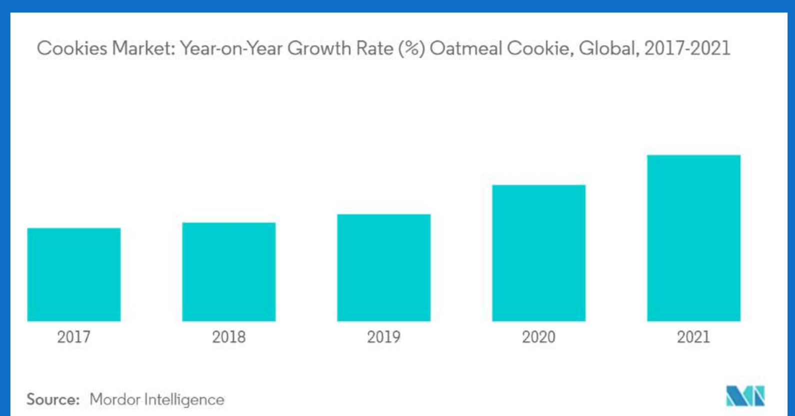
C🍪🍪KIEBOT
THE MONSTER ALL BAD COOKIES FEAR

Group 1
CBL Systems and control project
Yuvan Dwaraga - 1563793
Jon A. Almagro Aranguren - 1578421
Lotte Meeuwsen - 1439200
Arvind Ramanathan - 1541838
Gergely Palfi – 1557726
Arjen Smetsers - 1567691
Student number 6

Customer and problem definition

Where it all began...

It is an industry everybody has reaped the fruits from, yet no one knows much about: *the Cookie Industry.*



The cookie industry is estimated to grow by 5.5% annually between 2022 and 2030 to reach $40 billion. Especially in developing countries the demand for cookies is on the rise.

THE COOKIE NEEDS TO BE ORANGE, CIRCULAR, AND 7 CM IN DIAMETER. OTHERWISE, I'LL REMOVE IT!

RPCs:

Requirements:

- Recognize objects with a **90% accuracy** based on shape, color, and size.
- Ability to pick up an object with vacuum head from stationary belt **within 6 seconds** and a **success rate of 90%.**
- Ability to place the object to the desired position next to the belt **within 6 seconds** and a **success rate of 90%**.
- Cookies should not be **dropped** when the arm is in motion. They should only be dropped at a pre-defined location.

Preferences:

- Robot arm can move **10 items per minute** to be efficient.
- Accuracy of vacuum grip can be **+-5mm** from the center of an object.
- Designated area of the dropping of position can be **at 90 degrees** next to the robot**.**
- Account for as many frictional components as possible, hence improve tracking performance

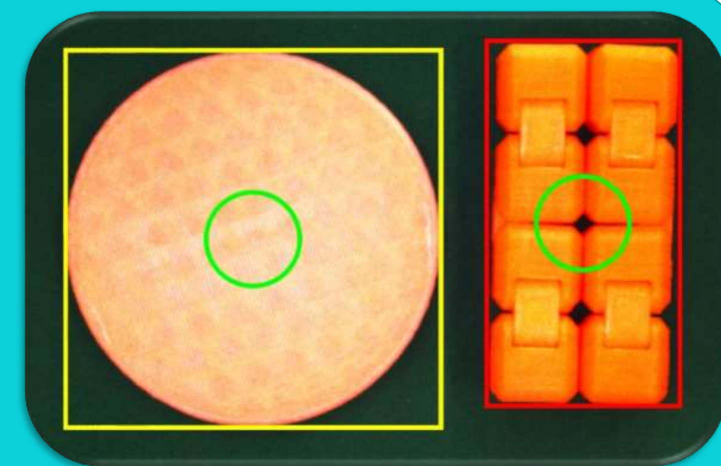
Constraints:

- Goals must be realizable within a **9-week** project span.
- Robot task must be possible with a **single suction cup**.
- Desired tasks must use provided **sensors**.

Image Recognition

...The selling point of the CookieBot...

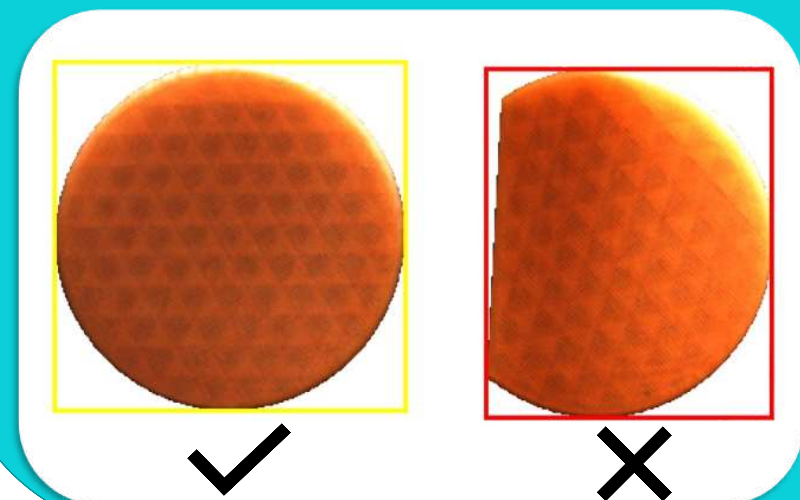
- Several reference objects were **3D-printed**, both faulty and correct ones.
- The **color thresholder app** was used to create the desired color.
- The camera outputs the **coordinates** of faulty objects only.
- Objects that touch the **frame** are not registered so that these are not seen as faulty.



CONCLUSIONS:

- *No constraints have been exceeded.*
- *Image recognition can only be done for a conveyor belt speed of 40 mm/s or less (inverse kinematics required a speed of 20 mm/s,).*

Cookie size	Lower bound	Upper bound
Accuracy	97.9%	90.5%



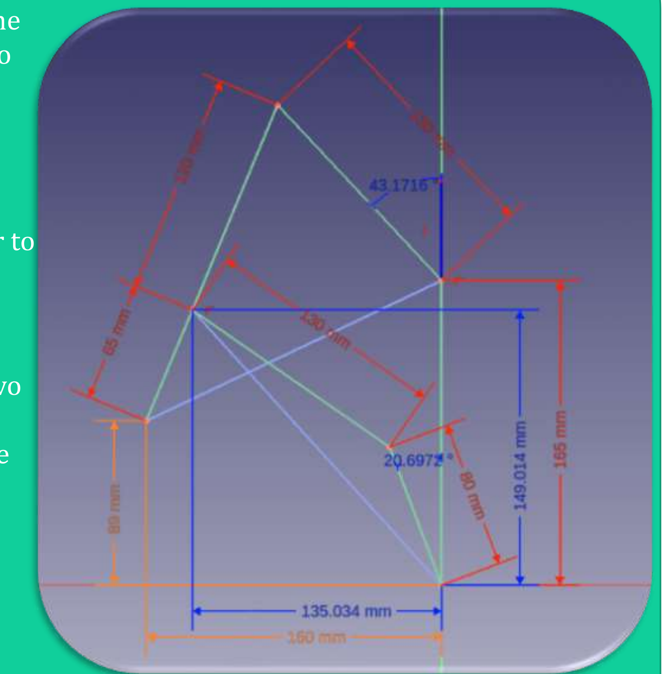
RECOMMENDATIONS:

- Improve the script so that the system can handle **higher conveyor belt speeds**. If the speed becomes too high, then the frames see the cookie as a larger object than it is. This will make the system **more efficient**.
- For now, only three objects have been considered. Testing with a **more diverse object set** would be better to further test the limits of the system. Objects within a **5% or less dimension window** can be tested to see whether the system still sees them as faulty.
- The area of this specific object has been **manually** set in the script; meaning that it can only work with this specific object. The script could be improved in a way that it is more **resistant** to object changes while still detecting faulty objects.

Inverse Kinematics

...The act of getting the robot to the right position...

A parametric FreeCAD model of the robot arm was created and used to validate the angles.



1) Transform XY plane to **polar coordinates**
2) Calculate diagonals from motor to tip of arm
3) Use **cosine rule** to get necessary angles
4) Calculate link position of the two triangles
5) Repeat diagonal and cosine rule for bottom triangle

Restricting solutions to triangles **above the diagonals** gives only one solution set.

The script has **failsafe checks** that exit impossible calculations.

InverseKinematics.m	GergeKinematics.m
MATLAB solve function	Pure trigonometry
Slow (20-30s)	Fast (0.02s)
Imaginary numbers	Single real solution set
Multiple solutions	

FUN FACT: THE SCRIPT WAS NAMED AFTER ONE OF THE CREATORS, SO HE GOES BY GERGEMATICS NOW!

Conclusions and Recommendations

RPC reflection

- Faulty cookie recognition has an accuracy of 97% for cookies that are smaller than the desired cookie or broken, and 90% for too large cookies.
- Picking and placement of the cookie is performed within 5 seconds, max 12 cookies per minute.
- The robot is placed correctly on the cookie 9 out of 10 times, and always drops the cookie at the correct position due to the adequate kinematics and controllers.

In conclusion, the CookieBot meets or exceeds the pre-defined requirements to perform Quality Control in the cookie industry, yet improvements can be made to increase the speed of the automated Quality Control process.

Recommendations

- Adjust image recognition, kinematics and controllers to function well at higher conveyor belt speeds.
- Tuning feedforward controllers could improve the movement of the arm further, also at higher movement speeds.

Customer and problem definition — Image recognition — Inverse kinematics — FRF measurements — Controller development — Stateflow

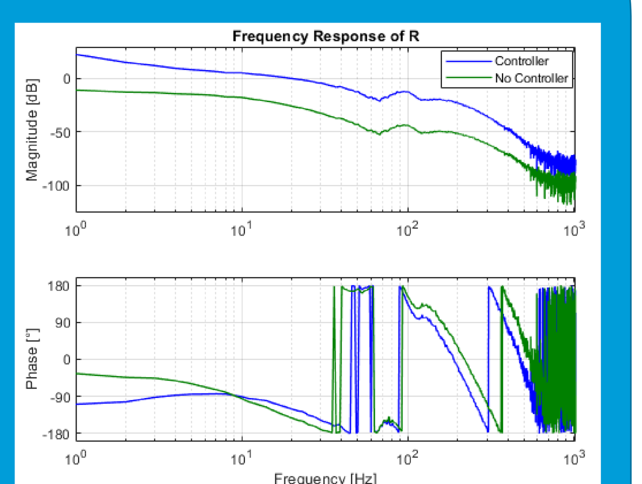
FRF

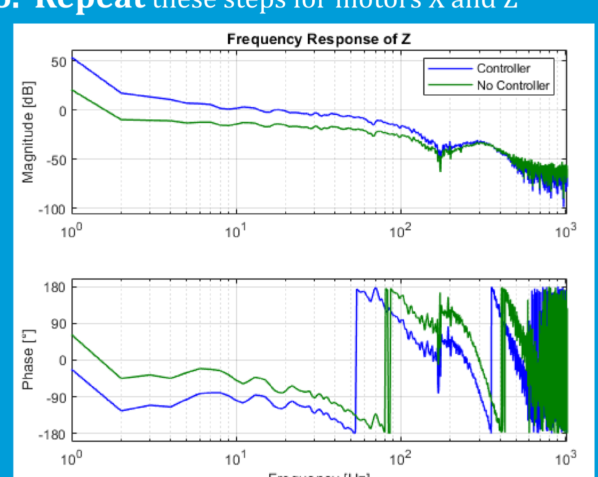
...The art of understanding the robot...

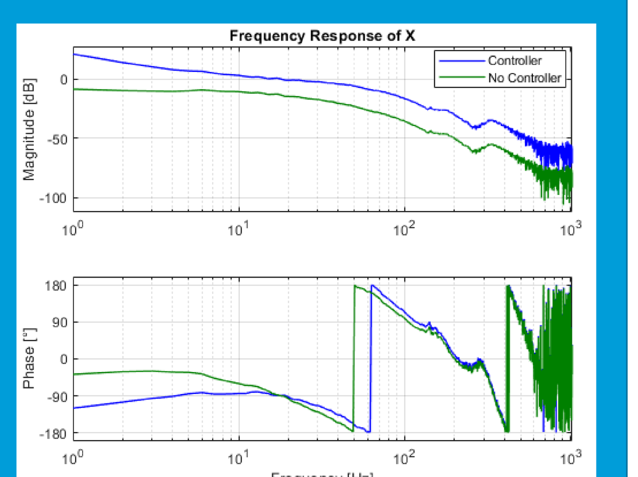
FRF measurements are conducted to understand and predict the behavior of the system.

Method:

1. Add **white noise** to the feed-forward block and connect it after the feedback block.
2. Add a **low-frequency sine wave** as a reference and increase the gain to 5 in the feedback block.
3. Connect **'ToFile'** blocks to measure the input and output for sensitivity
4. Use the input and output in **'tfestimate'** to set the window, overlap, and sampling frequency to generate the transfer function of the sensitivity.
5. Convert the sensitivity transfer function to the plant using the equation **H = 1/C*(1/S – 1)** and make a bode plot of the resulting transfer function.
6. **Repeat** these steps for motors X and Z







Are results as expected?

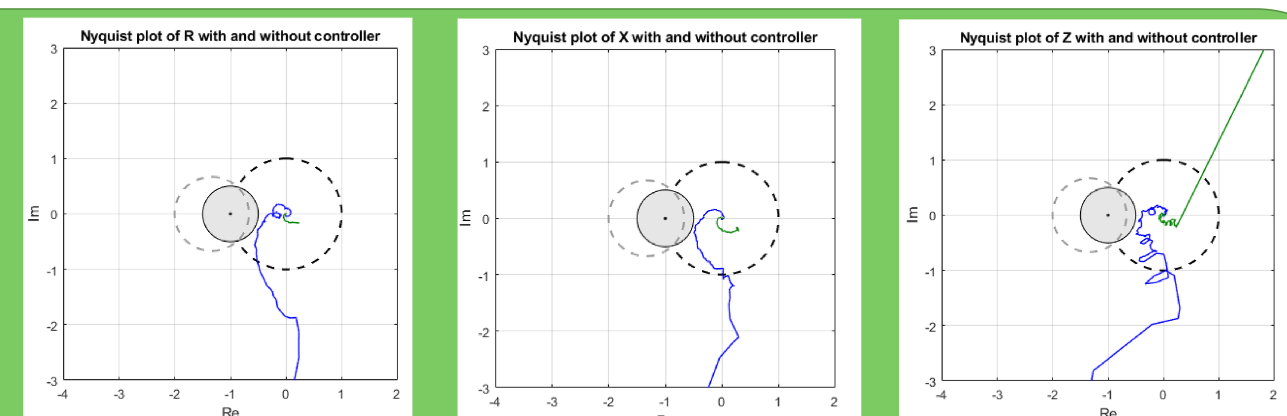
R	X	Z
-20 dB/dec 90° phase shift ✓	-20 dB/dec 90° phase shift ✓	Sharp slope at 1 Hz ✗ Robustness + tracking error prioritized

Feedback

...We learn from the past...

The goal of feedback control is to ensure good **tracking** of the reference signal while maintaining **stability** and **robustness**. The metrics used to judge the stability of the system are

- **Bandwidth ≈ 20 Hz**
- **Gain margin > 2 dB**
- **Phase margin > 45˚**
- **Modulus Margin < 6 dB.**



Controller R and X meet the requirements for stability. However, due to difficulties in designing a controller for the Z motor, it was decided to reduce the bandwidth and prioritize robustness and low tracking error.

	Add-ons	Band-with	GM	PM	MM
R	Gain, lead compensator, low pass filter, integrator	20.17 Hz	10.6 dB	58.8˚	4.94 dB
X	Gain, lead compensator, integrator	19.1 Hz	9.1 dB	91˚	5.24 dB
Z	Gain, lead compensator, low pass filter, integrator	14.1 Hz	8.3 dB	83.9˚	5.1 dB

Feed-forward

...And try to predict the future...

A Feed-forward controller modifies the input signal in a pre-defined way. It was used to overcome static and dynamic friction, and inertia.

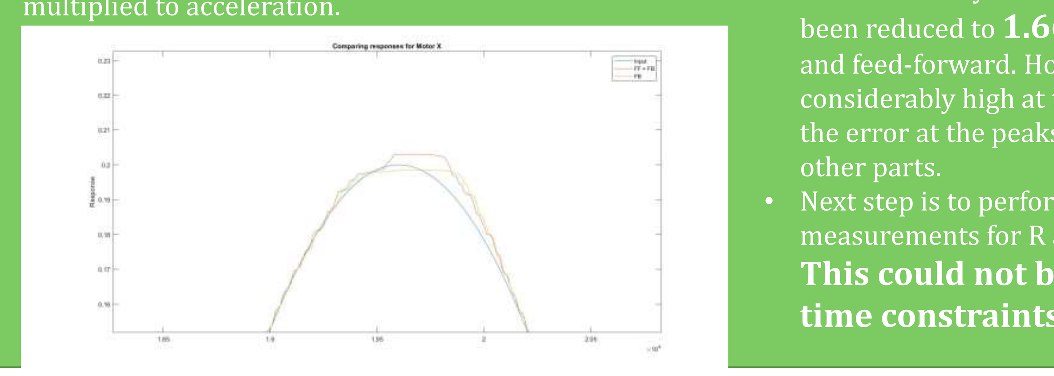
Process

- A ramp block with different slopes was tested for static friction. The smallest value for movement was noted.
- Constant block with increments of 0.2 were used five times in each of the motors to move it. Using this, the velocity of the motor was plotted against angular position. The peaks were extracted for each of the motors and plotted with the line of best fit. This line's slope was used as the constant for dynamic friction.
- Exploiting the fact that the arm is a second order system without any springs and dampers, inertia can be derived using this formula:
  $I = \frac{1}{2\pi f e^{\frac{20}{M}}}$ where "f" refers to frequency and "M" refers to the corresponding magnitude in dB.

Results for Motor X

	Static friction	Dynamic friction	Inertia
X	+/- 0.700*	+/- 0.206*	0.023

- * = depending on sign of velocity
- The first two gains are multiplied with the velocity and added to the input angle whilst the third component is multiplied to acceleration.



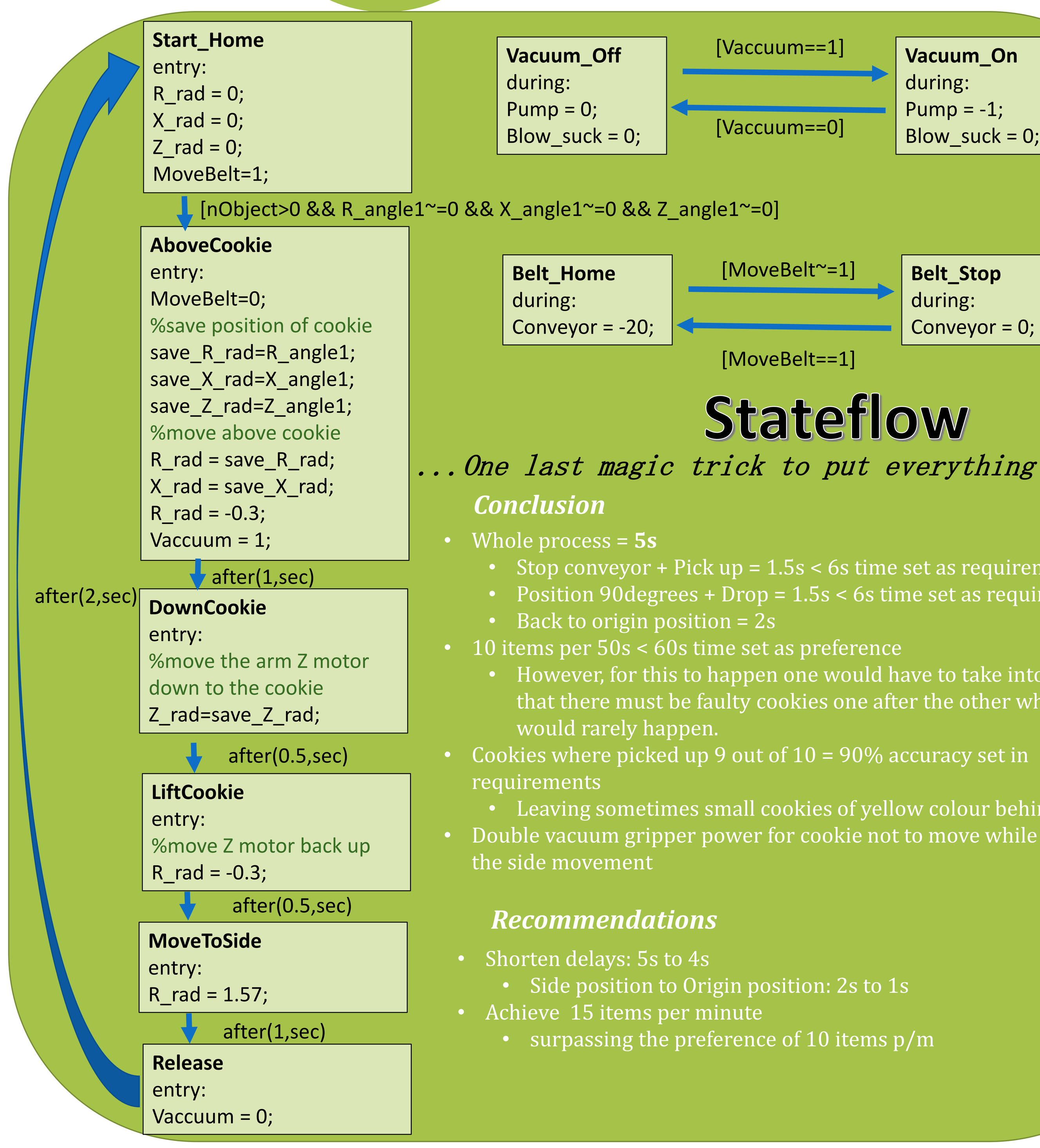
Conclusions

- Error difference between having only feedback and both feedback and feed-forward is **clearly noted**. Results only for motors X were collected.
- Error with only feedback was 1.91%, but it has been reduced to **1.66%** with both feedback and feed-forward. However, the difference is considerably high at the peaks. It makes up for the error at the peaks by better tracking at other parts.
- Next step is to perform Feedforward measurements for R and Z motors as well.

This could not be done yet due to time constraints.

Stateflow

...One last magic trick to put everything together



Conclusion

- Whole process = **5s**
  - Stop conveyor + Pick up = 1.5s < 6s time set as requirement
  - Position 90degrees + Drop = 1.5s < 6s time set as requirement
  - Back to origin position = 2s
- 10 items per 50s < 60s time set as preference
  - However, for this to happen one would have to take into account that there must be faulty cookies one after the other which would rarely happen.
- Cookies where picked up 9 out of 10 = 90% accuracy set in requirements
  - Leaving sometimes small cookies of yellow colour behind
- Double vacuum gripper power for cookie not to move while doing the side movement

Recommendations

- Shorten delays: 5s to 4s
  - Side position to Origin position: 2s to 1s
- Achieve 15 items per minute
  - surpassing the preference of 10 items p/m