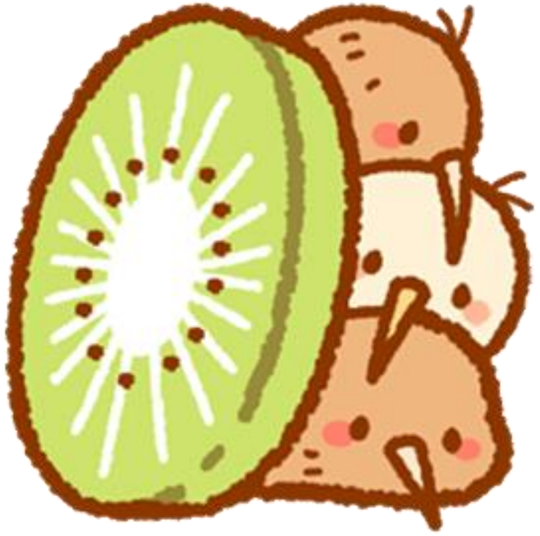
ES197 Group Project - Kiwi

Self-balancing robot



Oliver Beresford, Felix Bowyer, Betty Liu,
Adam McBride

Project Summary

- Program a self-balancing robot.
- Build and upload a Simulink model in order to achieve this.
- Robot should be able to self-balance on carpet and possibly other surfaces.
- Robot should be able to do so for a substantial period of time.

Customer Requirements

- Our self-balancing robot needs to meet four key requirements as issued by the customer:

  The robot should balance itself on its two wheels for a significant amount of time on the provided carpet and possibly on other random surfaces for the project.

  The Self-Balancing Robot (SBR) will be returned to the project leader after the demonstration on Friday, intact, in its box with all the accessories present.

  MATLAB R2020b and Simulink will be used to set-up the control system of the robot.

  The robot will be named. The name and the robot must be suitable for use on Warwick's Outreach activities to celebrate Coventry City of Culture 2021, so suitable to use with children.

Self-balancing robot

- This requirement formally states:

  "The robot should balance itself on its two wheels for a significant amount of time on the provided carpet and possibly on other random surfaces for the project."

  This means that we need to do program the robot such that:

  If it starts to tip over, it uses its motors to tilt back to a set angle.

  It stays at approximately a set angle for a substantial period (we decided on 1 minute).

  It can do so on carpet.

  (Ideally) it can do so on other surfaces.

Returning the robot

- The requirement formally states:

  "The Self-Balancing Robot (SBR) will be returned to the project leader after the demonstration on Friday, intact, in its box with all the accessories present."

  This means we need to do the following:

  Avoiding breaking the robot by being careful with it and following all safety advice given in the project briefing

  Keep all the accessories together in the box when not in use to avoid losing any.

  Return everything, in the included box, to the project leader on Friday after the demonstration.

Control system

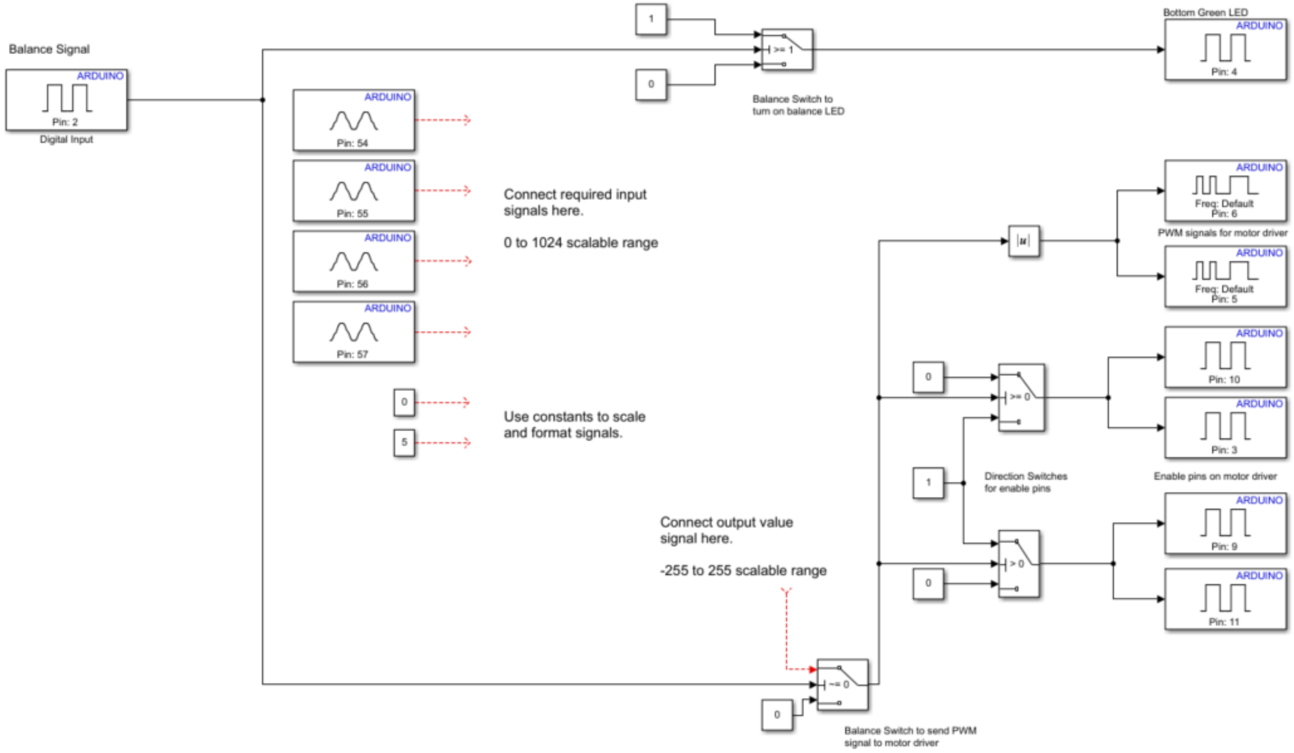
- The requirement formally states:

    "MATLAB R2020b and Simulink will be used to set-up the control system of the robot."

  This means that we need to do the following:

    Design a model in Simulink using the provided template.

    Upload the designed model to the robot.



SBR_START_MEGA.slx - Block Diagram

Robot's name

- This requirement formally states:

  "The robot will be named. The name and the robot must be suitable for use on Warwick’s Outreach activities to celebrate Coventry City of Culture 2021, so suitable to use with children."

  This means that we need to do the following:

  Robot must be given a name that is suitable for children.

  Since the age range of "children" is unspecified, we can assume that the lowest age rating of the British Board of Film Classification and/or the Pan European Game Information (Universal and Age 3 respectively).

  Both ratings state:

  No sounds or pictures that are likely to frighten young children

  No bad language.

Software Requirements

- The only customer requirement that dictates software requirements for this project is that the robot much be able self-balance. In order to achieve this, the software must be able to do two fundamental things:

  - Detect the robot's pitch from the internal measurement unit (IMU).
  - Affect both motors in order to move the robot.

  Plus

  - Only tilt is needed

Software Requirements

Improvement --- how to?

- Detect an alteration in pitch
- Detect how fast and how far it has moved away
- Avoid oscillating too much
- Correct its tilt

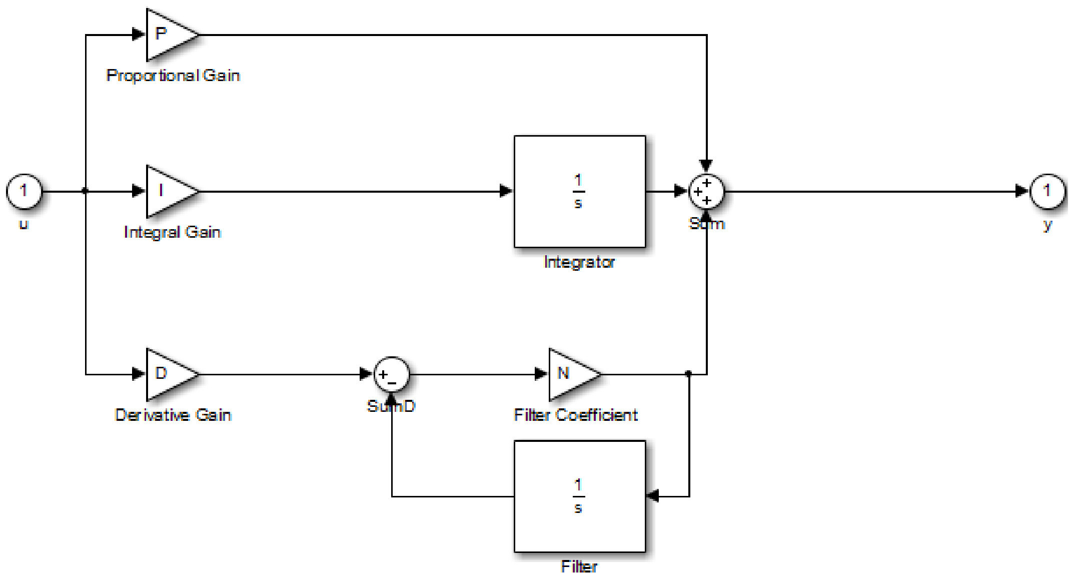
Implementation

- Proportional Integral Derivative (PID) systems

$$P + I\frac{1}{s} + D\frac{N}{1+N\frac{1}{s}}$$

MATLAB 2020b – Simulink – PID Compensator Formula

https://stackoverflow.com/questions/39757487/matlab-pid-filter-coefficient



- Tuning is needed

Test-enhanced tuning

- Agile testing used to further refine the PID parameters of the system
- Previous values decided with manual tuning methods – increase P until consistent oscillations, increase D until not wandering, increase I until stable.

Var	Increase	Decrease
P	More frantic movement, overshooting	Too slow to react to tilt, falls over more often
I	Reaches narrower oscillations in equilibrium, but too sensitive to push	Slightly wider oscillations in equilibrium, but recovers more easily from pushing
D	Oscillates faster and narrower; sometimes falls out of stable equilibrium. More resistant to pushing.	Much more prone to overshooting.

Experimental difference in robot behaviour when changing individual PID parameters

CL Response	Rise Time	Overshoot	Settling Time	S-S Error
K_p	Decrease	Increase	Small change	Decrease
K_i	Decrease	Increase	Increase	Decrease
K_d	Small change	Decrease	Decrease	No change

Effects of each of the PID coefficients (K_p – Proportional, K_i – Integral, K_d – Derivative. [1]

- Testing showed that K_P was roughly ideal, K_I should be lower, K_D should be higher
- K_I decreased from 1 to 0.75, K_D increased from 0.15 to 0.2.
- Robot shows better stability, especially on less tactile surfaces, and is more resistant to movement.
- The tuned values aren't perfect, but they are very good, and as good as can feasibly done in a reasonable time frame / without mass computation.

[1] https://ctms.engin.umich.edu/CTMS/index.php?example=Introduction&section=ControlPID

Testing

- To test the robot, we used a mixture of agile and unit testing.
- This was suitable, as it allowed us to combine testing and iterative development to quickly produce and optimise a functioning robot.
- To begin we developed a test plan that outlined everything that would take place during the testing period.

Test No	Title of test	Purpose of test	Requirements
1	Pitch Test	Ensure the robot reacts to changes in pitch correctly	2
2	Balance Test	Ensure the self-balancing robot works on the carpet for a period	2
3	Push Test	Ensure the self-balancing robot works on the carpet for a period	2
4	Material Test	Check how long the robot can balance on different materials.	1

Pitch Test

Purpose of test: Ensure the robot reacts to changes in pitch correctly.

Set-up/Equipment:

- Two participants
- Kiwi – the robot
- Computer with Simulink program and OBS
- Camera

Customer Requirement –

The robot should balance itself on its two wheels for a significant amount of time on the provided carpet, and possibly on other random surfaces for the project.

Test 1 Procedure

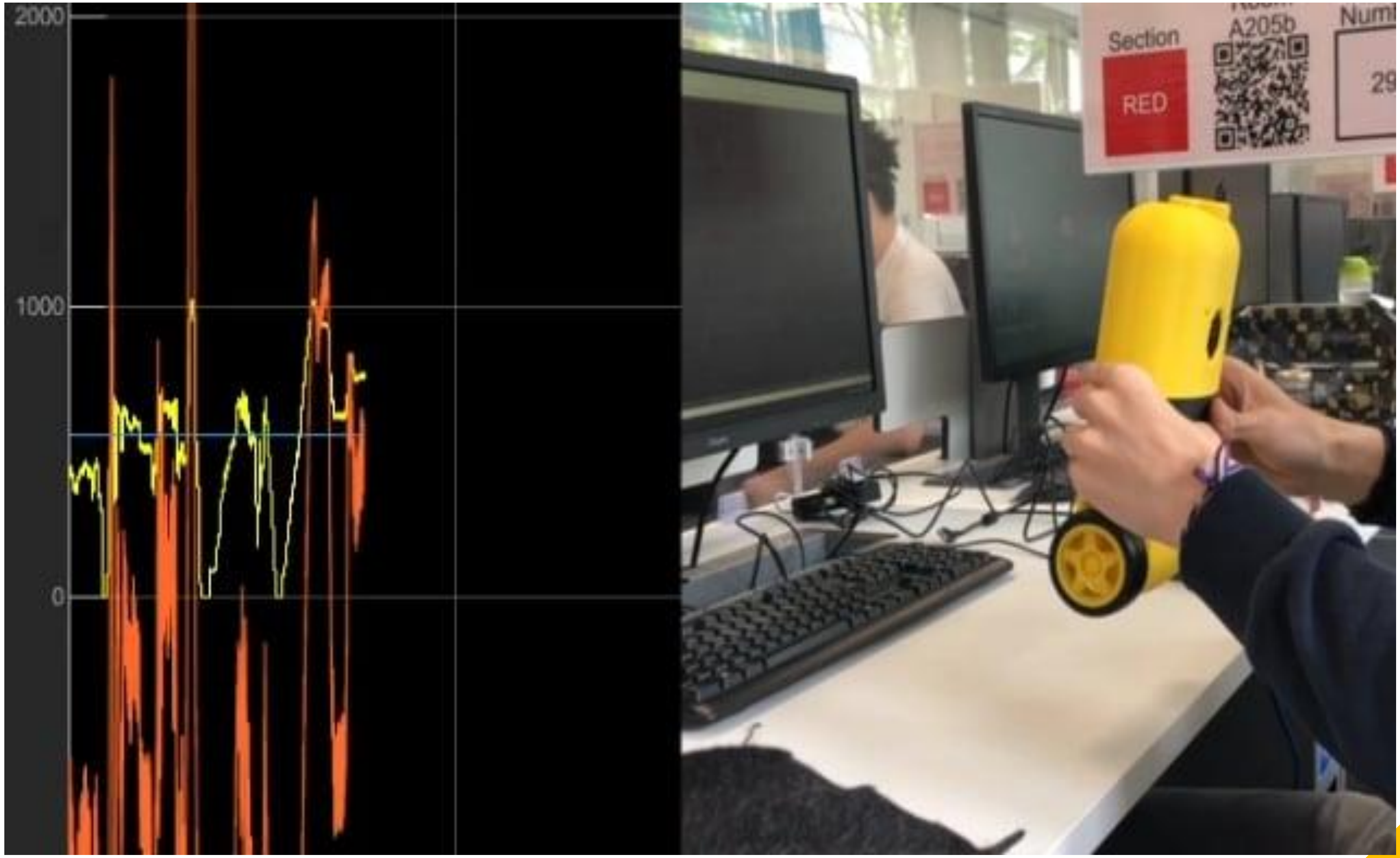
First, connect the robot to the computer and open the scope of the robot.

Start OBS to record the screen and record the scope through Simulink.

The second participant should begin recording the robot.

Tilt the robot in all directions to record the scope.

Compare the scope outputs to the robot motor function.


2000
1000
0
Section
RED
A205b

Pitch Test Results

Test no.	Test Iteration	What is being tested?	Input	Input Type	Expected Output	Success
1	1	Does the robot react appropriately to stimuli when held?	Start the robot upright	Valid	The wheels will not spin as the pitch matches the potentiometer.	Yes
	2		Tilt the robot forward to pitch limit	Boundary	The wheels will spin forward to attempt to match the pitch to potentiometer.	Yes
	3		Titled the robot backwards to pitch limit	Boundary	The wheels will spin backwards to attempt to match the pitch to potentiometer.	Yes
	4		Tilt the robot backwards past the pitch limit	Invalid	The wheels will not spin as the pitch is outside the boundary.	Yes
	5		Tilt the robot forwards past the pitch limit	Invalid	The wheels will not spin as the pitch is outside the boundary.	Yes
	6		Tilt the robot to the left	Invalid	The wheels will not spin as the pitch will not change.	Yes
	7		Tilt the robot to the right	Invalid	The wheels will not spin as the pitch will not change.	Yes

Balance Test

Purpose of test: Ensure the robot reacts to changes in pitch correctly.

Set-up/Equipment:

- Two participants
- Kiwi – the robot
- Carpet
- Timer
- Camera

Customer Requirement –

The robot should balance itself on its two wheels for a significant amount of time on the provided carpet.

Test 2 procedure

First, participant one powers on the robot.

The second participant then records the robot.

First participant places the robot down on the carpet.

The second participant will begin timing as soon as the robot is placed down.

Once the robot has fallen, the timing and recording will stop.


RED
0.000

Test results

Test No.	Test Iteration	Average Time Standing (seconds)	Problem Recorded	Solution Implemented
2	1	~1	High overshoot	Increase Derivative
	2	5.56	System diverged	Increase Integral and Derivative
	3	3.46	Low stability (jitters)	Decrease Derivative
	4	6.53	No steady state reached	Increase Proportional
	5	60+	Inconsistently reached steady state	Increase Derivative
	6	60+	None, move on to further testing	-

This test was performed in an agile manner, and so was used to diagnose issues and incrementally improve the robot's performance during implementation.

Push Test

Purpose of test: Ensure the robot reacts to changes in pitch correctly.

Set-up/Equipment:

- Two participants
- Kiwi – the robot
- Carpet
- Camera

Customer Requirement –

The robot should balance itself on its two wheels for a significant amount of time on the provided carpet, and possibly on other random surfaces for the project.

Push Test Procedure

First participant one powers on the robot.

The second participant can then start to record the robot.

They can then place down the robot on the carpet.

Wait for the robot to reach a steady state.

The first participant can then push the robot.

The second participant should record if the robot falls or stabilizes.

Test results

Test no.	Test part	What is being tested?	Material	Recovery % (of 10 trials)
3	1	Test if Kiwi can be pushed and re-stabilise	Thick Carpet	100
	2		Medium Carpet	100
	3		Floor Carpet	0
	4		Felt	10
	5		Table Top	0

On harder surfaces, the robot failed to get enough traction to effectively self-right but was very good at self-righting on thicker carpets.

Materials Test

Purpose of test: Test if the robot can stand on a variety of materials.

Set-up/Equipment:

- Kiwi – the robot
- Carpet
- Timer

Customer Requirement –

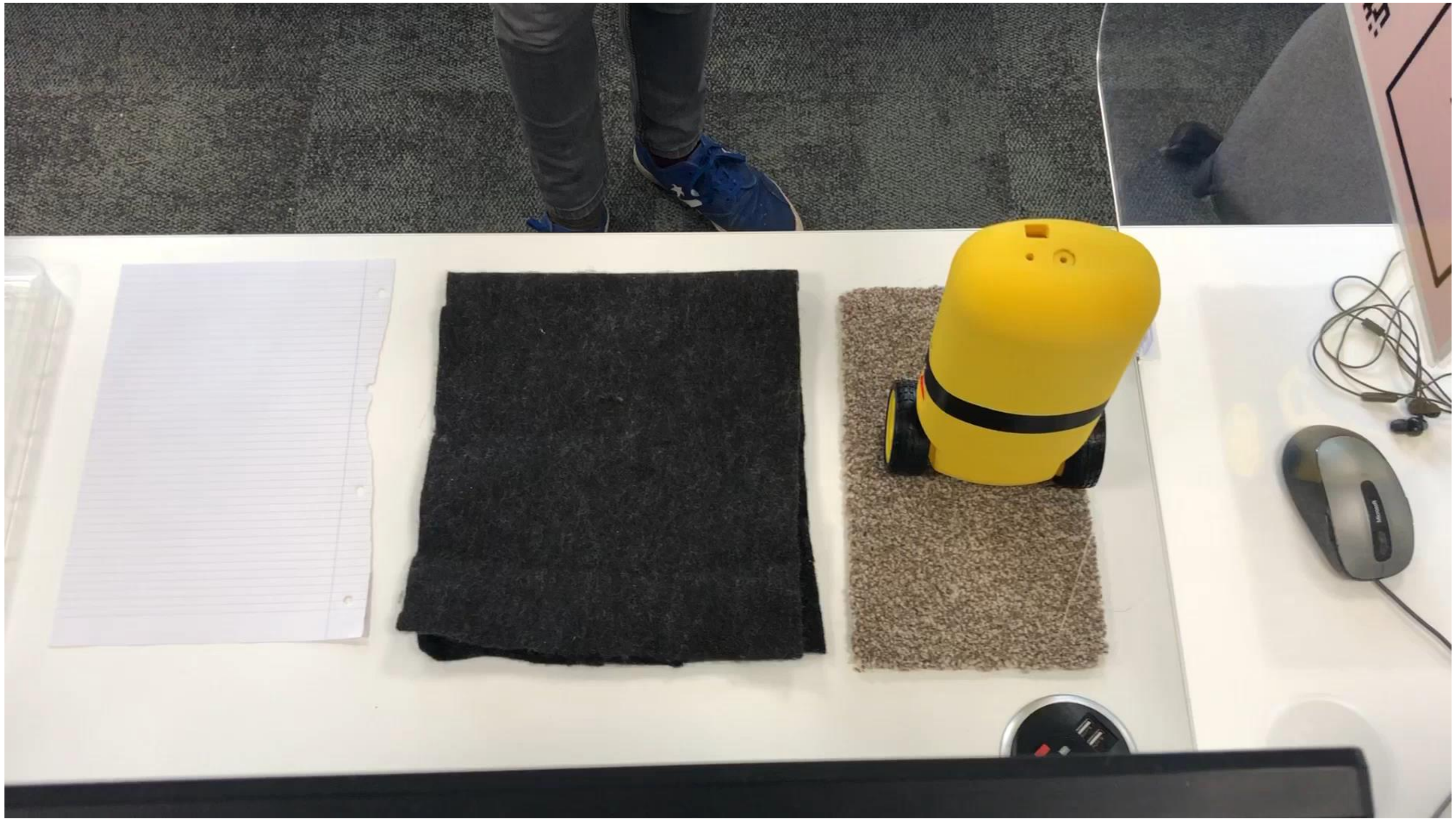
The robot should possibly balance itself on its two wheels for a significant amount of time on other surfaces.

Materials Test Procedure

Participant turns on robot and places on material, and waits until it reaches a steady state

Robot is timed for up to 1 minute, or until it falls

Time is recorded and experiment is repeated



Test results

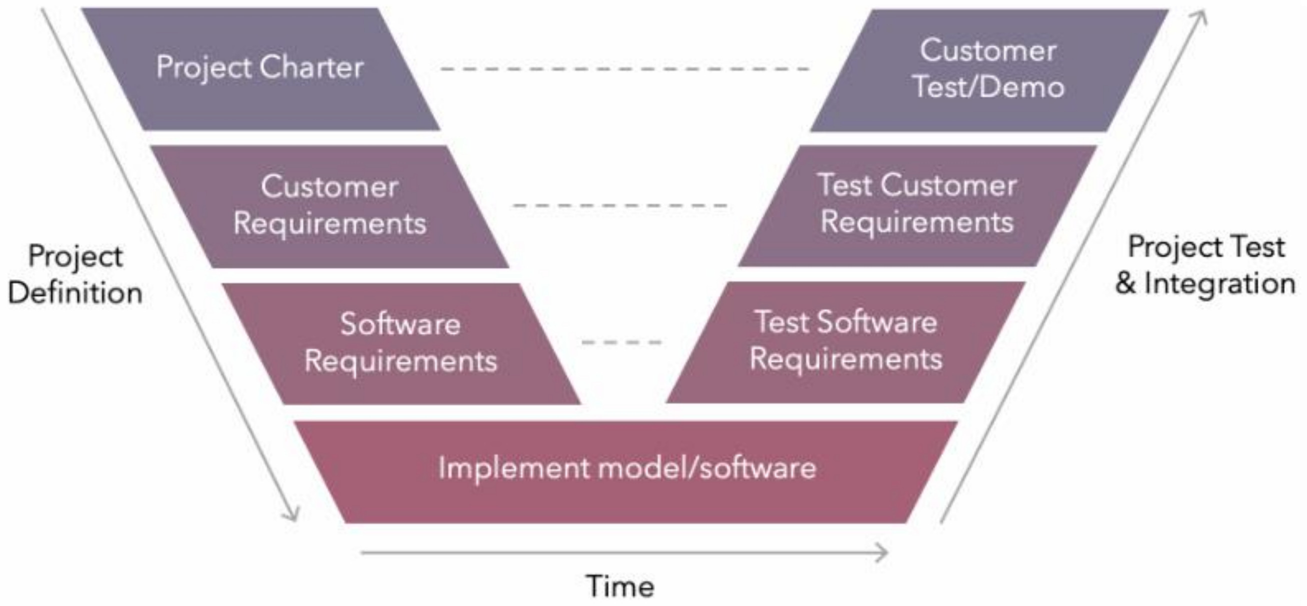
Test No.	Material	Average Time Standing (seconds)	Standard deviation (seconds)
4	Thick carpet	60+	N/A
	Medium carpet	60+	N/A
	Floor carpet	60+	N/A
	Felt	60+	N/A
	Plastic	20.44	11.99
	Table	22.25	11.60

Timing was stopped at 60 seconds.
Much better on soft materials – more traction.
High standard deviation shows inconsistent results on hard surfaces.

Demonstration

The V-Model

- The Verification and Validation Model (known as the V-Model) covers design, implementation and testing with regards to a project.



Define stage

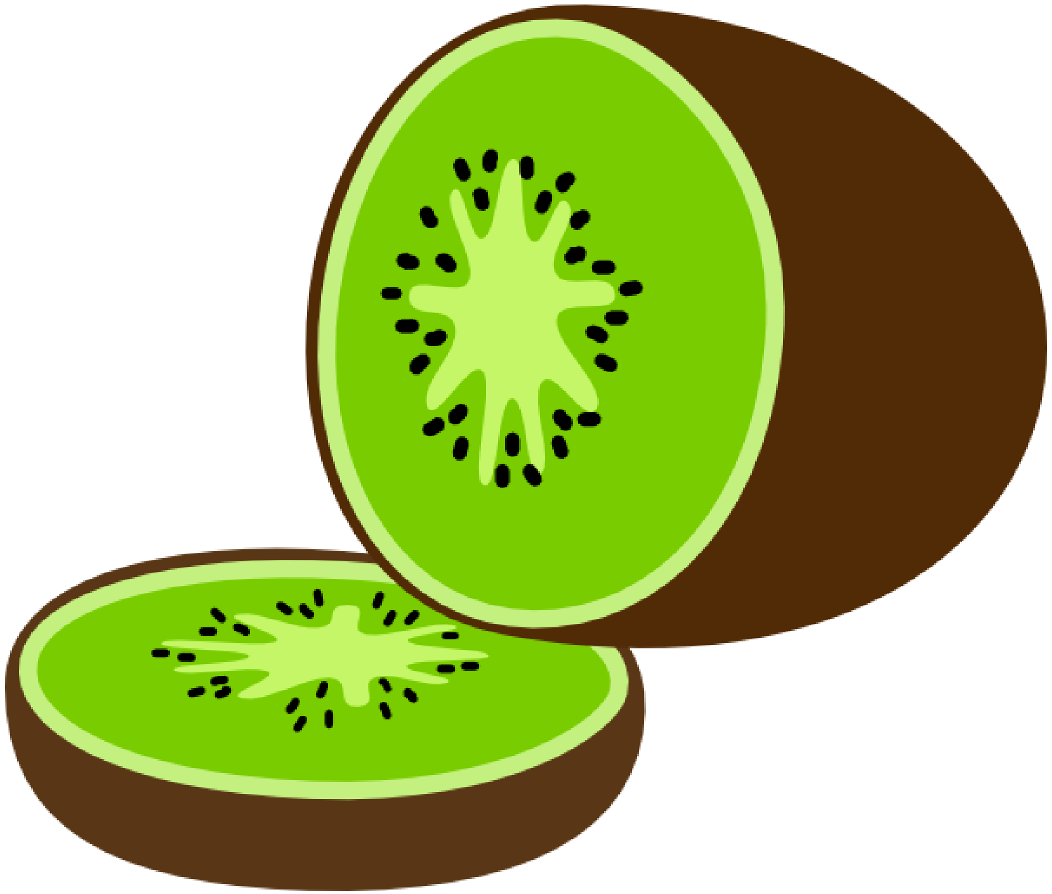
- Project charter – get everyone on the team on the same page. Ensure we have an idea of how we will implement the functionality and have the correct prior knowledge.
- Customer requirements – look at project brief together, decide how long the robot should stand, break requirements down into abstracted, attainable steps.
- Software requirements - decide together how the software should act, i.e., it should avoid large oscillations, mainly stay in place.

Implement and Test stages

- Implement Simulink model using PID controller
- Integrate some testing into implementation in order to create an effective robot
- Test robot against customer and software requirements set out in project charter
- Demonstrate robot to customer (as just done)

Our Robot's Name

- Kiwi
- Cute, suitable for kids



Questions

Thanks for listening, any questions?