can be found. This translational velocity will have a horizontal and a vertical component. Each of which consists of two terms; the translational velocity of the body in body fixed frame $v_{\_body\_BF}$ and the tangential velocity of the body around its center of mass $v_{\_body\_tan}$.

$$v_{\_hinge\_BF} = v_{\_body\_BF} + v_{\_body\_tan} \quad [4]$$

Expanding this expression into a system of equations in the horizontal and vertical component gives

$$v_{\_hinge\_BFx} = v_{\_body\_BF} + (y)\omega_{body} \quad [5]$$

$$v_{\_hinge\_BFy} = v_{\_body\_BF} + (x)\omega_{body} \qquad [6]$$

where the coefficients of ω_{body} can be represented using the vector $\begin{bmatrix} y \\ x \end{bmatrix}$. This vector represents the coordinates of the pivot point where x is the horizontal distance and y is the vertical distance from the center of mass of the Segway; that is d. An ideal case is assumed, that the pivot point is located exactly down the middle of the width of the Segway. The vector is then equal to $\begin{bmatrix} d \\ 0 \end{bmatrix}$.

The system of equations 5 and 6 can be modelled by a 0-junction, as can be seen in the following figure.

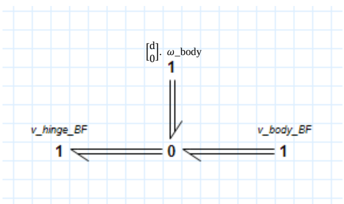

Figure 10 Connection piece between the body and hinge point sub-models

This junction may be used to connect the body and the hinge-point sub-models with the assistance of a TF is characterized by the vector $\begin{bmatrix} d \\ 0 \end{bmatrix}$. This connection is implemented in figure 11.

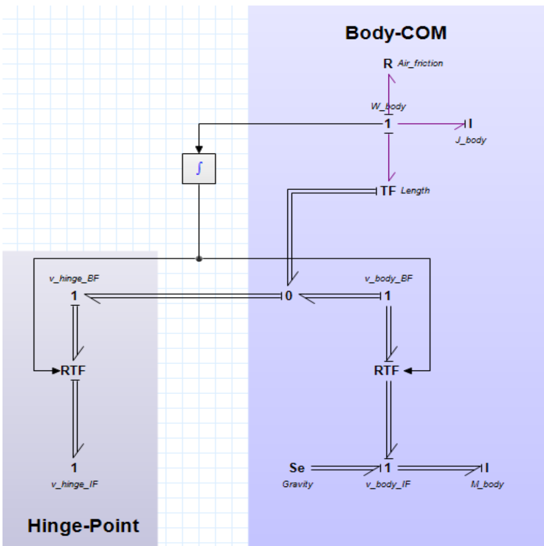

Figure 11 Connection between the Segway body and hinge-point

Note that the angle φ coming out of the integrator output has been connected as the input to the RTF of the hinge-point.

3) Segway Wheels

The next part that needs to be modelled is the wheel of the Segway.

- Identifying Velocities

Much like the modelling procedure of the body, the points of interest and the velocities in the wheel may be identified first. There is only one point of interest in this case; the COM of the wheel. That is because the COM aligns with the pivot of the wheel.

The wheel will have an angular velocity $\omega_{\_wheel}$ about the pivot point. A 1-junction is used to represent it in the rotational frame. This angular velocity will cause translational displacement and thus velocity when the wheel is in contact with the ground. The translational velocity caused by the wheel is modelled by a 1-junction.

As the wheel will only move horizontally this velocity will be a component in the horizontal direction, called $Vx_{\_wheel}$.

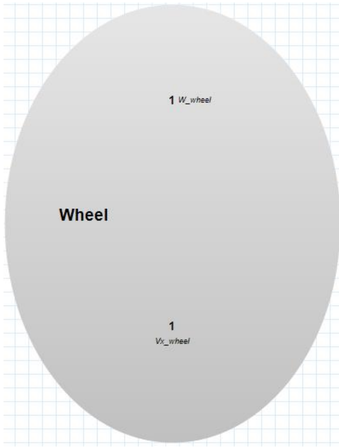

Figure 12 Modelling the velocities in the wheel

- Connecting Velocities

Between $\omega_{\_wheel}$ in the rotational frame and $Vx_{\_wheel}$ in inertial frame, a TF element with a transformer ratio equal to the radius of the wheel is used. This value is used due to the relationship between the angular and tangential velocity found in equation 3. The input to the TF is then $\omega_{\_wheel}$, while the output is $Vx_{\_wheel}$.

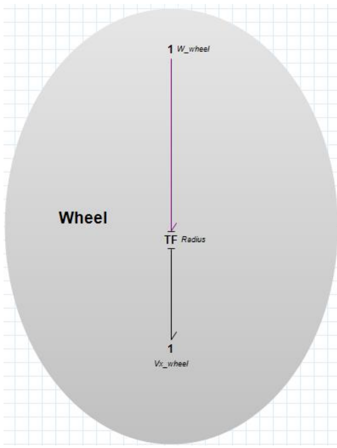

Figure 13 Velocities in the wheel connected by a transformer

Now that the tangential velocity is obtained, it is important to note that this velocity only takes place in the horizontal direction. A constraint needs to be modelled satisfying the condition that the Segway will not move upwards or downwards. A demultiplexer is used to connect the horizontal component of wheel's velocity in inertial frame, while the vertical component is set to zero using a flow source Sf. This connection is shown in figure 14.

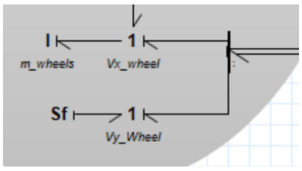

Figure 14 Use of a demultiplexer to set a constraint in the vertical direction

- Connecting Elements

The elements that need to be connected to the wheel include the moment of inertia of the wheel and the mass of the wheel. Both will be represented with an I element. The moment of inertia of the wheel, called J$_{wheel}$ is modelled in the rotational frame. The mass of the wheel, however, will be modelled in the inertial frame, and is known as m$_{wheels}$.

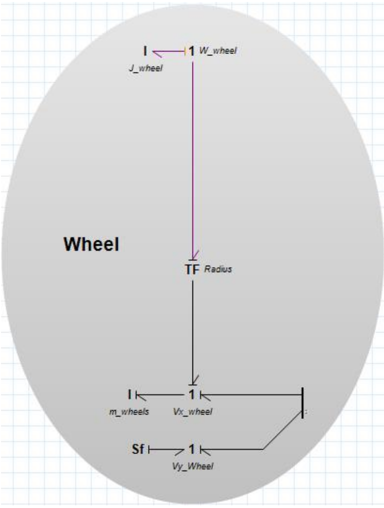

Figure 15 Final model of the wheel

4) Sensors

To finally complete the modeling part, the sensors included in the Segway are also modeled.

The first sensor is the gyro and is formed by creating an output signal from ω_body. Next the Accelerometer is modeled by taking the integrate of ω_body and then passing it through a cosine and a sine function such as the result signals are assigned the names z_accelerometer and y_accelerometer. These names indicate the position of the Segway when it was first placed in the floor as shown in figure 16 top Left and when it was positioned standing upright as indicated in figure 16 bottom left

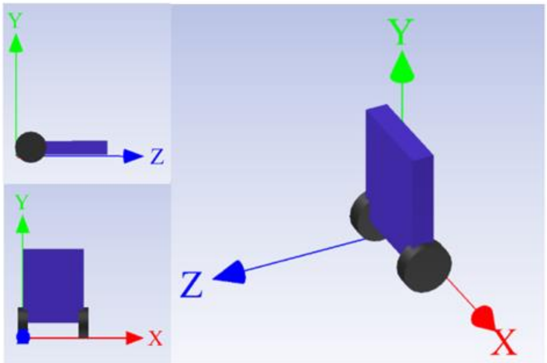

Figure 16 Model of the Segway in 2D and 3D frame reference

Likewise, the position of the wheels and the body in the inertial frame can be modeled using an integrator and taking the 1-junction velocities respectively for each case. The resulting signals are y_position body, z_position body and Wheel_position as the latter one only moves in one direction, namely Z direction as it can be seen from the picture above.

5) Final Model

Finally, a complete version of the Segway with all its components can be modeled. The following figure is shown as a guideline of how the whole system looks like but because of the size page a full version of this model is included in the appendix section.

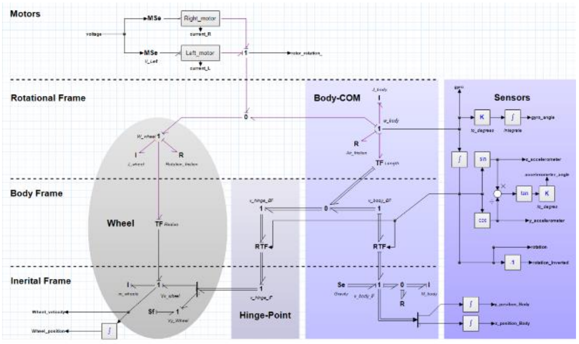

Figure 17 Final model of the Segway

III. Methodology

This section provides the method of obtaining the parameters to characterize the model derived previously. To achieve that, a brief overview on the sensors and the orientation of the axes used to do the experiments is given first. Proceeding this is an explanation of how the controllers were designed to accomplish the initially stated goal.

A. Sensors

1) IMU

The IMU (Inertial Measurement Unit) is a small chip (BMX055) fixed approximately at the center of mass of the Segway body. It contains both an accelerometer and a gyroscope. It will be used to measure the angle of the Segway.

- *Axes Orientation*

To use the IMU, the orientation of axes on the Segway must first be identified. In achieving this, 20sim 4C was connected to the Segway to receive data from the sensors.

The accelerometer was used to find the relevant axes by placing the Segway in different orientations on a table. The three different axes X, Y and Z were then viewed at the same time. The axis with the value -9.81m/s^2 was taken to be the one for the corresponding orientation. For the accelerometer this happened to be the Z and Y axes as the horizontal and vertical axes respectively. Note that the horizontal axis is taken to be parallel to the ground.

In the gyroscope, a similar procedure was done to find the axis at which the relevant angle had to be taken. This happened to be X-axis.

Finally, a mapping for each sensor was done. This is shown in table II and the corresponding axes shown in figure 18. The number seventeen there stands for the number of Segway that belong to this group and is was taken as the front of the Segway as the reference for the 3D modeling section down below.

TABLE II. MAPPING FOR EACH SENSOR

Sensor	Accelerometer	Gyro
Range	[-1,1]	[-1,1]
Maps to	[-2g , 2g] m/s^2	[-500,500] °/s

B. 3D Modelling

To validate the simulated behavior of the model, a 3D animation was constructed. The shape of the Segway was recreated by placing a cuboid to mimic the body, and two cylinders to mimic the shape of the wheels. These objects were positioned to create the whole Segway. The Segway was then given an offset in the horizontal direction to have a better view of the motion.

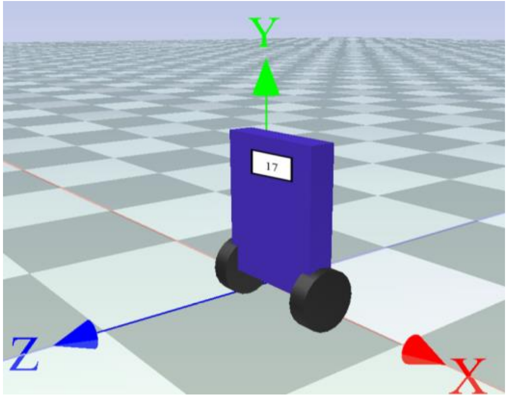

Figure 18 Segway standing upright in the floor

C. Characterisation

1) Moments of Inertia

There exists two moment of inertias in the model. The moment of inertia of the wheel and the one belonging to the body.

An experiment was carried to find the moment of inertia of the body. The value was then calculated in two ways. Firstly, the Segway was turned upside down and with the wheels removed, the axle was placed on two wooden planks at each side, supported by two tables. A string was secured at the axle and was left to hang vertically downwards. At the bottom end of the string, two masses were hung to produce a straight line falling downwards towards gravity. On the side of the Segway a piece of tape was placed vertically. A line was drawn down the middle, using the string as reference.

Figure 19 Experiment setup to find the moment of inertia of the Segway body

The Segway was given a push by hand to instigate harmonic oscillation. A video of the oscillation was then recorded where each time the line drawn on the Segway overlapped the string the time was noted. The time period between each overlap was taken as the period. This turn out to be 0.82s.

Figure 20 Segway body treated as a pendulum demonstrating oscillation

Another way to calculate the period was achieved by recording the angular velocity of the Segway's body using the gyroscope. The signal produced was saved and used to record the total time taken for 8 oscillations.

The resultant time was then divided by the number of oscillations to obtain the average period. The period found here is 0.81s. The two values were used to calculate the average to find the period of the Segway's oscillation acting as a pendulum.

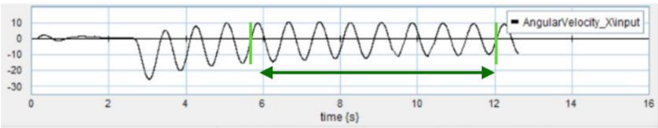

Figure 21 Screenshot of angular velocity signal using gyro sensor and 20sim4C

The formula for the period of a physical pendulum in equation 7 can be used to obtain the moment of inertia of a pendulum whose pivot is a distance d from the center of mass. In the same equation, M is the mass of the Segway body, g is the acceleration of freefall, and T is the period calculated previously.

$$T = 2\pi\sqrt{\frac{I}{Mgd}} \qquad [7]$$

Using the parallel axis theorem, it can be obtained that I is the sum of the moment of inertia at the center of mass and the term Md^2 .

$$I = I_{COM} + Md^2 \qquad [8]$$

Since the moment of inertia modelled in the bond graph is the one about the center of mass, equation 8 will be used to obtain this value.

As for the moment of inertia related to the wheel, it was calculated using two methods as well. The first method was done in different variations. This required the use of a ramp that was sectioned in 5cm intervals. Refer to figure 22 for clarification.

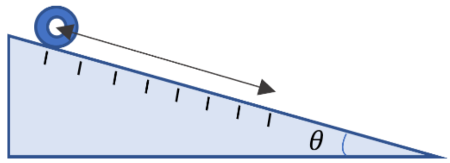

Figure 22 Reference diagram for ramp experiment to calculate the moment of inertia of the wheel

The wheel was placed carefully on the top of the ramp and was let go to freely roll down to the end. In the meantime, a video of this action was recorded, and a video editor was used to freeze the time at each 5 cm interval to record different points in time. See appendix for further information.

The curve fitting option `>> pcoeff=polyfit(time,position,2);` in Matlab is then used to produce a replica of the position of the wheel versus time. The resulting graph is obtained:

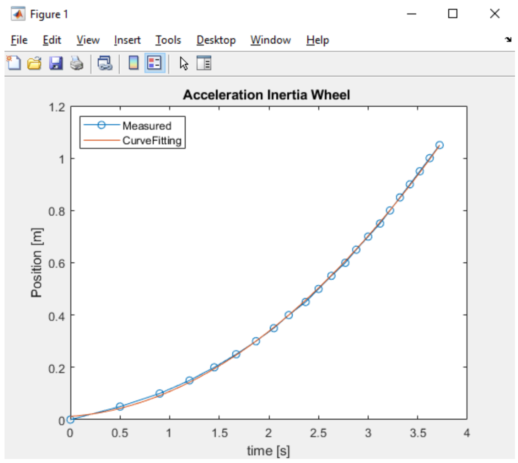

Figure 23 Curve fitting of position vs time of Segway wheel along a ramp

A short code for this was written in Matlab which can be found in Appendix A. This code produces a quadratic equation for the position x representing the wheel at a distance L with respect to the beginning on top of the ramp. The equation found is:

$$0.3434t^2 + 0.0084t + 0.0122 \qquad [9]$$

which is comparable to one of the kinematic equations of motion,

$$\frac{1}{2}at^2 + vt + x \qquad [10]$$

The value for acceleration a can be obtained by equating the coefficients of t^2 and solving for a. This value is needed for finding the moment of inertia.

By analyzing the net torque applied on the wheel as it is placed on the ramp, equation 11 describing the moment of inertia is derived,

$$I = r^2\left(\frac{mg}{a}\sin(\theta) - m\right) \qquad [11]$$

where r is radius of the wheel, m is the mass of the wheel, g is the acceleration of freefall, a is the acceleration of the rolling wheel, and θ is the angle formed by the ramp with the ground.

The radius of the wheel can be measured with a regular ruler. The angle is obtained by simple trigonometry; by finding the arcsine of the ratio of the height of the ramp with the length of the ramp itself. The mass is measured with a scale, and a is the acceleration obtained previously. Now that all the variables in equation 11 are known, the inertia of the wheel is calculated.

This experiment was done for different ramp steepness, and different ramp lengths to attain values for the inertia in different situations.

The second way to obtain a value for the moment of inertia of the wheel was done using the formula for inertia of a hollow cylinder. The formula used can be found in equation 12

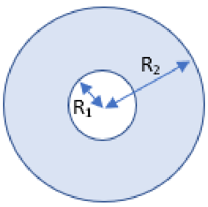

Figure 24 Cross-sectional area of a hollow cylinder

$$I = \frac{1}{2}m({R_1}^2 + {R_2}^2) \qquad [12]$$

Where R_1 and R_2 are the interior and exterior radii, and m is the mass of the wheel. The inertia of a hollow disk does not depend on the depth of the wheel, thus only a cross-sectional area is shown in reference figure 24.

The values found from both methods were averaged out to obtain the moment of inertia of the wheel.

2) Masses

Two masses need two be included in the bond graph model; the mass of the Segway body, and that of the wheels. A scale was used in both cases to measure the mass, however for the wheel, a higher precision (0.01g) scale was used since it was a smaller mass. This is shown in the following figure.

Figure 25 Segway wheel on a higher precision scale

3) Air resistance

To obtain a value for the modelled air resistance, a signal from the gyroscope sensor is recorded after giving the Segway a slight push by hand while hung upside down. This signal produced is an oscillation that is recorded and used as a reference to obtain a value for the air resistance using the curve fitting feature in 20sim.

D. Results

All the experiments discussed before results in the following values for the modeling section:

Parameters of motor model			
M_body	***m_wheels***	***TF*** ***wheels***	***R*** ***air friction***
2.216 kg	0.2870 kg	0.04 m	2.09E-02 kgm/s^2
J_body	***J_wheels***	***TF*** ***body***	***Angle*** ***body_offset***
1.25E-02 kgm^2	2.82E-04 kgm^2	$\begin{bmatrix}0.11\\0\end{bmatrix}$ m	0.05 rad

IV. CONTROL

To reach the goal proposed at the beginning of this report, the Segway model need to be controlled. The previous sections covered the modeling of this system when no control was implemented. The Segway in reality undergoes the force of gravity among other forces that make the system fall down to the earth. This can be easily proved by the system when no motors are working and when the initial condition of the Segway is standing upright.

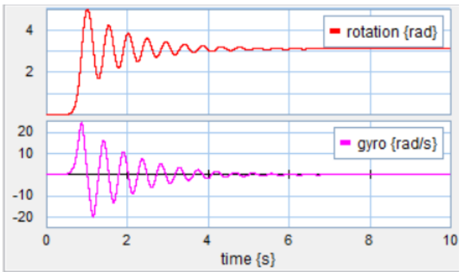

Figure 26 Segway falling down

Figure 26 on the top shows the rotation of the Segway with respect to the axis Y and the angle φ described in section 1 of the modeling part. The plot down below shows the working of the gyro sensor when a change in velocity is experienced. As it can be seen from figure 26, the system should be standing with an angle of zero rad and with an angular velocity of zero. This is clearly not the case for which a control system needs to be implemented.

The first question that should be solved is how to use the power of the motors to keep the Segway to an angle of zero rad with respect to Y. The next part of the controller should be in charge of controlling the velocity of the Segway when the motors are working. Lastly but not least in order to accomplish the challenges proposed at the beginning of this paper, the Segway needs to be responsive to disturbances and e.g. when the user wants to move the Segway to his own willing.
All these guidelines are covered in this section and are subdivided in three subsections, namely: angle controller, velocity controller, and user controller

A. *Angle Controller*

To be able to control the angle that the Segway experiences when is let go, a robust angle sensor needs to be created.
This can be easily done if one realizes the properties of the IMU chip integrated in the Segway. The first being: the gyro capability to measure changes of angular position when the Segway is rotated around the X axis. The second property from which the IMU sensor may be beneficial to get φ is by using the signals measured by the accelerometer in the Y and Z direction. In other worlds relating to figure 16. The gyro sensor will measure a change if for example the Segway is rotated around the X axis from Y to Z when is about to fall and the accelerometer sensor will measure the components of where the Segway is located in real time with respect to Y and Z.
Now with these signals in the case of the gyro, one can integrate the change of angular position to get φ. Similarly, taking the components Z and Y of the accelerometer and passing them through an arctan function, one can get phi.

The procedure mentioned previously allows the calculation of φ. Nonetheless, the gyro sensor works with high frequencies which need to be filtered out by a high pass filter. In contrast, the signals from the accelerometer work with low frequencies to which a low pass filter should be applied. All together if one would like to get an accurate value close to reality of the angle experienced by the body Segway when is falling down a complementary filter must be used. The idea behind this is to incorporate the best result from each sensor

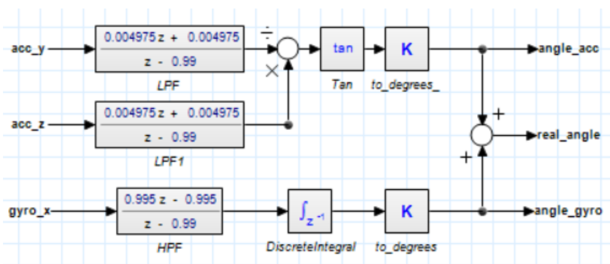

Figure 27 Complementary filter for 4C

Figure 27 shows this implementation in 20 sim. There, LPF, LPF1 and HPF, are the transfer functions in Z domain of a low pass filter and a high pass filter respectively.
The next step is to validate this design with actual Segway.

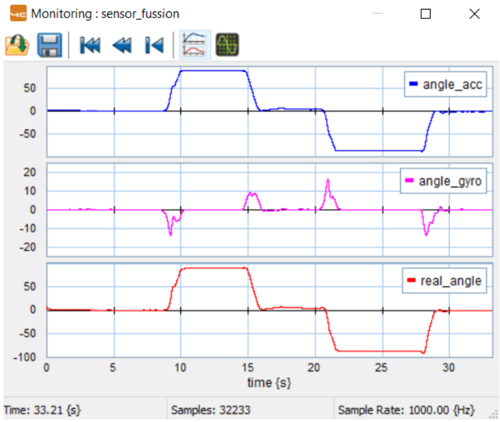

Figure 28 Segway angle sensors controller
Note: The units for the Y axis in this figure are in degrees

Figure 28 shows the output signal of figure 27 When the Segway was held upright, in 90 degrees with respect the floor and -90 degrees. At the moment of testing, the Segway was stressed by abrupt moments whit positive results reason for which the second part of the angle controller is explained below.

Once phi can be measured a transfer function relating the power of the motors and the rotation -phi- of the body Segway can be extracted. The input of such a transfer function is the motor signal which input a reference to the modulated Source of effort, MSe to the motors of the Segway and phi called rotation in the final model. By doing so, is it found an unstable pole in the positive real part as shown in the figure below

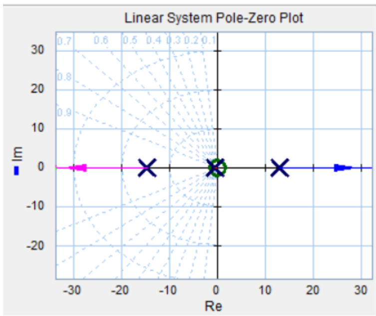

Figure 29 Close look at TF motor - rotation Pole-Zero plot

Figure 29 not only shows the instability of the plant motor-rotation but also shows that to any gain in a negative feedback loop system the close loop system will remain unstable. That is why sisotool in Matlab is used to find a better controller.

The controller suitable to do the job has to be as less complex as possible to avoid compensation in the bandwidth of the system. That is when PID controllers come in handy. They can be as simple as one desire without involucrate phase margin in the close loop transfer function [2]. Next the process to find a suitable configuration such that the new pole configuration of the root locus in Figure 29 shifts to the right and thus be stable. The procedure behind this is rather easy and the automated tool in Matlabb called pidTunner() was used to carried this action out. What is important in this step is to consider the time response of the system when a step function is set as the reference.

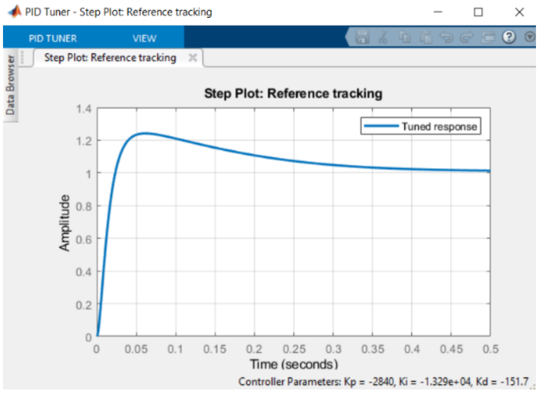

Figure 30 Step response of plant motor-rotation of Segway model with PID controller

Figure 30 shows the impulse response after adding the PID controller and as it can be seen the time response is of about 0.25s which is considered to be fast in practical terms when the motors of the Segway would have to use voltage to compensate the angle of the Segway and keep it to zero rad. The values shown below in figure 30 are then copied to the angle controller after creating a negative feedback loop with the proper signals discussed before. See figure 31 to further clarification

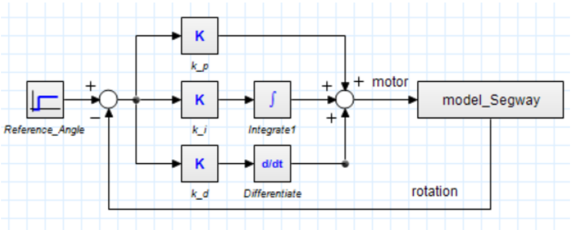

Figure 31 Implementation Angle Controller

Next the validation and performance need of this controlled need to be tested. First the implementation to 4C is created. This consist of the PID controller that was just explained, the integration of the complementary filter and a couple of monitor signals to keep track of the performance of the system. Figure 32 shows the final result plus a limiter from -1 to 1 in case of overshooting and a switch break such that if the angle measured by the gyro is more than ± 20 the signals going to the motor are shut down. By doing this, safety is ensured, and the motors are not burn out.

The final validation is shown in figure 33 and 34 and as it can be seen the results are positive reason for which to ensure that the Segway will not move by standing in one place a velocity controller is designed in the next section

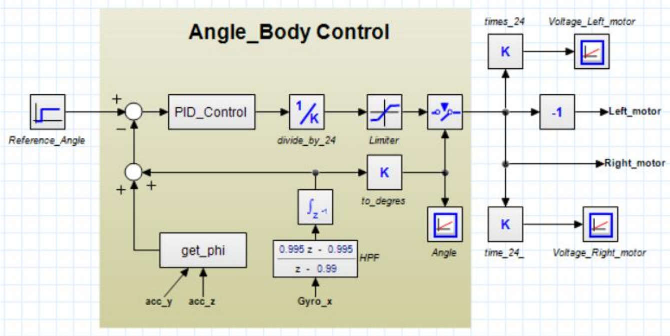

Figure 32 Implementation Angle controller 4C

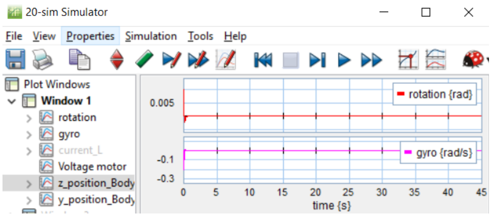

Figure 33 Validation Angle controller simulation

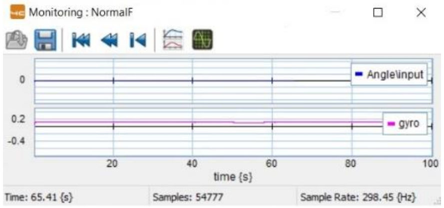

Figure 34 Validation Angle Controller real life

Note: The units in the Y axis in this figure are: rad for Angle/input and rad/s for gyro

B. *Velocity Controller*

Now that the angle of the body Segway with respect to Y has been controlled, to ensure e.g. that the Segway will remain in the same place while standing in an inclined floor as in a ramp a velocity controller will be designed. The procedure to accomplish this is as follows.

1. Use the linearization tool in 20 sim to get a transfer function from the Reference Angle of the previous controller to the signal Wheel_velocity of the 1-junction describing the tangential velocity of the wheels in the final model depicted in the Appendix of this paper.
2. Analyze the root locus of the plant created and investigate whether a gain is sufficient to stabilize the system
   2.1 If the previous step did not work out use pidTunner in Matlab and get a suitable controller
3. Implement the controller in a negative feedback loop
4. Validate the controller just created in simulations as well as in real life

The procedure above intends to summarize the steps considered to design a velocity controller. In step one, the transfer function is found to be of degree five and in step two the root locus of the pole_zero plot shows that a gain of 8.92E-03 is enough to make the system stable. Figure 35 and 36 shows the open loop transfer function and its corresponding zero_pole plot respectively

$$\frac{113\,s^3 + 3.853e{+}004\,s^2 + 4.243e{+}004\,s - 7.323e{+}006}{s^5 + 680.5\,s^4 + 1.163e{+}005\,s^3 + 1.102e{+}005\,s^2 - 2.19e{+}007\,s - 8.562e{+}006}$$

Figure 35 Open loop TF plant Reference_Angle - Wheel_velocity

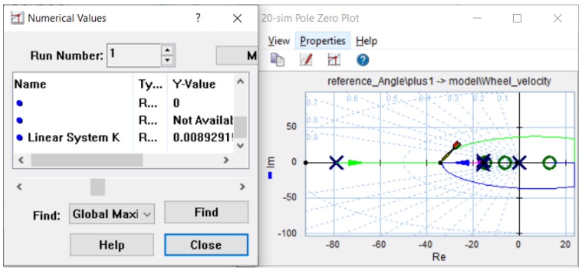

Figure 36 Root locus of figure 35

Although Figure 36 shows that a gain is sufficient to make the close loop transfer function stable when two controllers are implemented in a cascade fashion as is intended in this case, the inner loop response has to highly overpass the time response of the outer loop, the velocity controller in this case. Using Matlab again to analyze the step response of the whole system it is found that such a gain causes instability. Increasing the gain but this time greater than 8.92E-03 shows still an unstable system

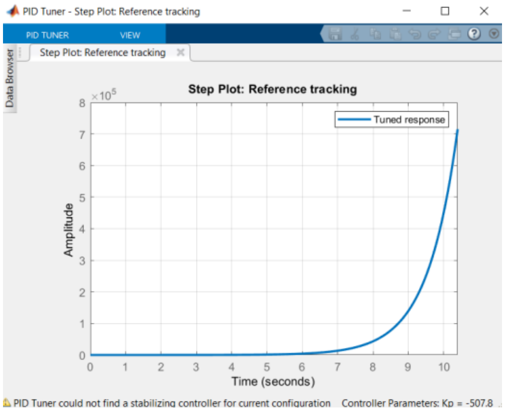

Figure 37 Unstable system with Kp greater than 8.92E-03

Therefore step 2.1 is proceed and a PID controller is once again designed. The whole system after implementing both controllers now looks like this:

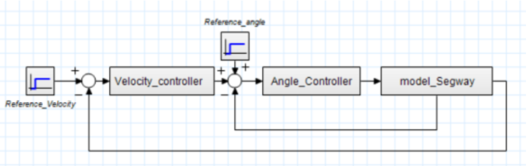

Figure 38 Velocity controller in cascade with Angle controller

The final step is the implementation in 4C to be able to test the performance of the Segway. In the left of Figure 39, the controlled created in step 3 is recreated but this time in z-domain. To obtain the signal that it was used in step 1, Wheels_velocity, one must use an encoder provided to analyze the angular displacement of the wheels. This is then, passed through a discrete derivate as indicated in the picture and then using equation 3, the radius of the wheel is multiplicated to obtain the velocity tangential. The same process applies for the other wheel and the summation of both wheel takin into account that one wheel rotates in the opposite direction is averaged by the attenuator indicated in the figure as well.

This sets the negative feedback loop an in this way a cascade of controllers is created.

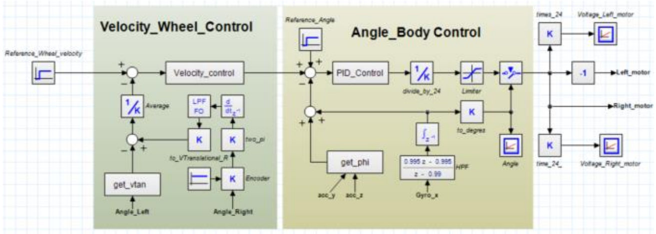

Figure 39 Velocity and Angle Controller in Cascade in 4C

Now the performance of this controlled is tested and the results of simulations and real life are compared

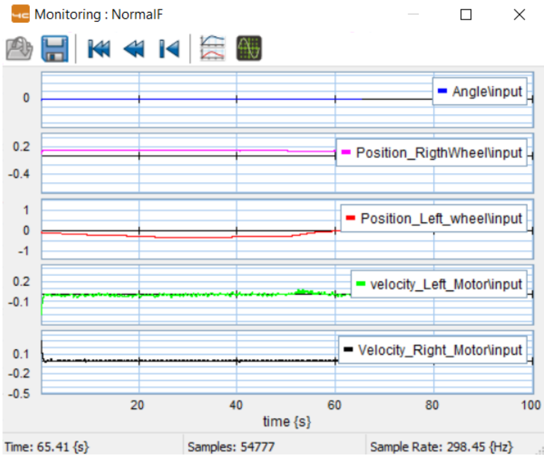

Figure 40 Performance Wheel_velocity controller real life

Note: The units in the Y axis in this figure are the same as Figure 41

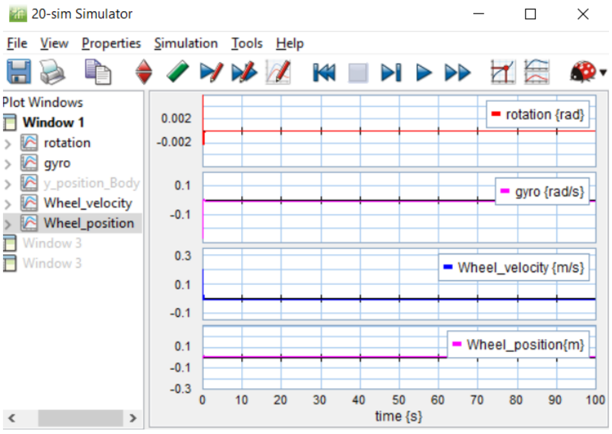

Figure 41 Performance Whee_velocity controller simulation

Figure 40, 41 clearly shows a steady state of the position and velocity of the wheels when the Segway was turn it on and let it go without any disturbance.

C. *User controller*

Finally, to be able to control the Segway a Bluetooth gamepad is given. The idea is that the user has full access to the Segway's motion. To accomplish this, two parameters need to be solved: back and forward motion as well as rotation of the Segway.

The first problem is solved by adjusting the reference in the velocity controller and it can be assigned e.g. the analog joystick Y signal such that moving it up will move the Segway up and similarly in the backward case.

The rotation of the Segway can be realized if the signals going to each motor are controlled and are e.g. stopped while the other wheel is moving. This can be seen as clamping one wheel while only using the power of the other wheel to rotate the Segway. For this case and as a matter of keeping it as logical as possible the digital button R2 and L2 can provide the limiting signals to meet this behavior.

Altogether, the conditions mentioned before are implemented in 20 sim, 4C and tested in real life.

Figure 42 and 43 shows how this control systems works

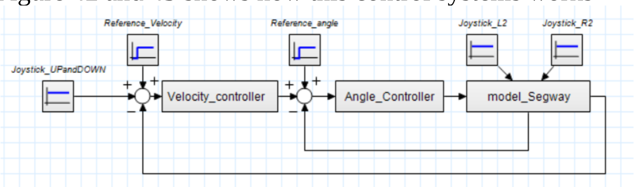

Figure 42 Implementation User control in 20 sim

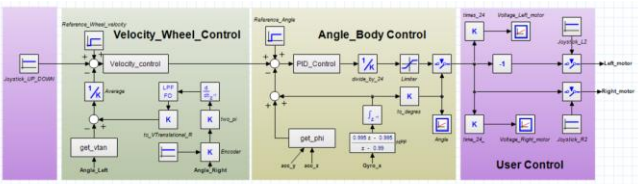

Figure 43 Implementation User control in 4C

Note: A bigger version of this figure is shown in the appendix of this report

Finally, the validation of the controller is tested and as a matter of proof the following picture should be sufficient.

V. Discussion

- Challenges:

A demo session was held which involved a series of challenges, namely:

- Standing still
- Standing still with a disturbance
- Driving from start to finish, turn around, and come back

All of these challenges were successfully tested, and the Segway used -17- to perform this task awarded among the first 10 Segway's with the best control system.

A couple of things, however, can be improved. The user controller for example. The decision of using switch breakers when a conditional signal -Joystick R2 and L2- limited the capability of the Segway to respond to abrupt changes in velocity. This was because of the constrains assigned to these switches. However, during the demo of this project this issue was solved.

VI. Conclusion

As proposed in the beginning of this report, a thorough model of the Segway body was formulated through the bond graph method, followed by multiple controllers to control the velocity and angle of the Segway such that a stable system is built. All simulations were done in 20sim and the controllers tested with 4C.

The validation results prove a well characterized system, given by the similarity of the compared signals in each figure presented before. Finally, the controller managed to stabilize the system and a robust Segway model is built.

References

[1] P. Breedveld, *Integrated Modelling of Physical Systems - Chapter 26*.

[2] *Control System Slides, PID controllers . 2018-SandC-Les4-Classification.pdf. Last viewed: 23 January 2020*

APPENDIX A

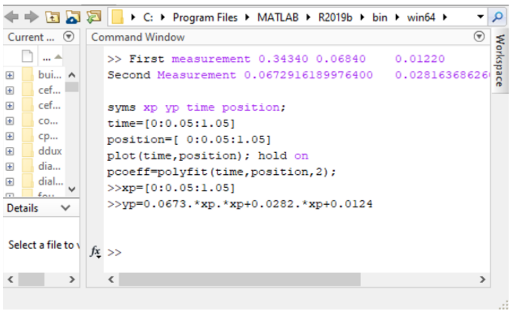

Figure 44 Code used to generate Figure 23

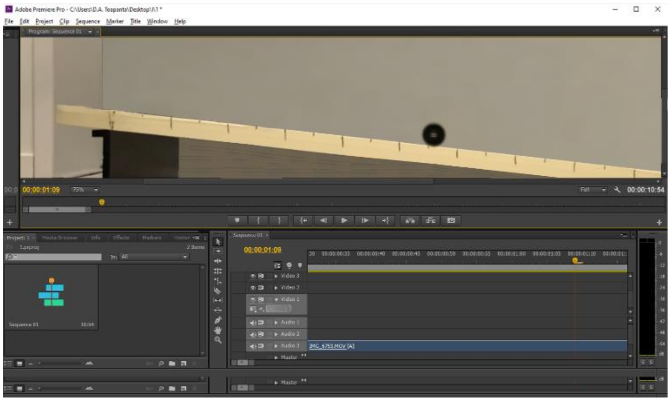

Figure 45 Video editor showing Right Wheel experiencing rotational motion

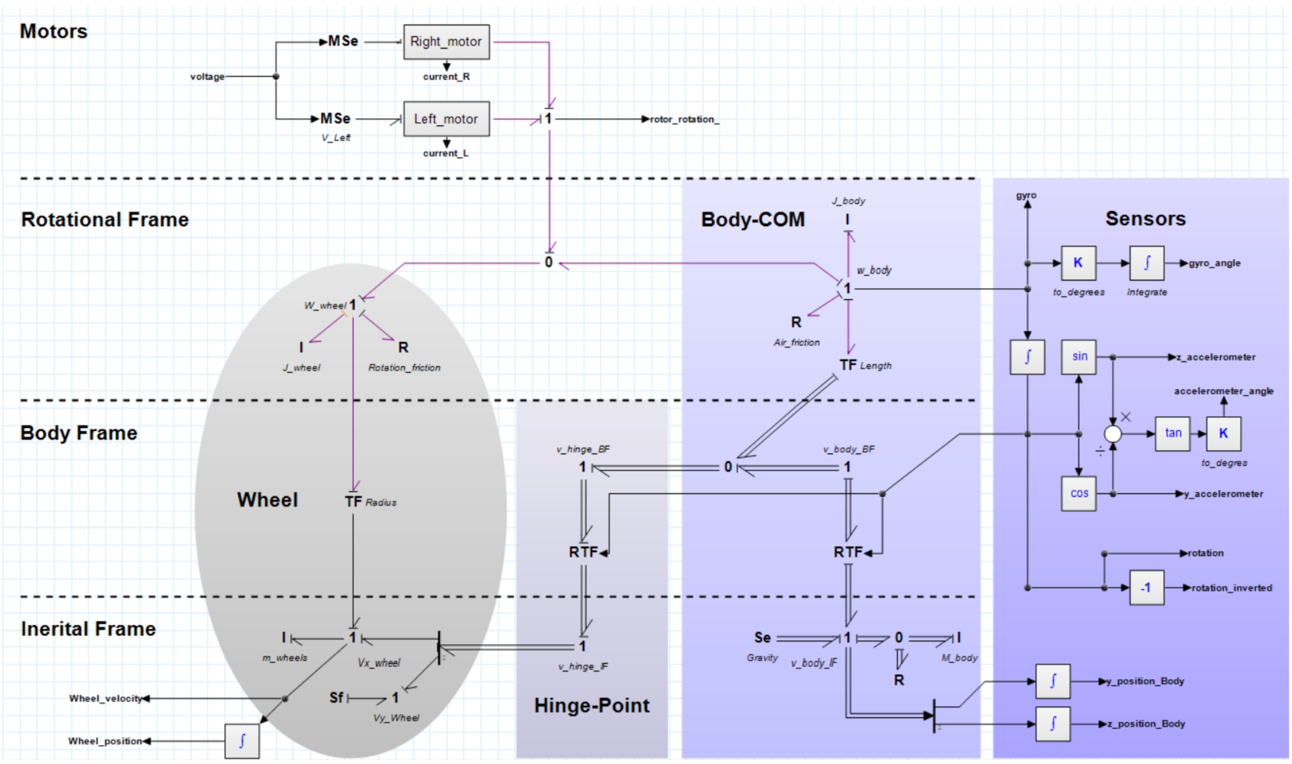

Figure 46 Complete model of the Segway

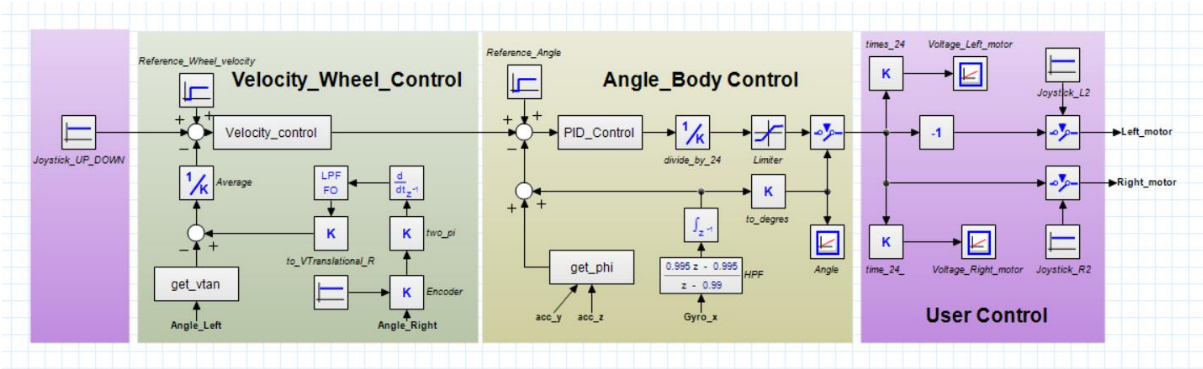

Figure 47 Final model control for 4C

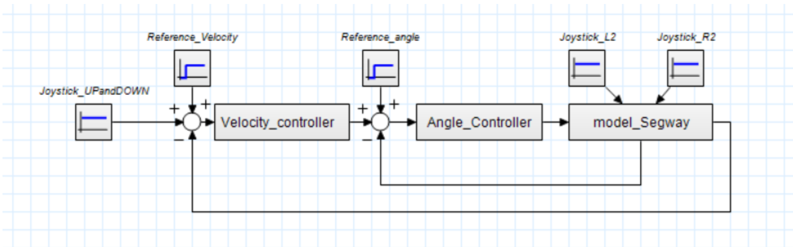

Figure 48 Final model plus cascade of controllers