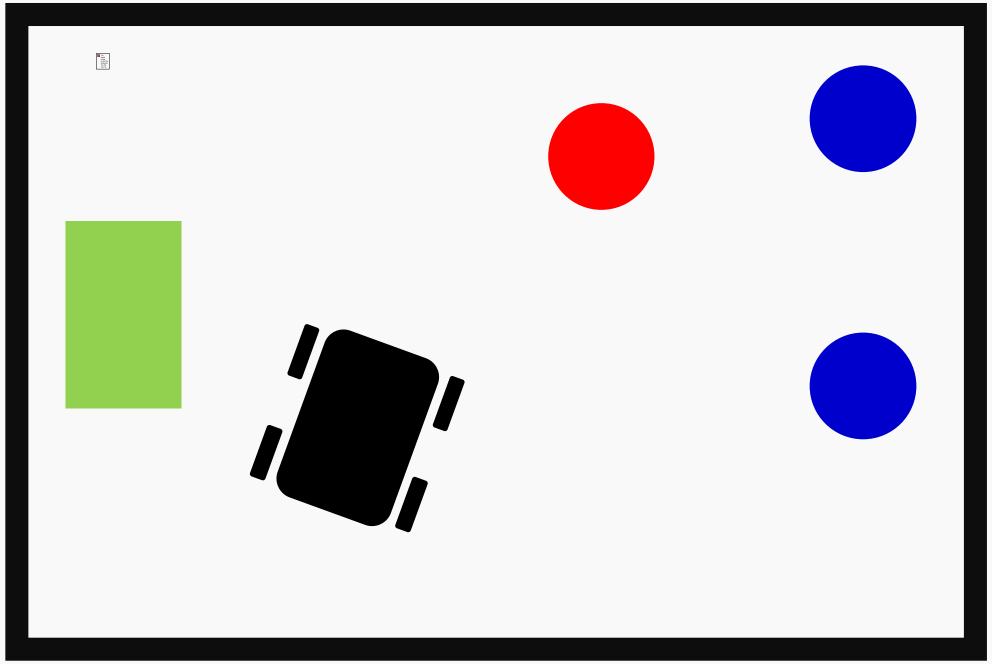


Trilogy: The End Game

Team. 진국

Anthony Barret, HyeonJae Gil, Byeongju Lee, Jihyuck Son
Joosung Lee, Yoonseop Lim, Seungyeon Jang

Presenter: Jongwon Lee
Prof. Kuk-Jin Yoon, TA Donghyeon Kim

KAIST

Dept. Mechanical Engineering, Korean Advanced Institute of Science and Technology

Contents Title

01

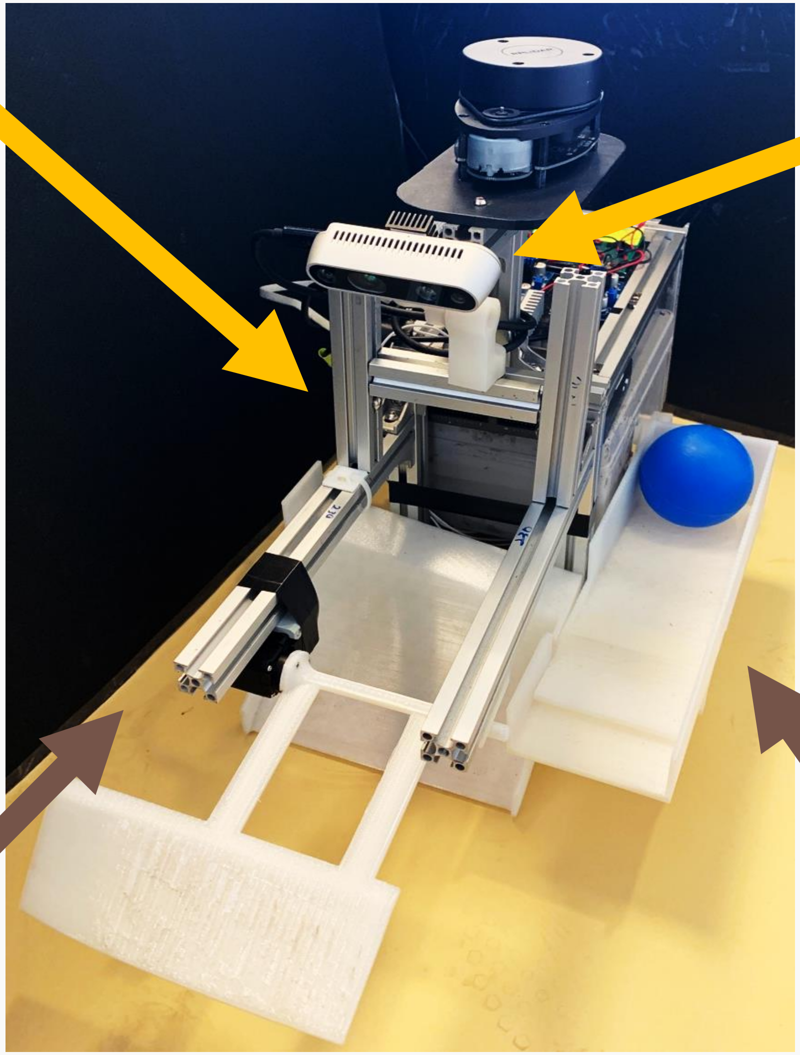
Design Philosophy

Design Philosophy

- **Goal:** ***Smart and compact***

Smart algorithm
Robust to any exceptional case

Smart sensor configuration
LiDAR + RGB-D Camera

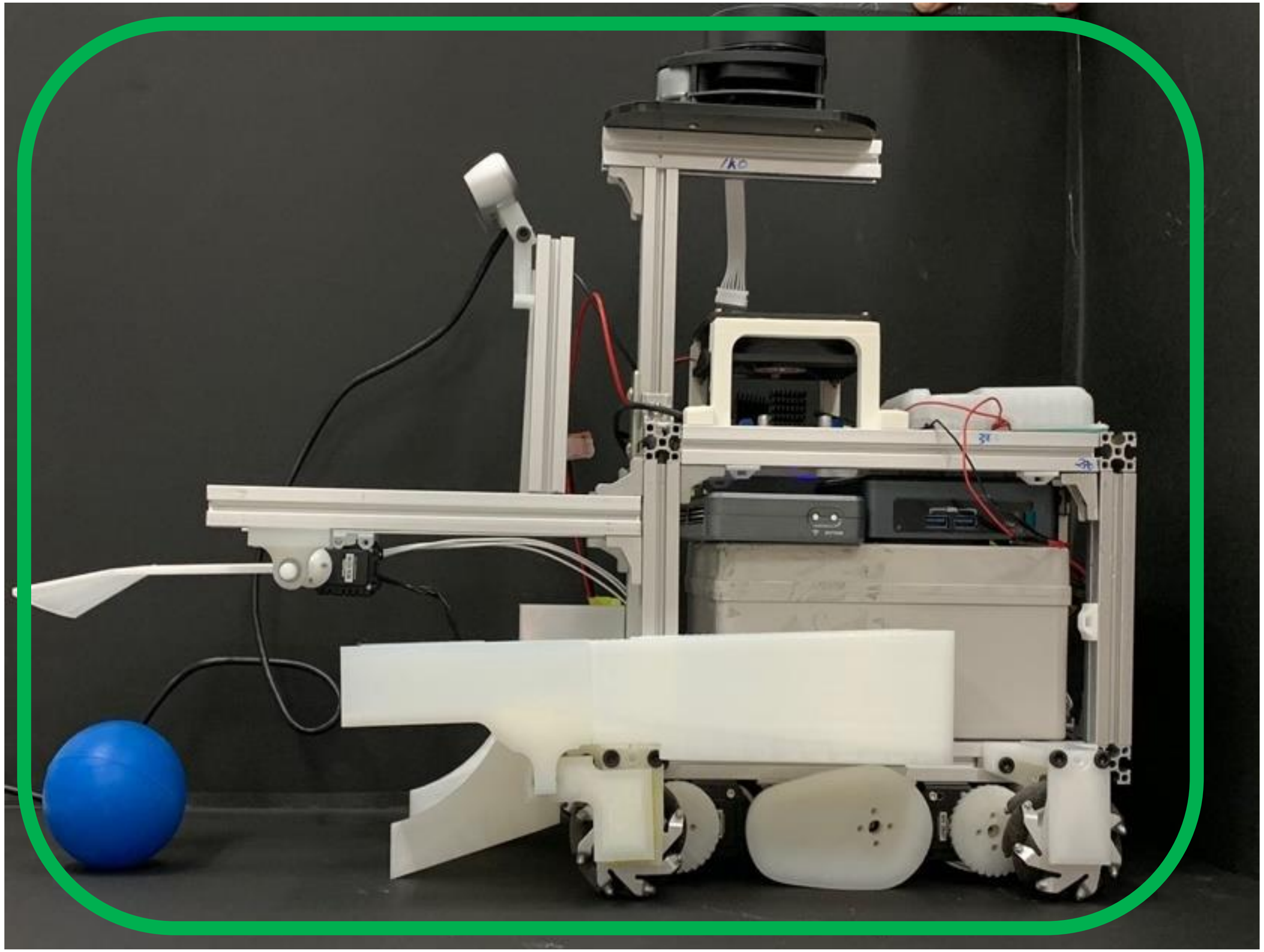


Compact design
Ensures minimal movement

Compact pickup & drop module
Minimizes redundant space

02 - System Design

Solidworks

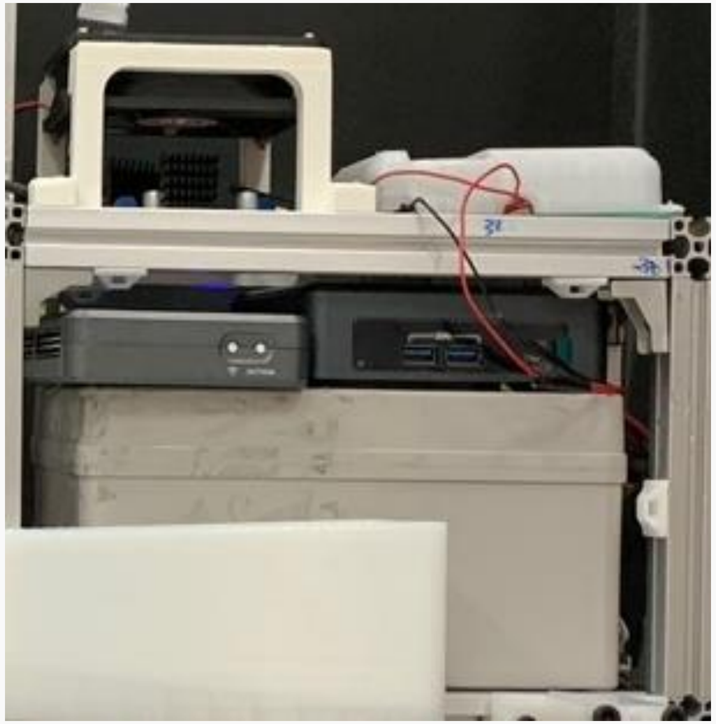


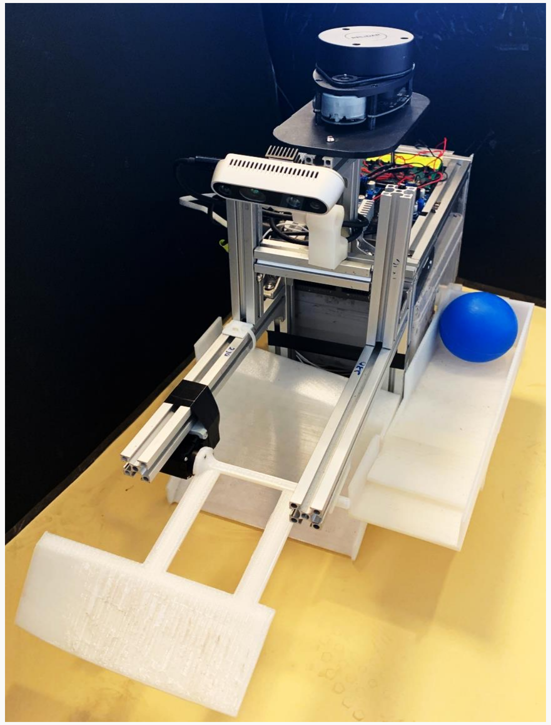
System Design

▪ Whole Hardware

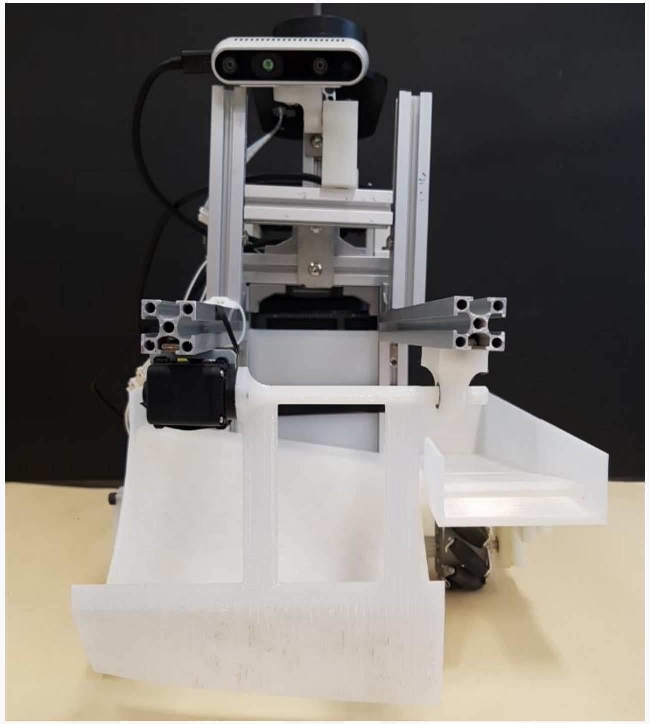
Goal: Hardware as compact and slim as possible

Compact





Slim



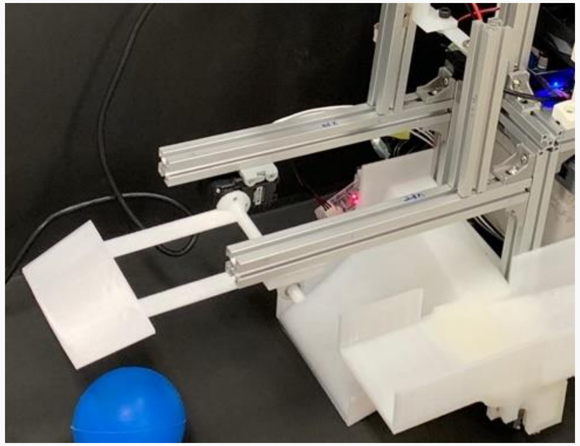
- Battery, NUC, myRio are piled up
- The load is well-considered

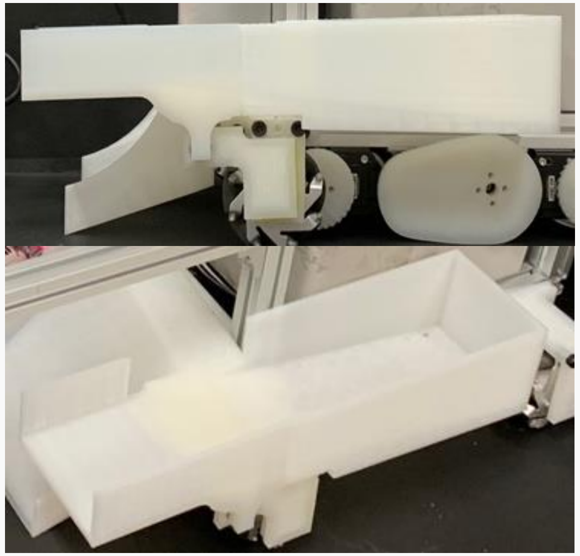
- Size : **41×29×42 [cm]**
- Mass : **6.8kg**

- Gear housing : 9cm
- Body width : 11cm

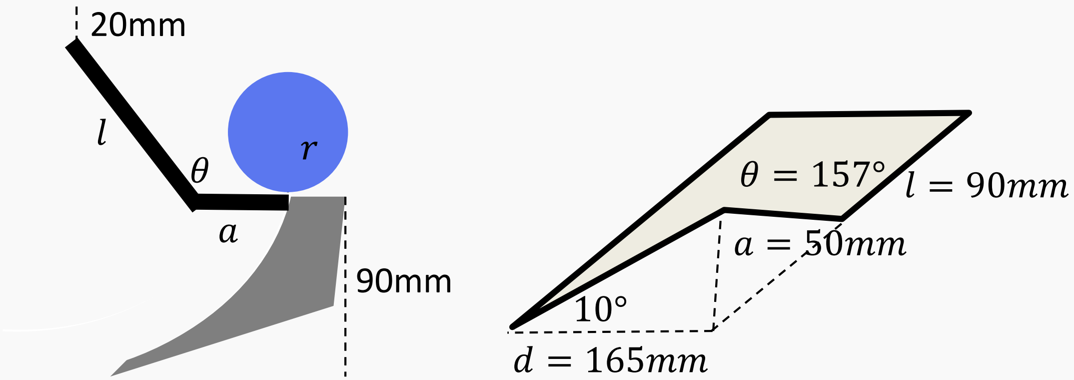
System Design

- **Pick up & Drop Part**

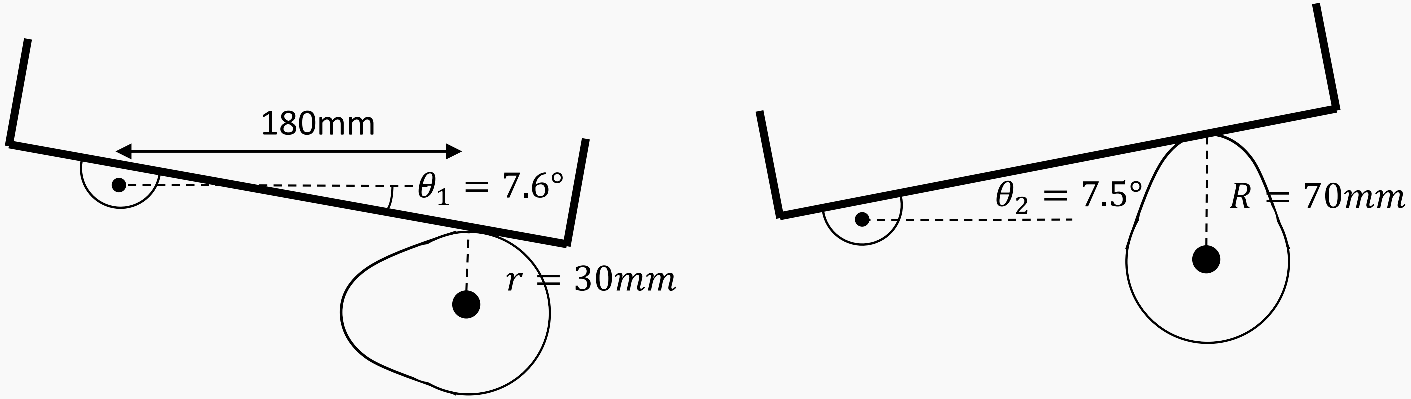




1. Twisted wing*



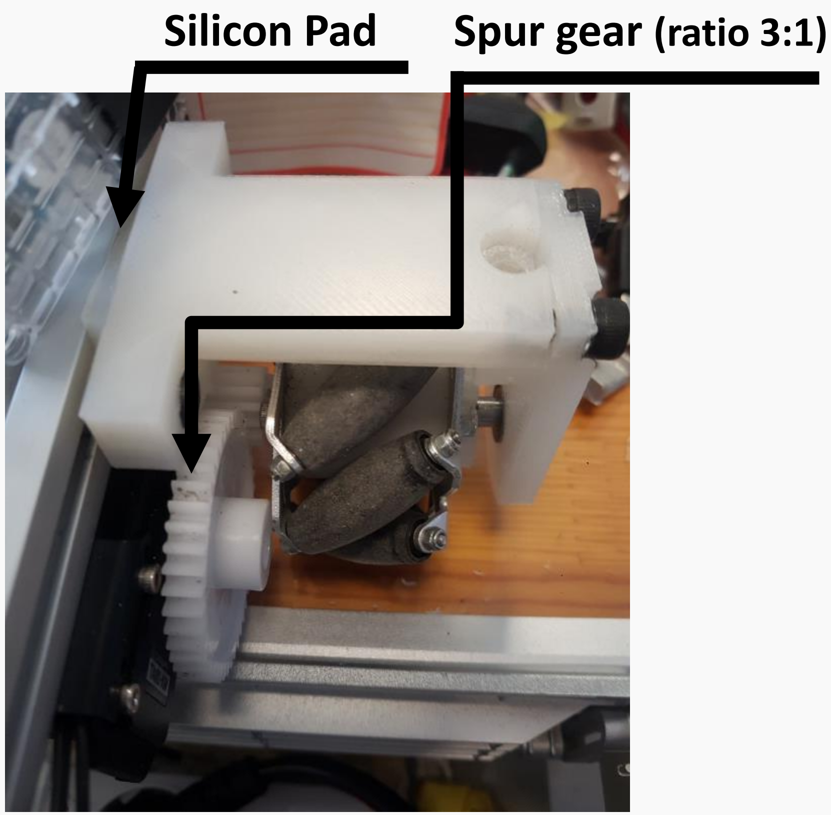
2. Cam & follower*



* Appendix-1, 2

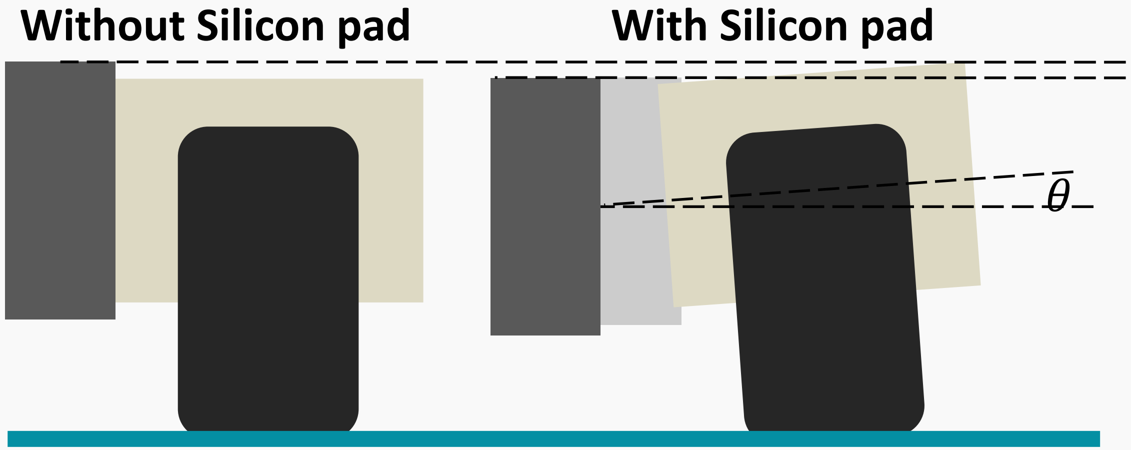
System Design

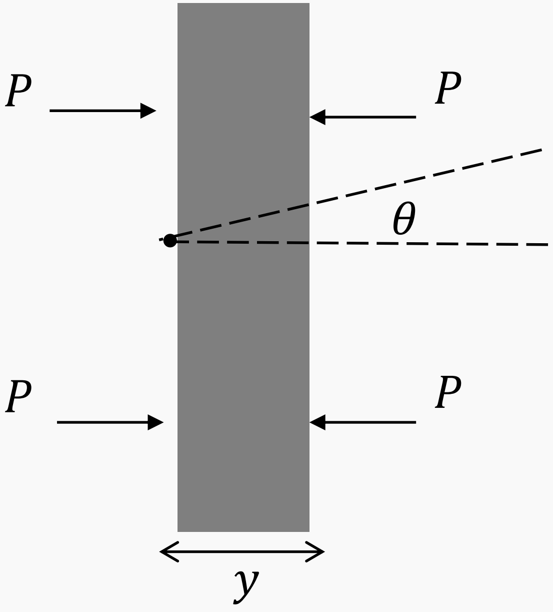
- **Gear Housing**



Attaching viscoelastic material

- Wheels get sufficient DOF
- Make the system more compact





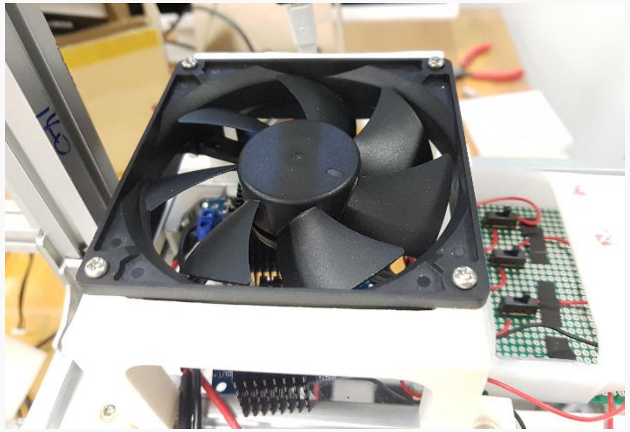
$$I\ddot{\theta} + c\dot{\theta} + k\theta = T$$

$$c = \mu y P$$

$$k = a_{silicon}\frac{A}{l}$$

System Design

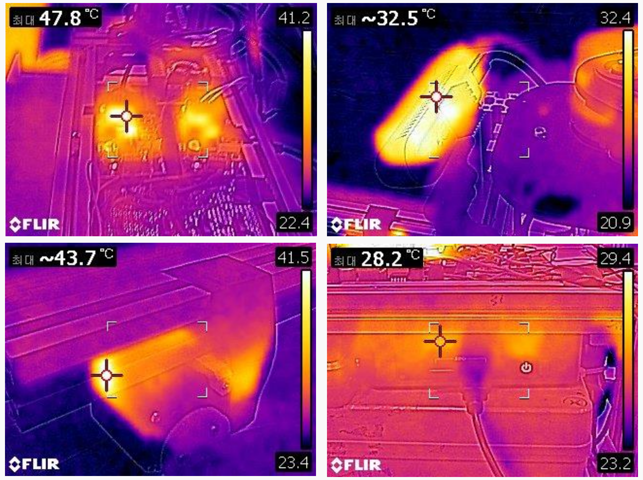
▪ Heat Problem



Cooling Fan

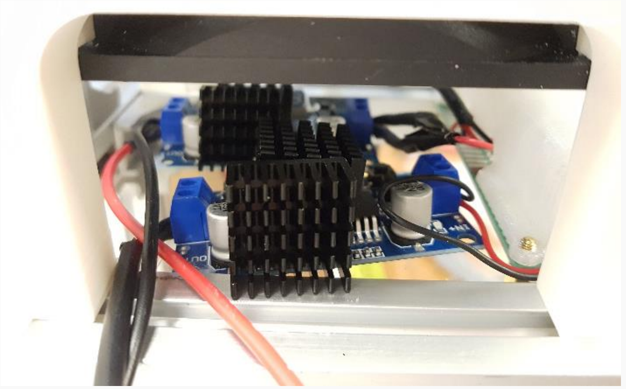
Size : 92×92×25
Input power : 2.16W
Speed : 3200rpm
Maximum air flow :
57.92CFM

▪ Before attaching fan and heatsink (after running once)



Room Temperature = 25 $°C$

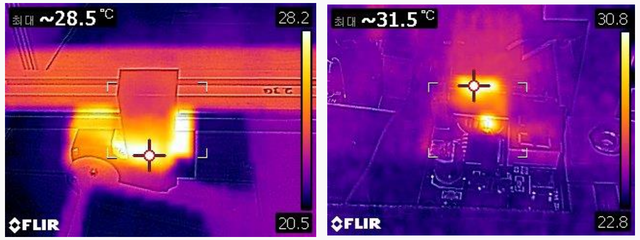
- Converter : $\mathbf{47.8°C}$
- Servo motor : $\mathbf{43.7\ °C}$
- RGB-D camera: 32.5 $°C$
- NUC : 28.2 $°C$



DC converters with Heatsink

Size : 4-20×20×10

▪ After attaching fan and heatsink* (after running once)



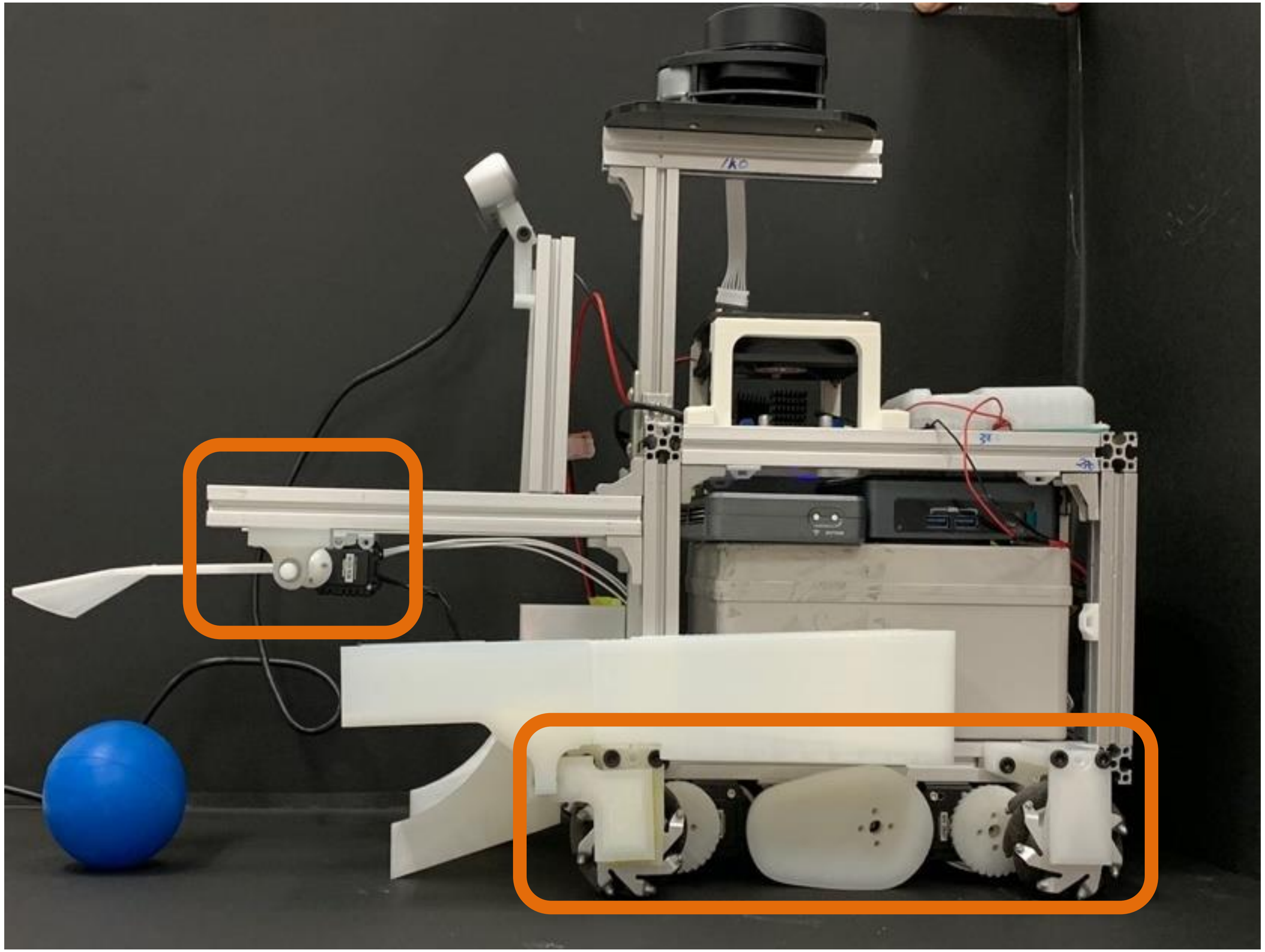
Room Temperature = 25 $°C$

- Converter : $\mathbf{31.5\ °C}$
- Servo motor : $\mathbf{28.5\ °C}$

* Appendix-3

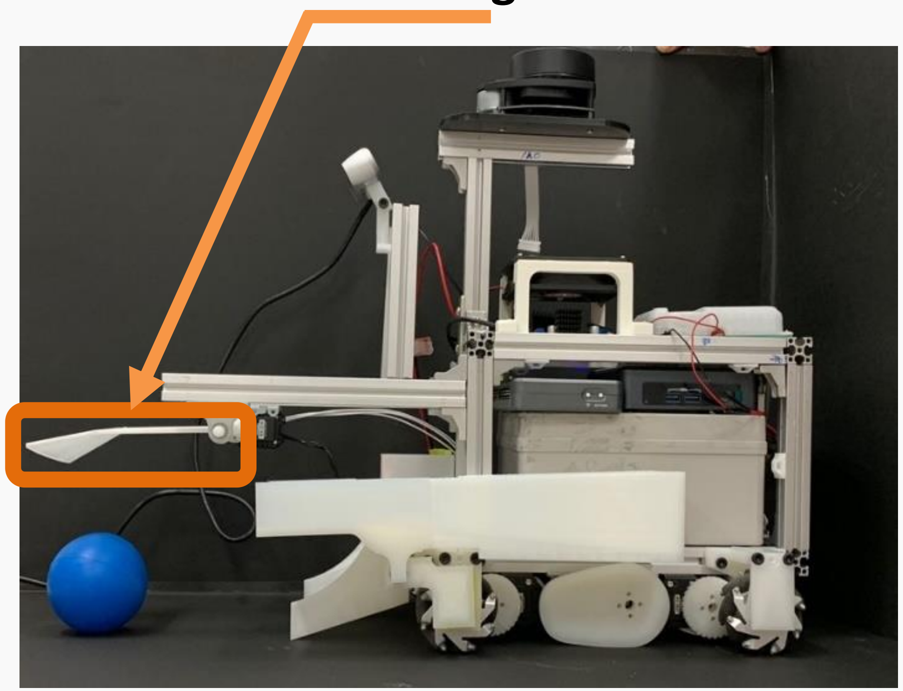
03 - Motor Control

Labview

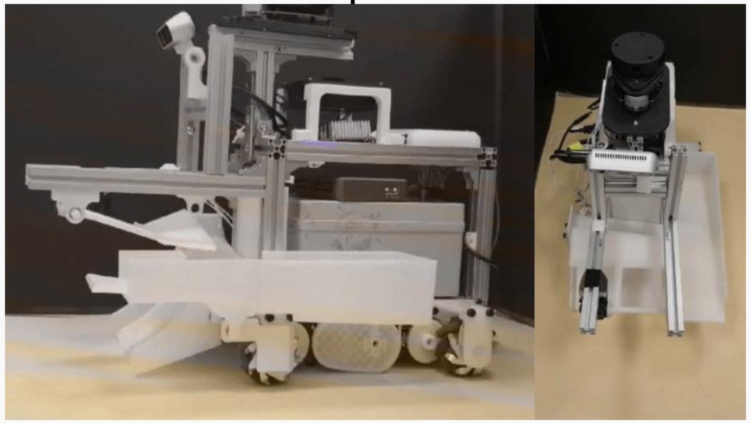


Motor Control

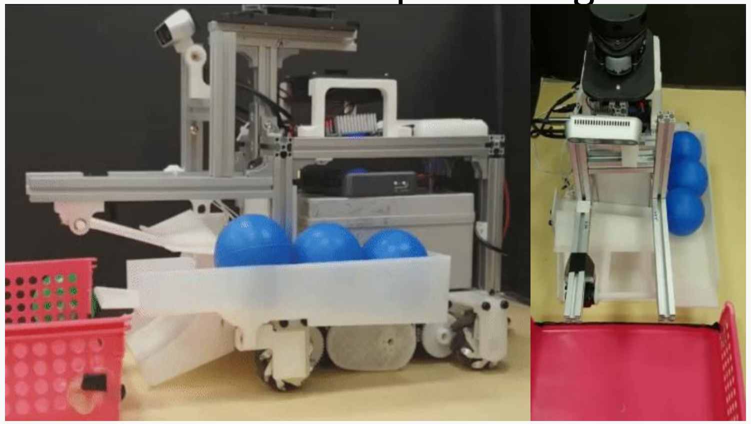
- **Multifunctional Wing**



Pick-up motion



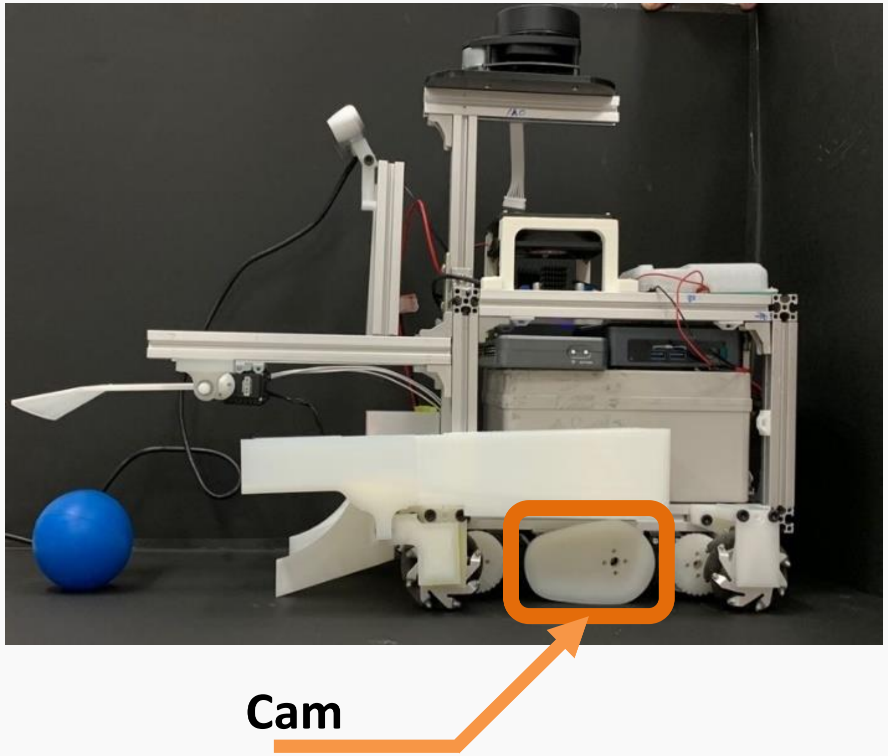
Guide for drop & storage



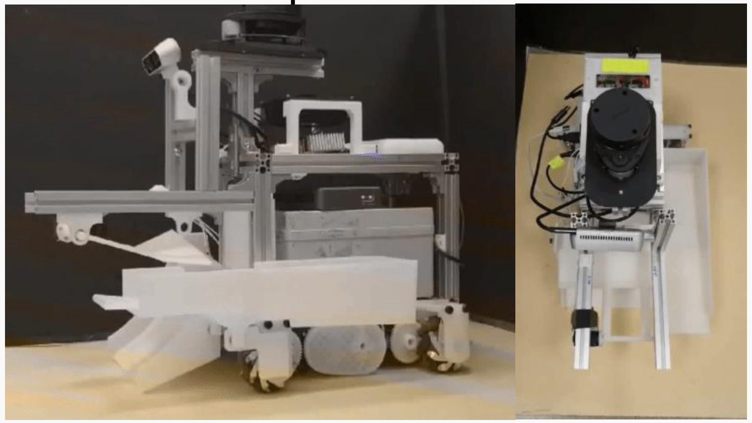
- Wing & cam are used for **both pick-up and drop**
- Make functions **more reliable**, while keeping **compact** design

Motor Control

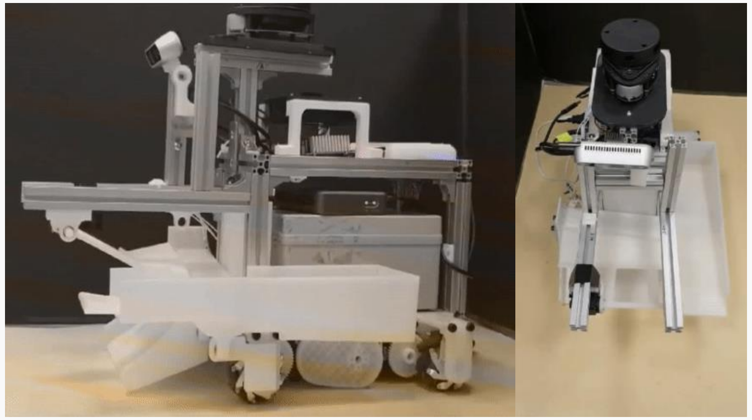
▪ Multifunctional Cam



Drop-off motion



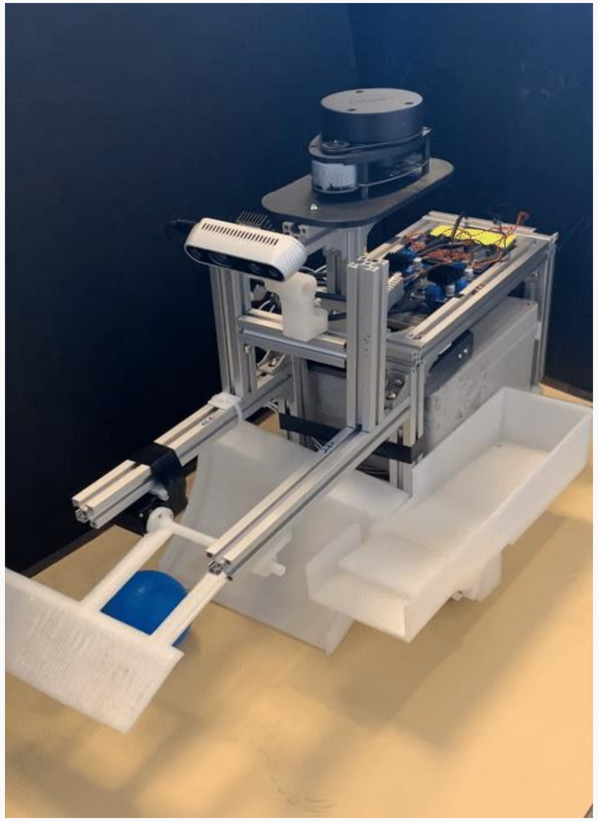
Vibration feeder

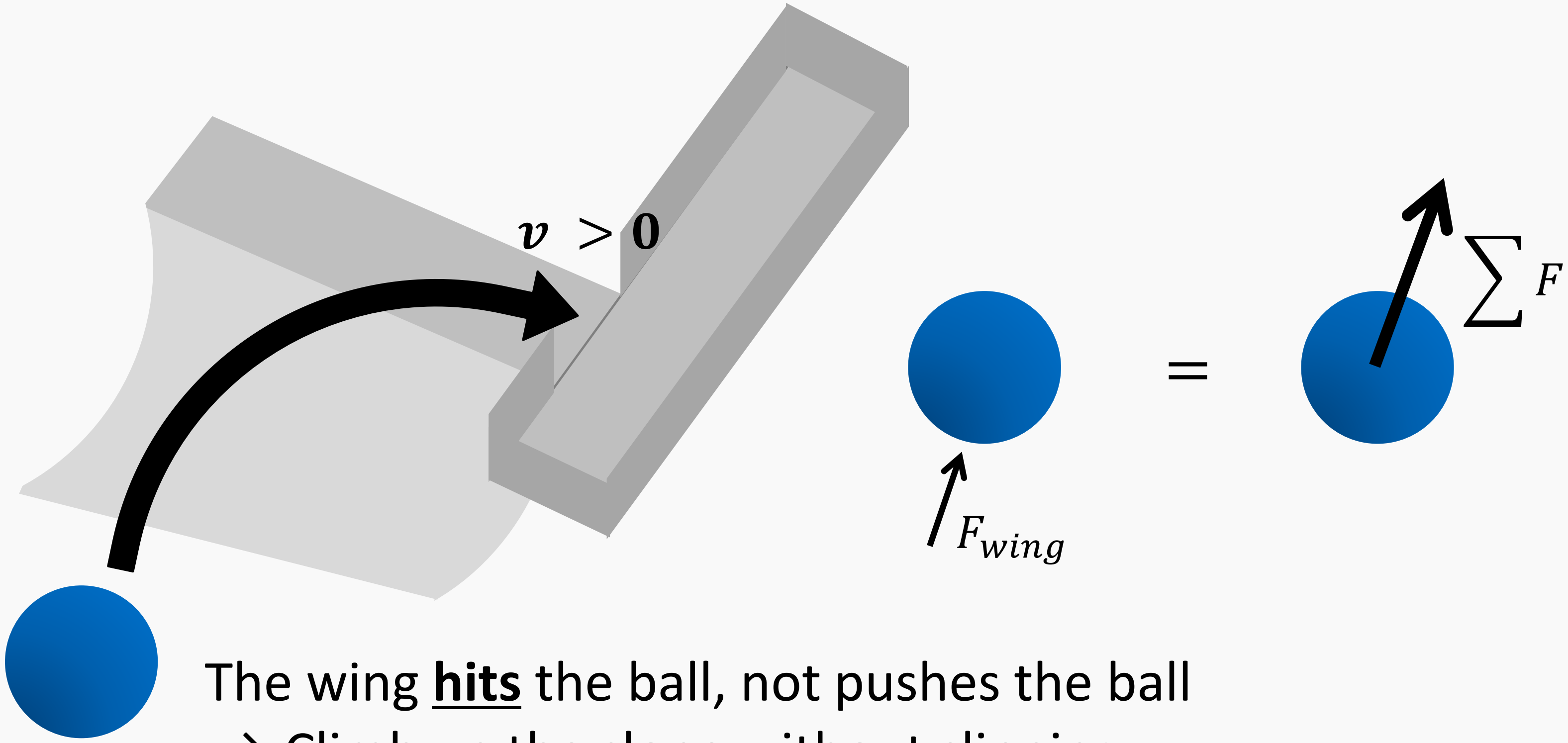


- Wing & cam are used for **both pick-up and drop**
- Make functions **more reliable**, while keeping **compact** design

Motor Control

1. Smart wing control for reliable pickup motion





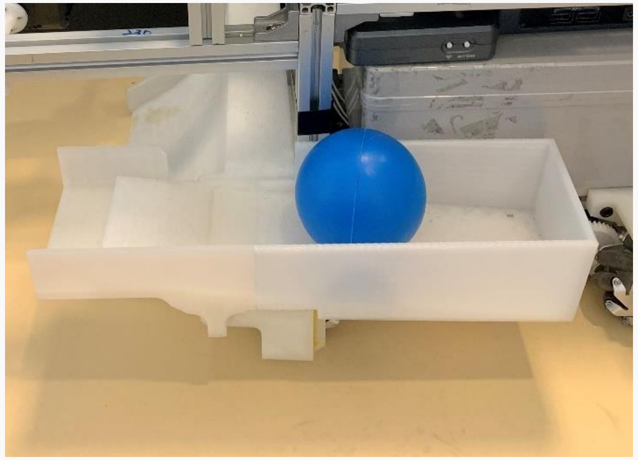
The wing **<u>hits</u>** the ball, not pushes the ball
→ Climb up the slope without slipping
∴ **<u>The ball can have enough velocity to enter the storage</u>**

We achieved more reliable pickup motion!

Motor Control

2. Cam control to overcome the sticking of a ball

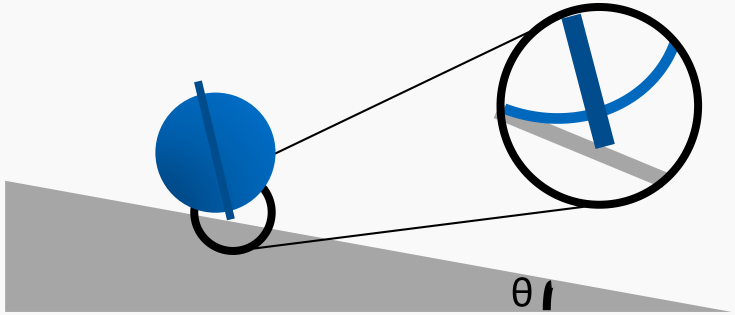
Observation



Ball might stop on the storage

- Blocks another entering ball

Cause



A ball stops due to unevenness*

- To avoid this, tilt angle must be 9.3^o
  → Make the drop-off motion hard

Solution

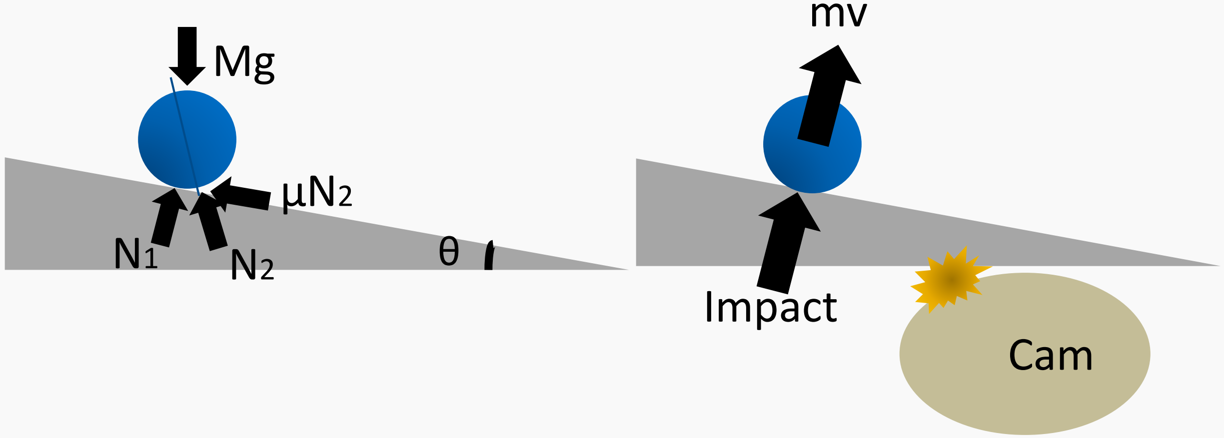
Idea from Vibration feeder

- Repeatedly tap on the storage
  → Overcome the stuck position

* Appendix-4

Motor Control

2. Cam control to overcome the sticking of a ball



Desired value : Cam angular speed N_{cam}

From the law of energy conservation,

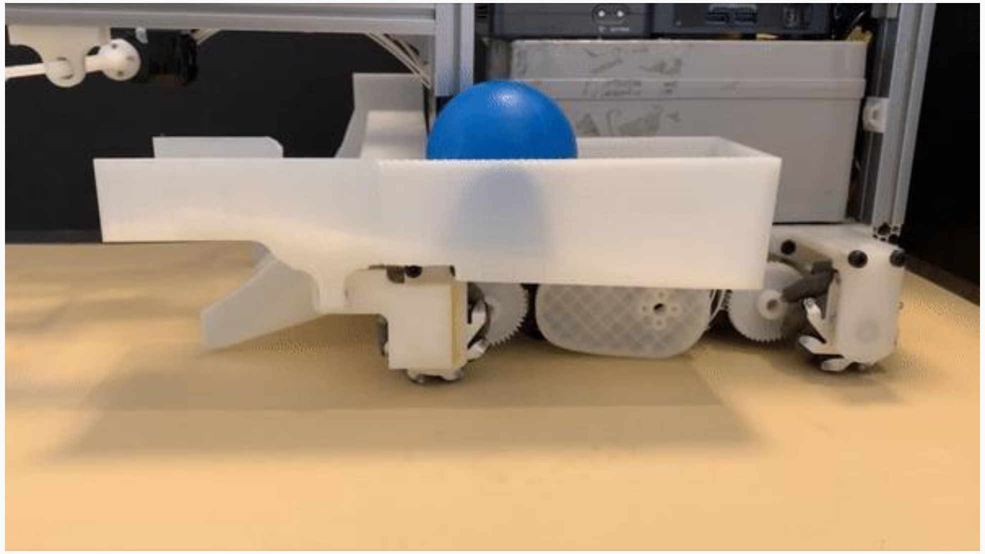
$$V_{ball} = \sqrt{2g\Delta h} = 0.001224\ m/sec$$

By the geometrical relation,

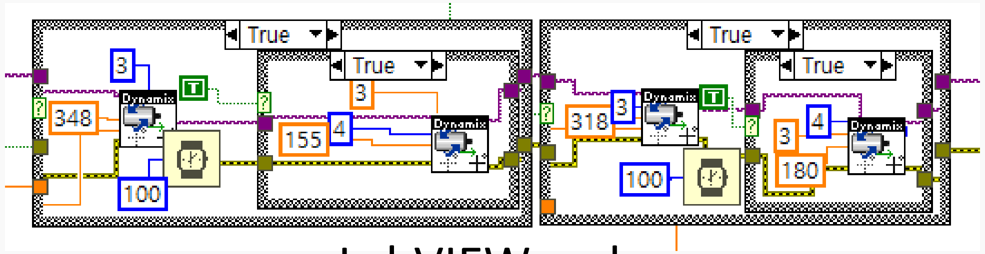
$$V_{ball} = \omega_{cam} r_{cam} = \omega_{sto} r_{sto}$$

$$\therefore\ \boldsymbol{N_{cam} = 1.25\ RPM}$$

※ Empirically, $\boldsymbol{N_{cam} = 3\ rpm}$ works well*



Implementation

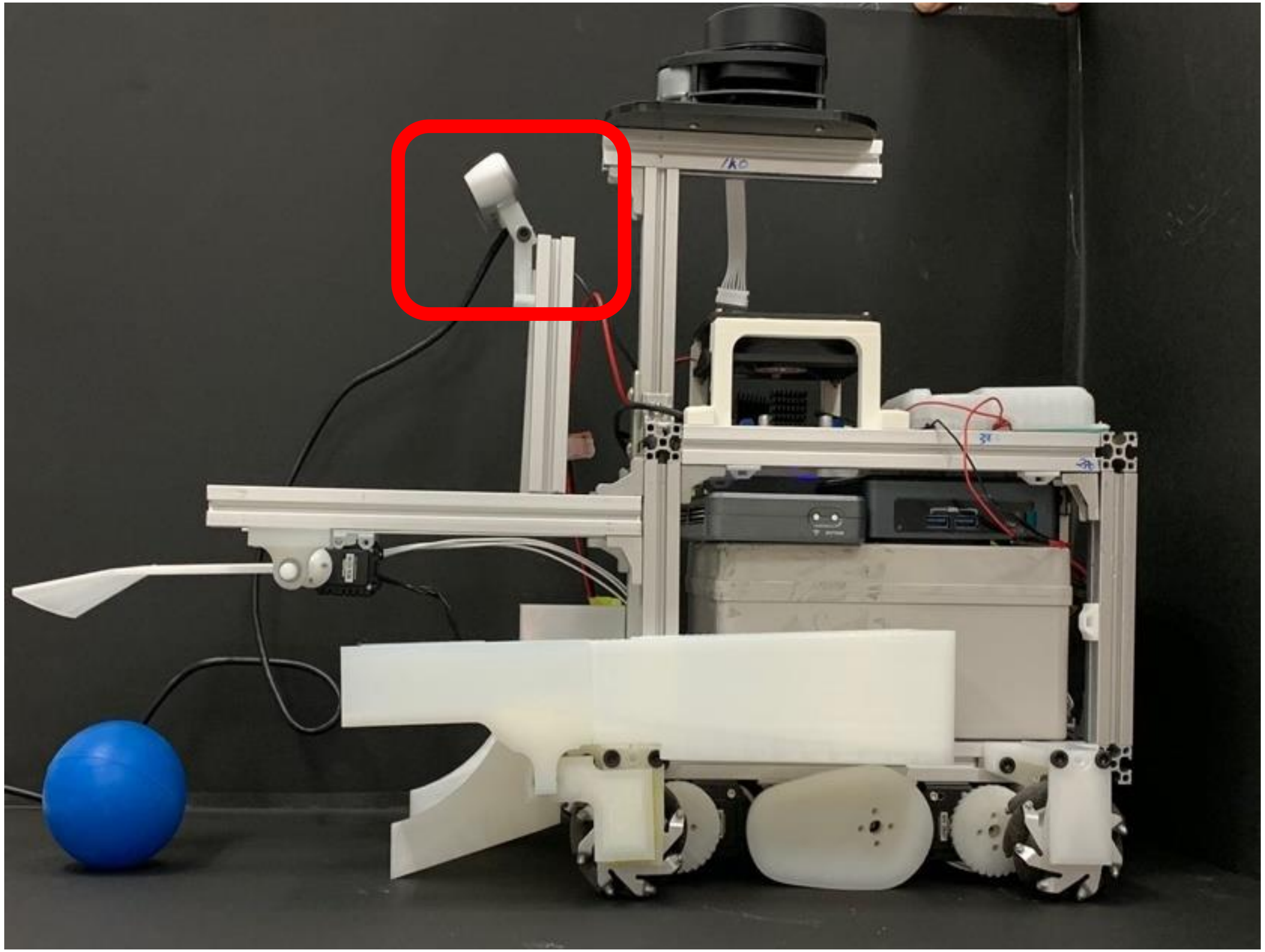


LabVIEW code

* Appendix-4

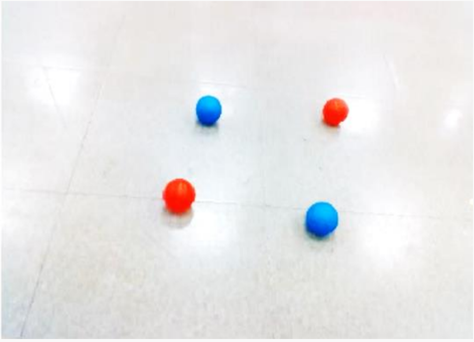
04 - Vision Processing

Open CV

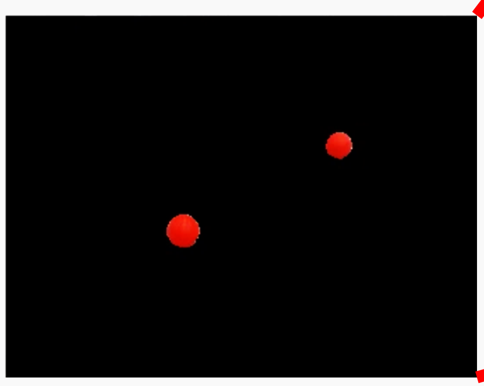


Vision Processing

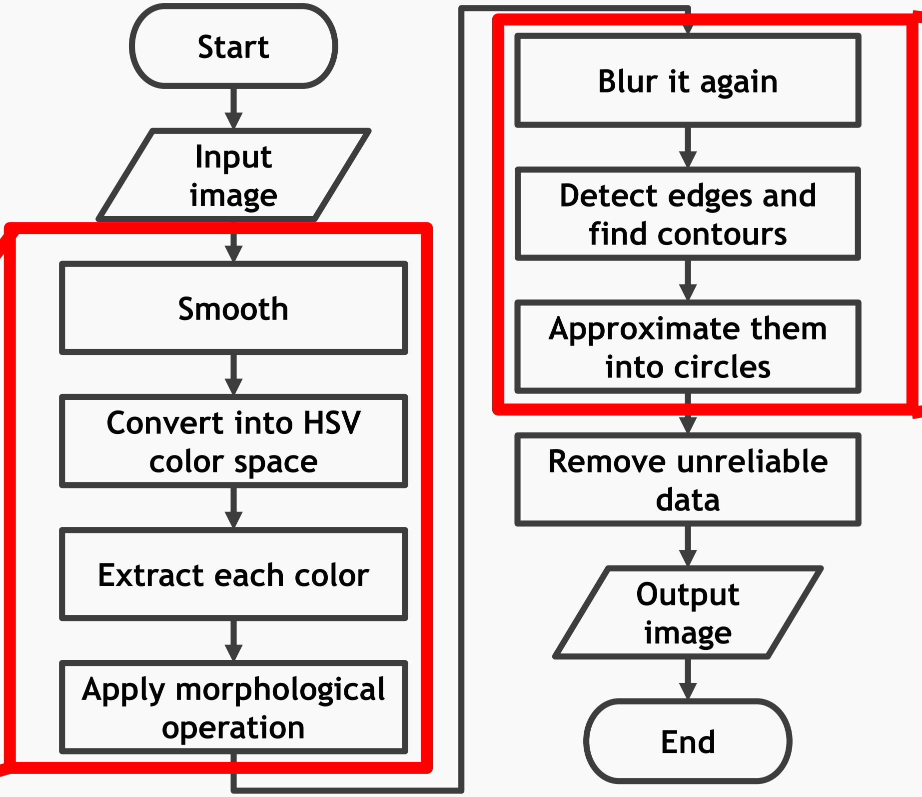
- **Ball Detection**



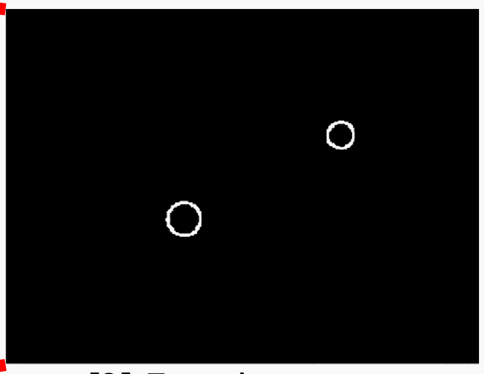
[1] Original



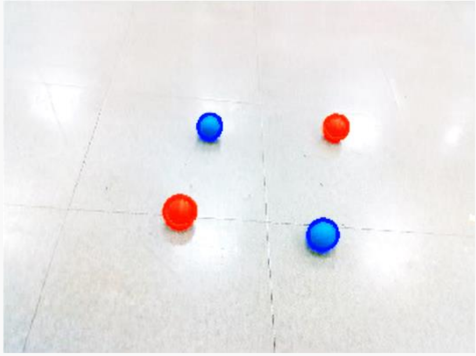
[2] Detected red



Flow diagram of ball detection



[3] Found contours



[4] Result

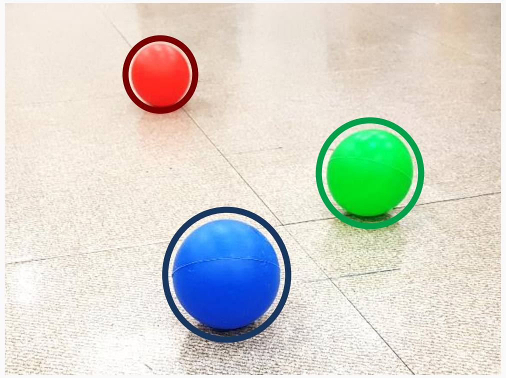
1. Filtered out three different color components in HSV color space.
2. Estimate the objects’ shapes and their center positions

Vision Processing

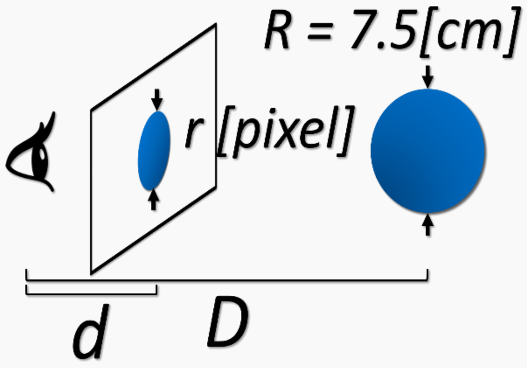
- **Measuring distance using RGB-D camera**

Given webcam

RGB Image



Scheme of calculating D



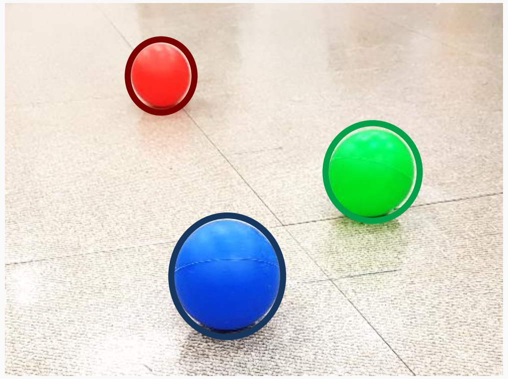
$$D = \frac{r}{R} d$$

→ To ensure D, r **should be accurate**

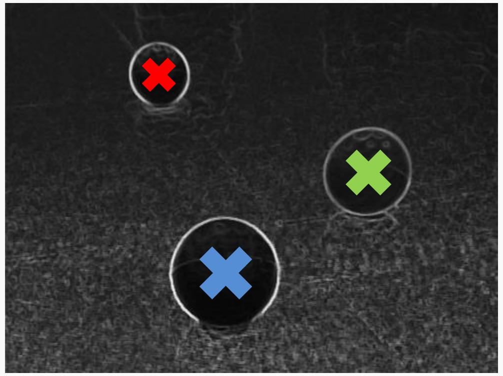
- "Calculate" a ball's distance
- Vulnerable to vibration and occlusion

Using RGB-D camera

RGB Image



Depth Image



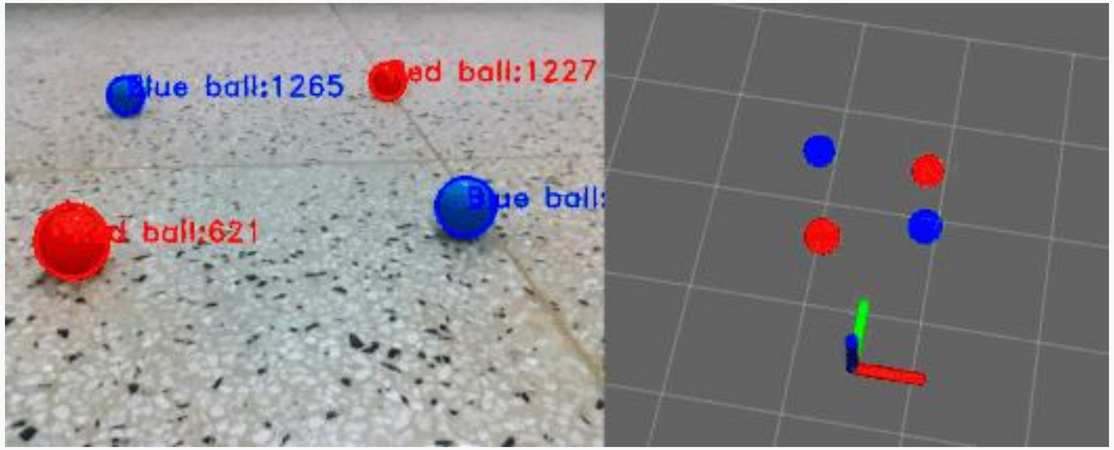
$$D = I_{depth}(x, y)$$

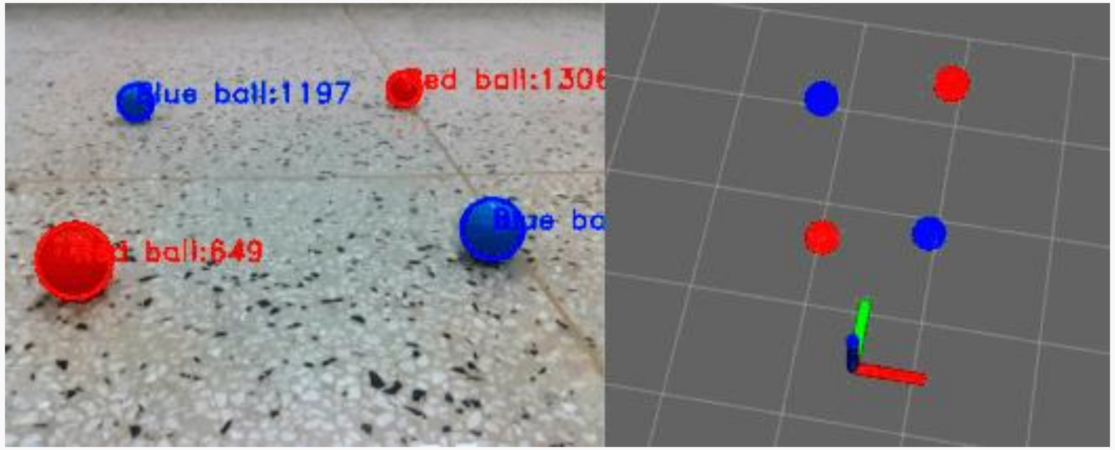
→ **Does not have to rely on r**

- Precisely "measure" a ball's distance
- Robust to vibration and occlusion

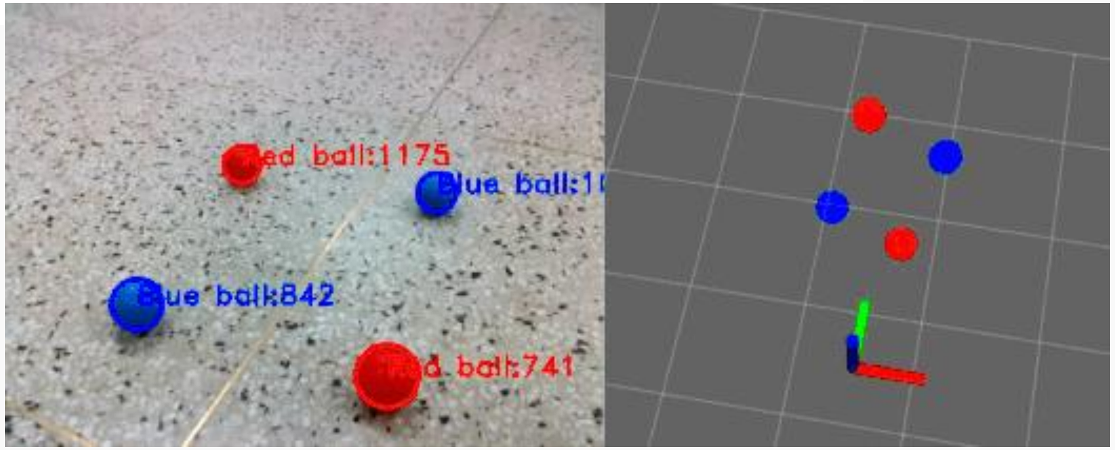
Vision Processing

- **Performance evaluation of using RGB-D camera**





Occlusion



Vibration

Given webcam

Highly affected by occlusion/Vibration

RGB-D camera

Occlusion/vibration <u>hardly affect</u>

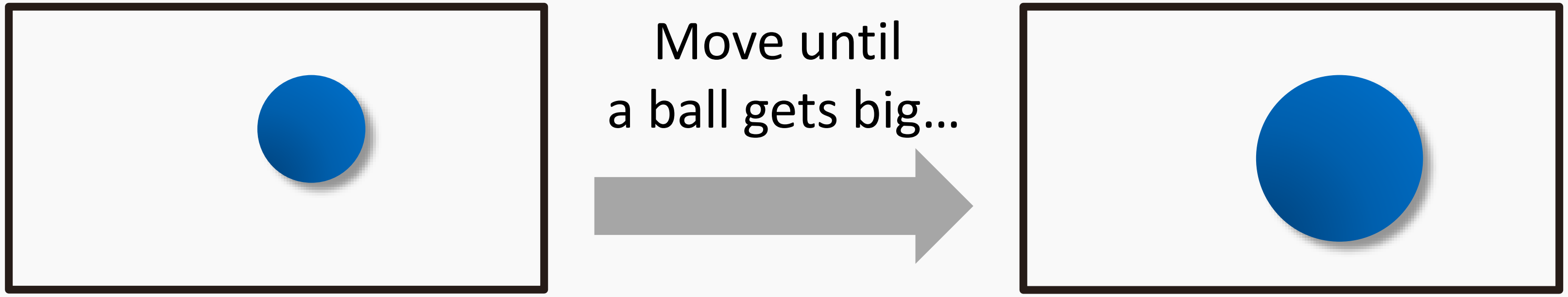
Vision Processing

▪ Extrinsic Calibration

The simplest way to command a robot is…

1. Align a ball

2. Move forward



→ Not sophisticated…

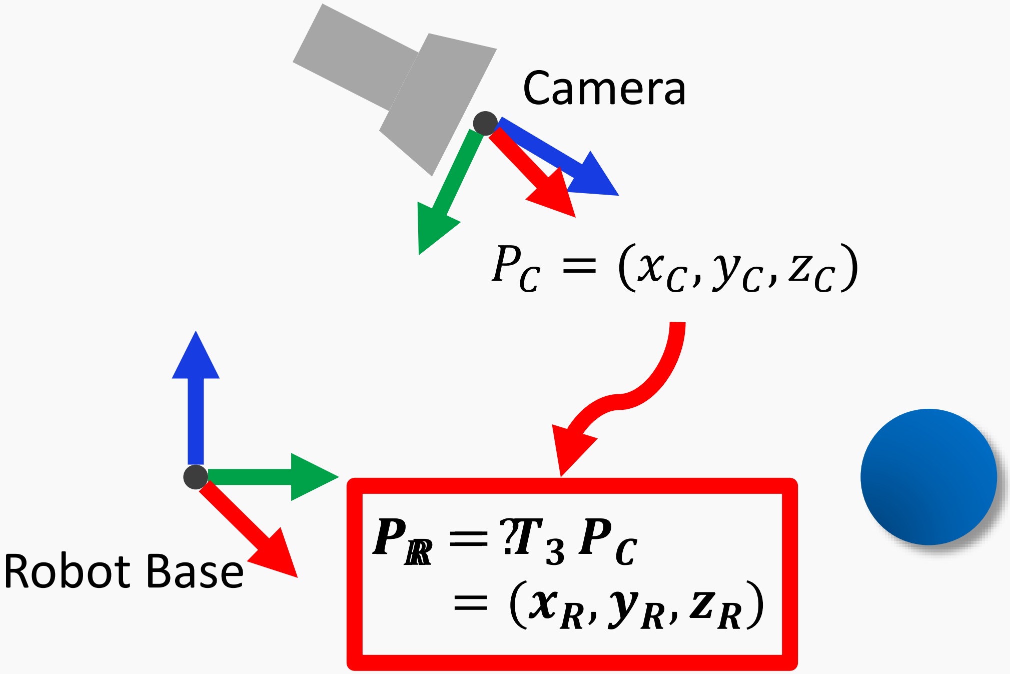
Can we command a ball’s position **with respect to the robot**?

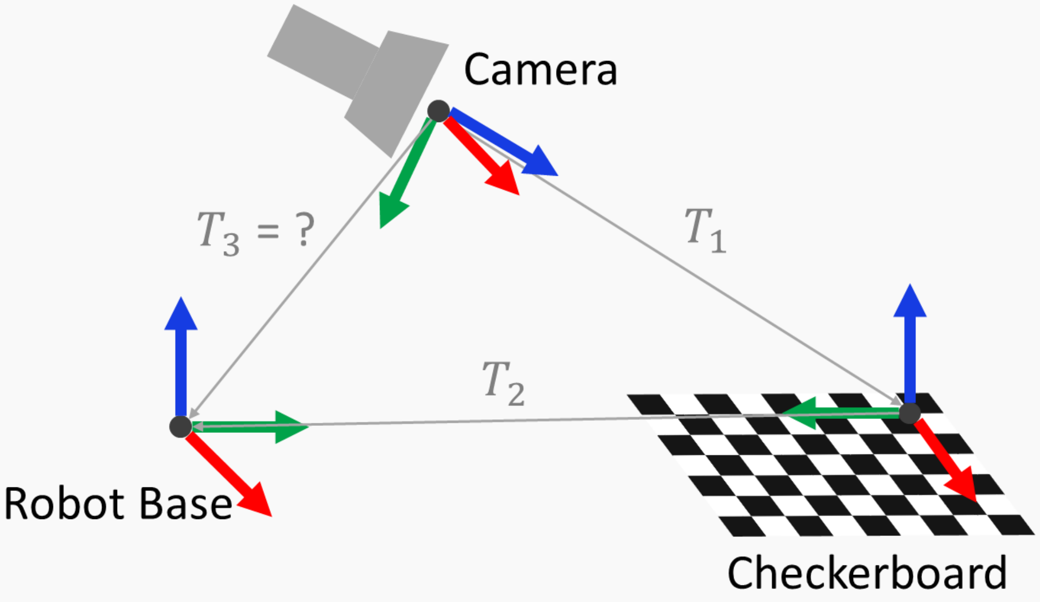
Vision Processing

▪ Extrinsic Calibration

Calibrate between the camera and the base of the robot via a checkerboard

- Guarantees a ball's location in conjunction with reliable range information from the RGB-D camera (~$1cm$ within $1m^2$)
- A ball's position with respect to the robot → **Ensures more intuitive & smart motion**





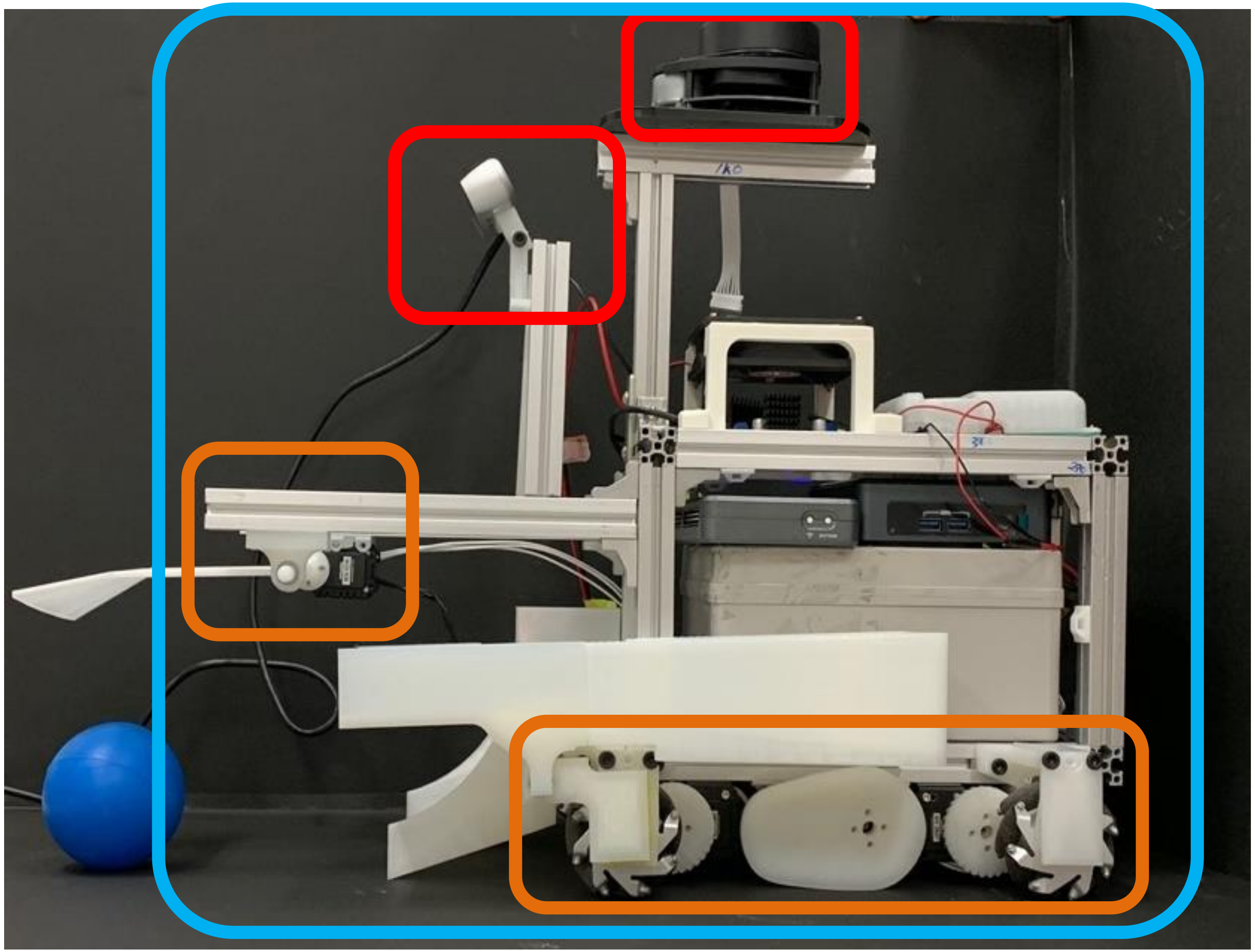
$$T_3 = T_1 T_2 \qquad P_R = T_3 P_C$$

05 - System Integration

Labview

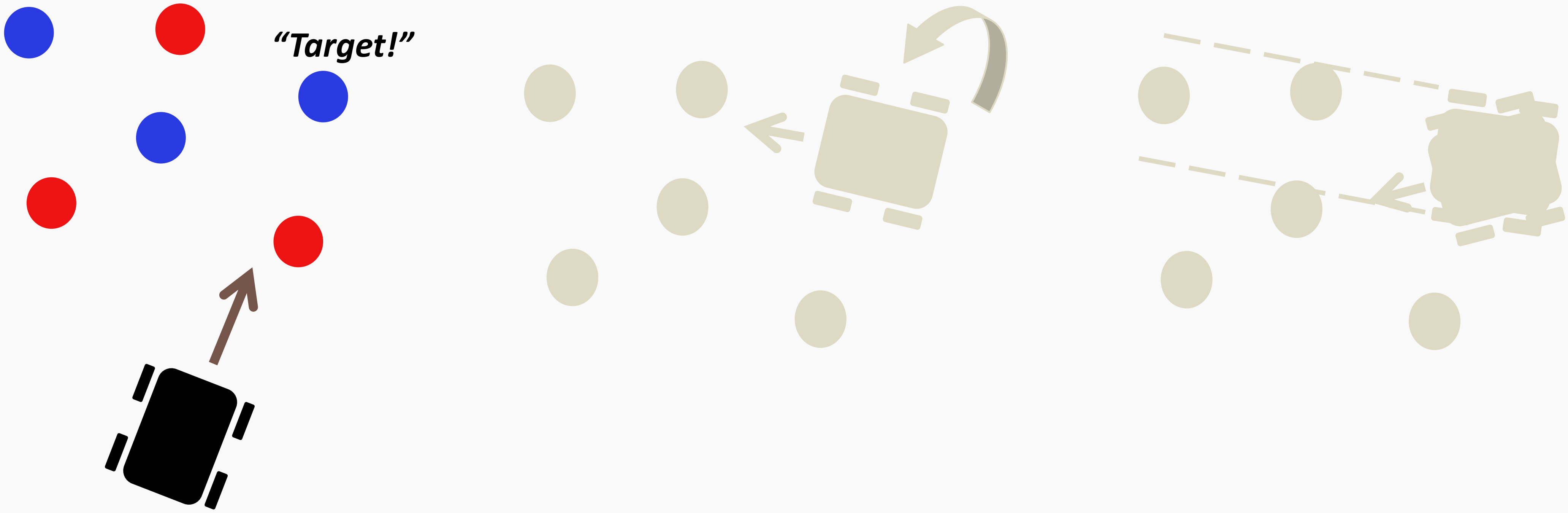
ROS

Open CV



System Integration

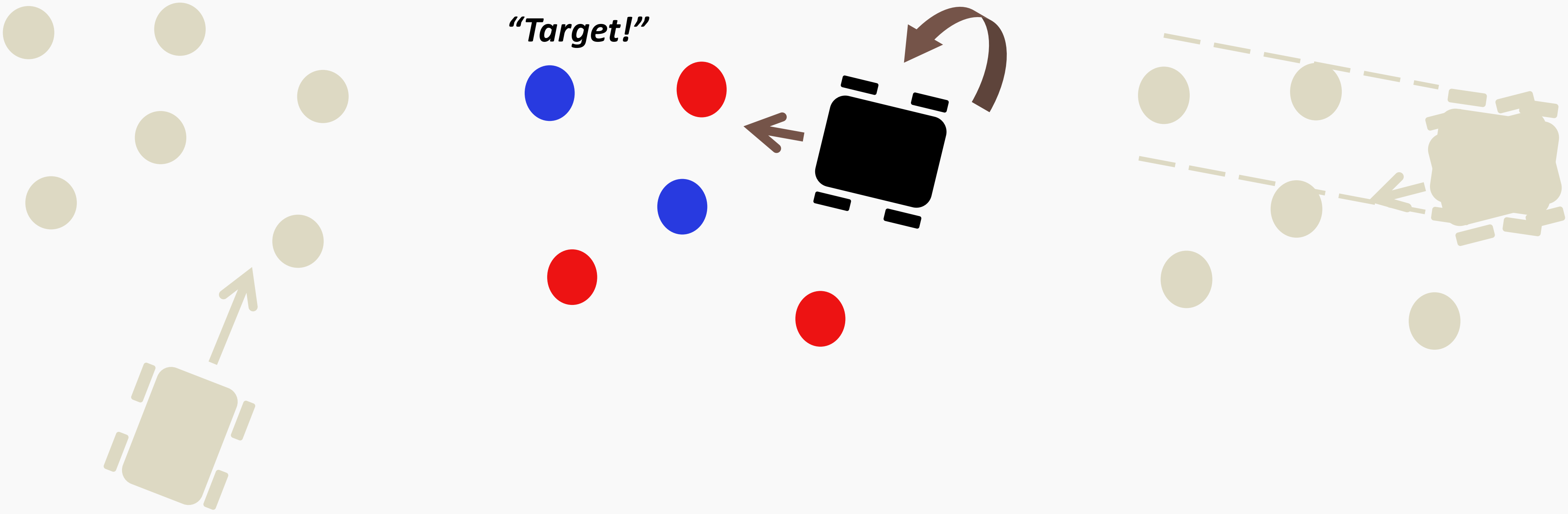
- **Path finding algorithm – Overview**



Rule 1 . If no exception, Target the **rightmost** ball

System Integration

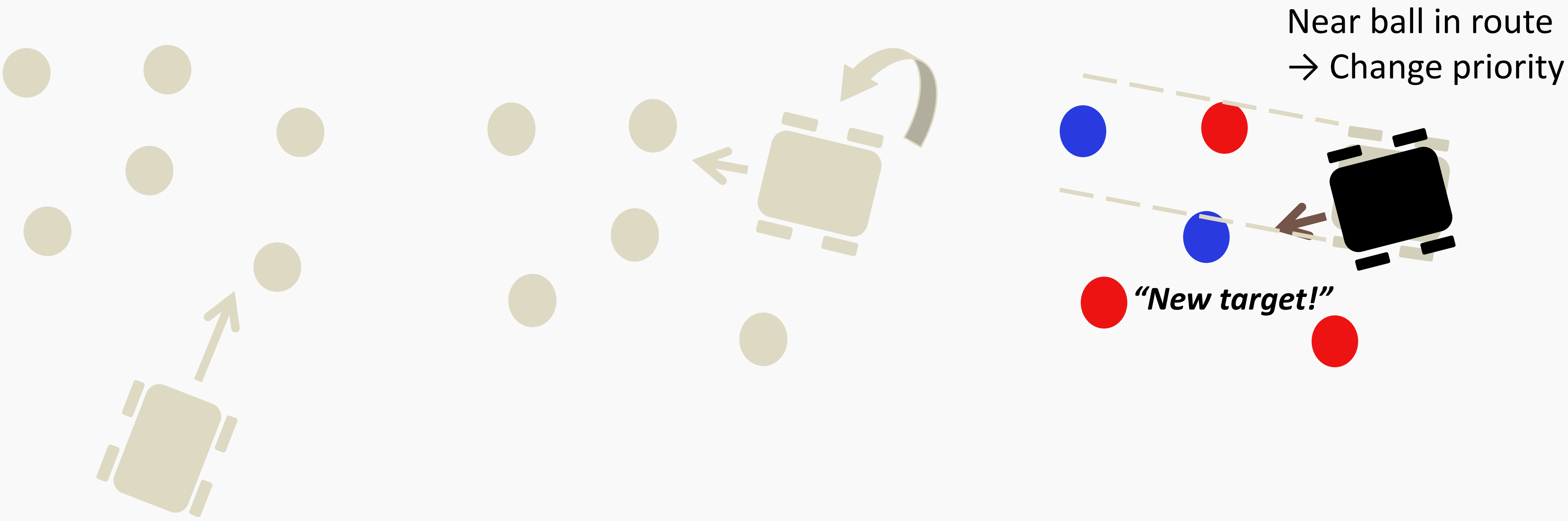
- **Path finding algorithm – Overview**



Rule 2 . If no exception, turn **CCW** to find next target

System Integration

- **Path finding algorithm – Overview**



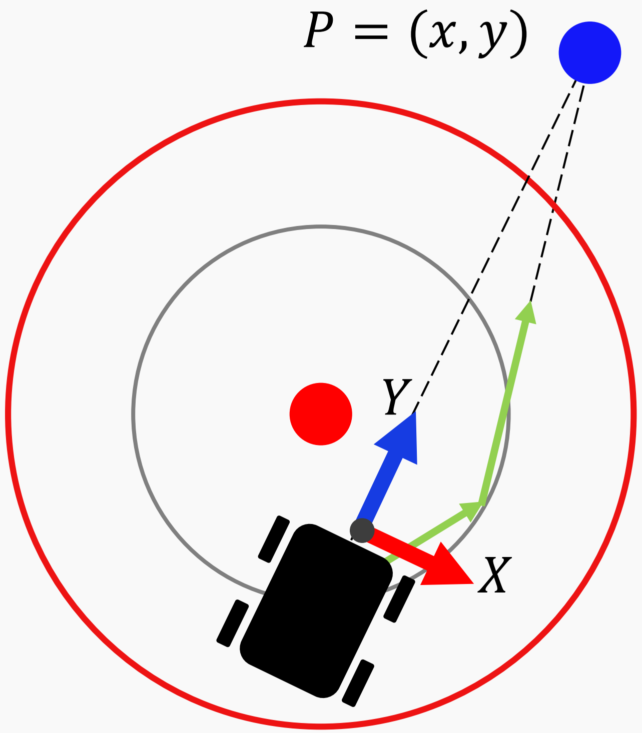
Exception : When another near blue ball is in route…

Rule 3 : If an exception happens, **change the priority** to near ball

Rotate **CW** to find next ball (so-called *Checker Method*)

System Integration

- **Path finding algorithm – Detouring Red Ball**



*No other balls inside red area (50cm boundary)

How to detour red?

- Move along inscribed isosceles triangles
- Using 2D coordinate & by its minimum width, **rotate minimum angle**

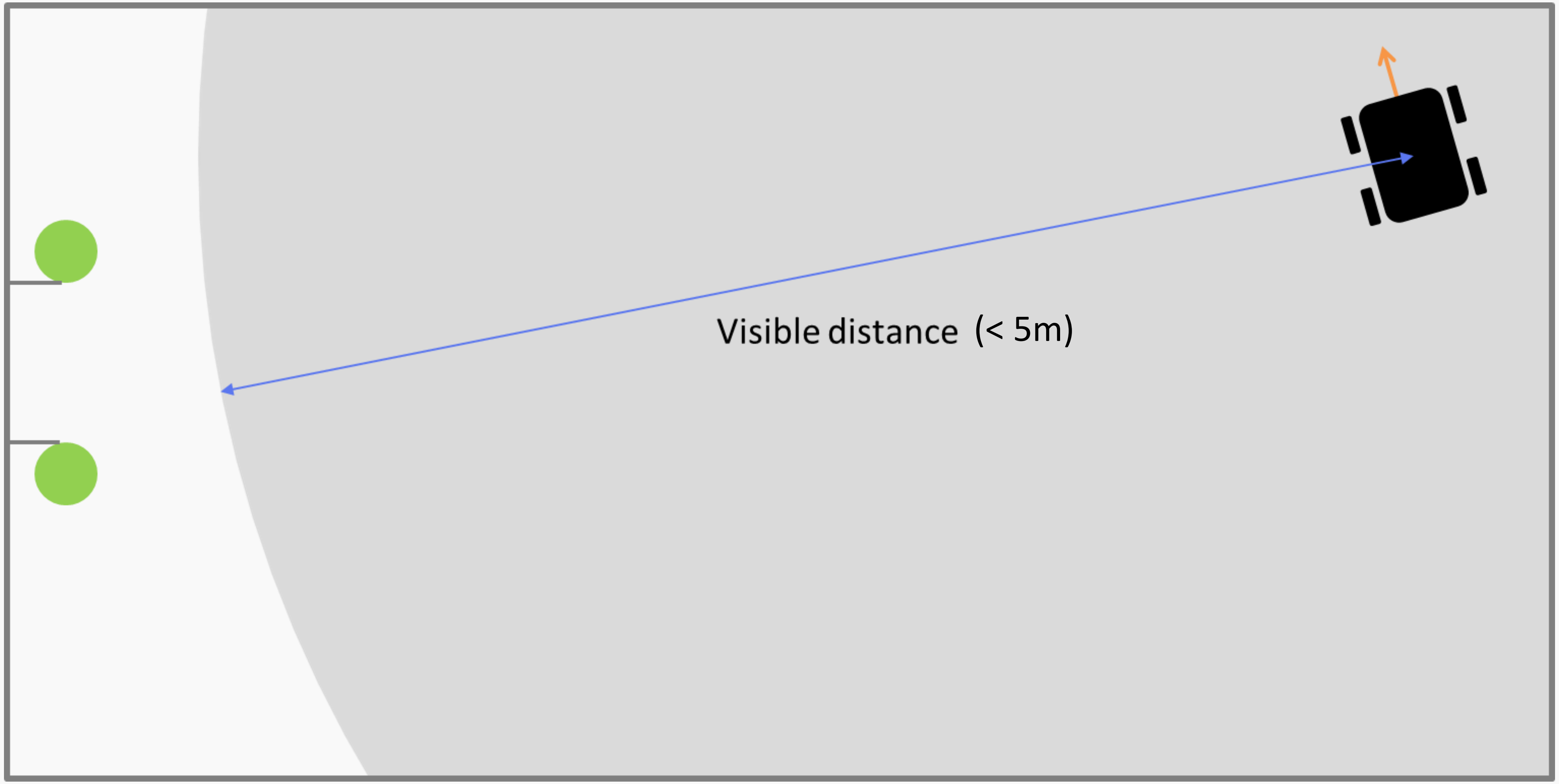
Advantages

- Does not affect other balls while detouring
- Collect blue balls no matter where red balls are at
  ∴ **Reliable & smart motion**

System Integration

- **Return to the starting point**

Problem when using camera only : it may not see the green balls



System Integration

- **Return to the starting point**

Solution : **Use rpLidar** to find the exact location of the goal*

1. Find corners

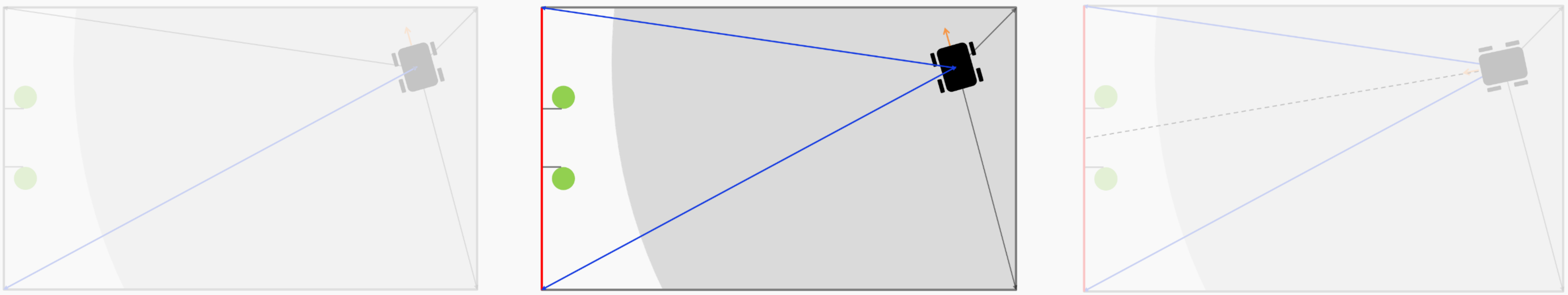
- Using lidar range data, find corners and their distances
- Choose the farthest corner

* Appendix-5

System Integration

▪ Return to the starting point

Solution : **Use rpLidar** to find the exact location of the goal*



2. Find the goal

- Calculate the side length using cosine law
- The goal is on the shorter side → we can find the goal position

* Appendix-5

System Integration

- **Return to the starting point**

Solution : **Use rpLidar** to find the exact location of the goal*

3. Align toward the goal

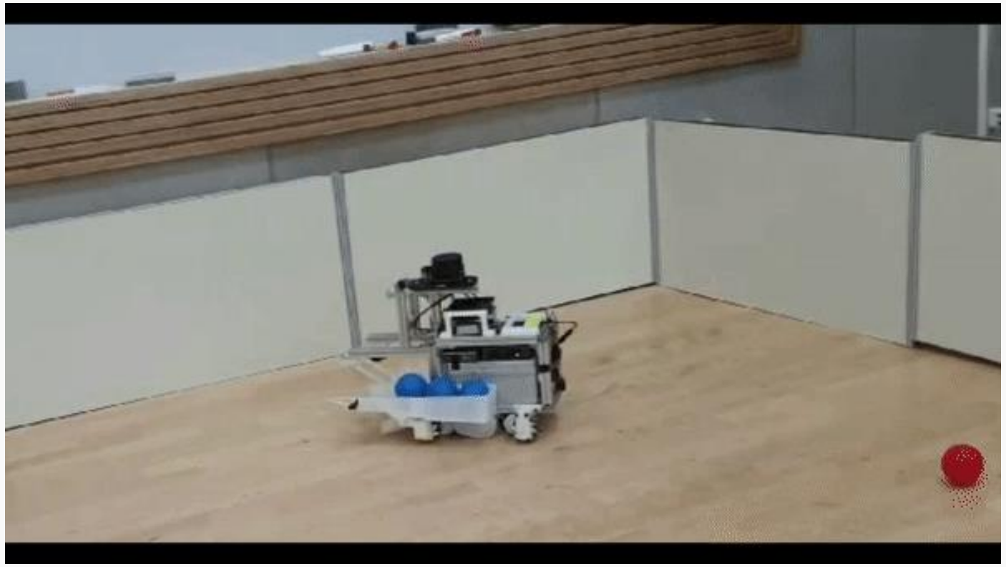
- Turn toward the goal position
- During the process, repeat 1-2 until with feedback control

* Appendix-5

System Integration

- **Return to the starting point**

Solution : **Use rpLidar** to find the exact location of the goal*



3. Align toward the goal

- Turn toward the goal position
- During the process, repeat 1-2 until with feedback control

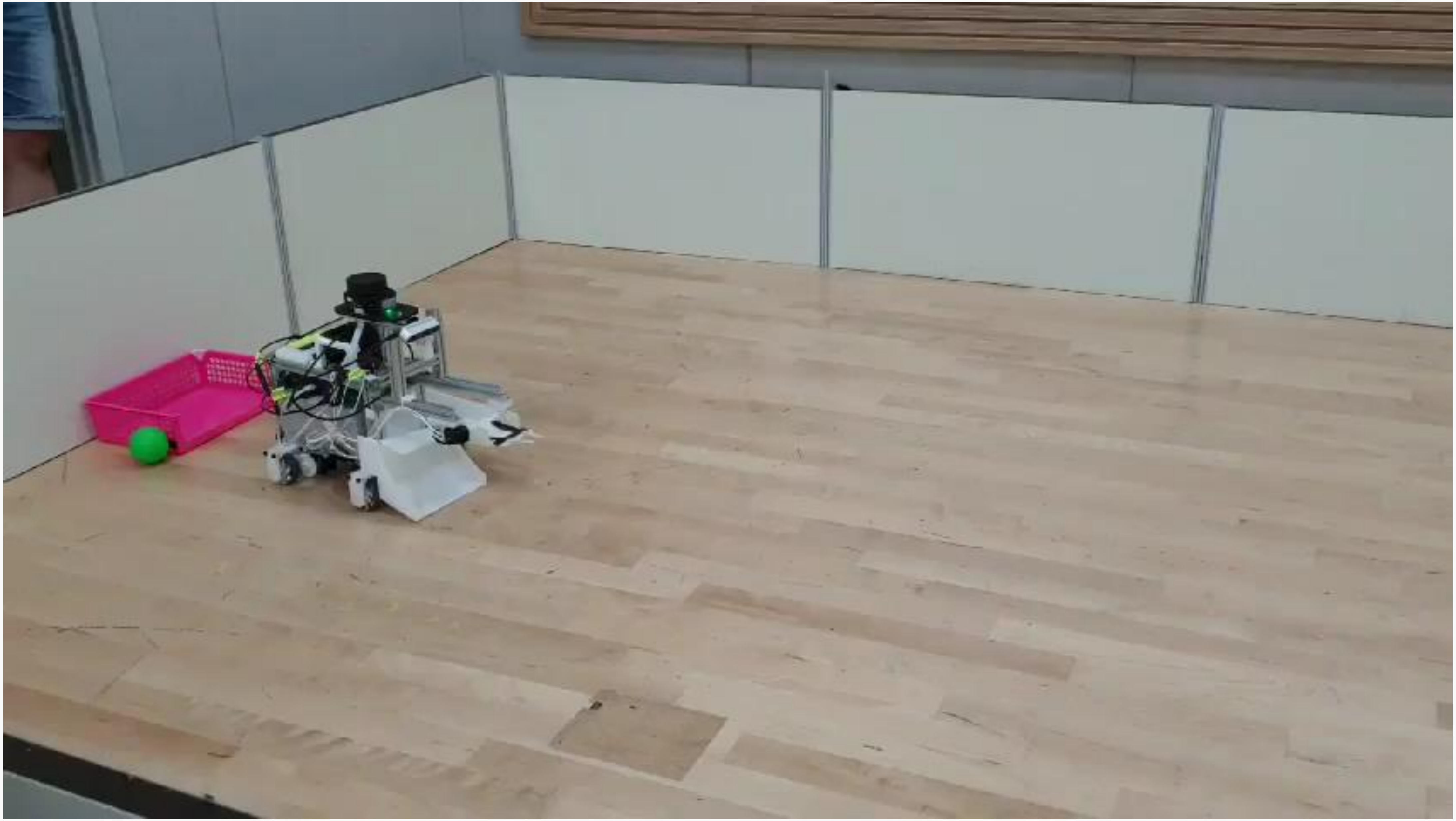
* Appendix-5

06

Demo Video

Demo Video – The Fastest Case

Demo Video – The Hardest Case

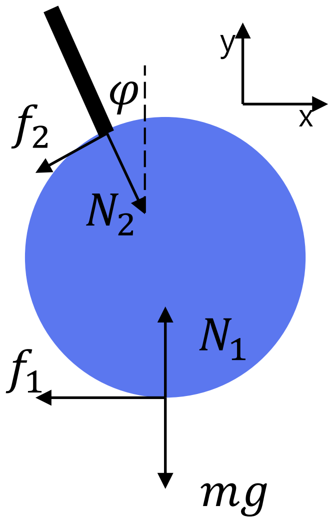


Q&A

Appendix-1 : Pick-up Mechanics

Why the ball should be located inside the wing

FBD of ball



$f_1 = \mu_1 N_1,\ f_2 = \mu_2 N_2$
(μ_1 : friction coefficient of ball on concrete floor, μ_2 : friction coefficient of ball on PLA)

$$\sum F_x = N_2 \sin\varphi - f_1 - f_2 \cos\varphi$$

$$\sum F_y = N_1 - mg - N_2 \cos\varphi - f_2 \sin\varphi$$

$$\sum T = r f_2 - r f_1$$

If ball is fastened when Φ is 0<Φ<π, it should satisfy these following conditions $\sum F_x = 0, \sum F_y = 0, \sum \mathrm{T} = 0$

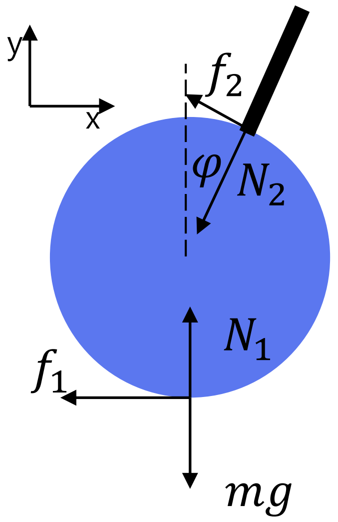
When you solve this equation, answer is $\varphi = \frac{1-\mu_2{}^2}{1+\mu_2{}^2}$, $N_2 = \frac{mg}{\frac{\mu_2}{\mu_1} - \cos\varphi - \mu_2 \sin\varphi}$.

However, this is a contradiction in specifying the magnitude of the force applied to the ball from the wing. Therefore, the case of ball located inside the wing is not fastened.

Appendix-1 : Pick-up Mechanics

Why the ball should not be located outside the wing

FBD of ball



$f_1 = \mu_1 N_1,\ f_2 = \mu_2 N_2$
(μ_1 : friction coefficient of ball on concrete floor, μ_2 : friction coefficient of ball on PLA)

$$\sum F_x = -N_2 \sin\varphi - f_1 - f_2 \cos\varphi$$

$$\sum F_y = N_1 - mg - N_2 \cos\varphi + f_2 \sin\varphi$$

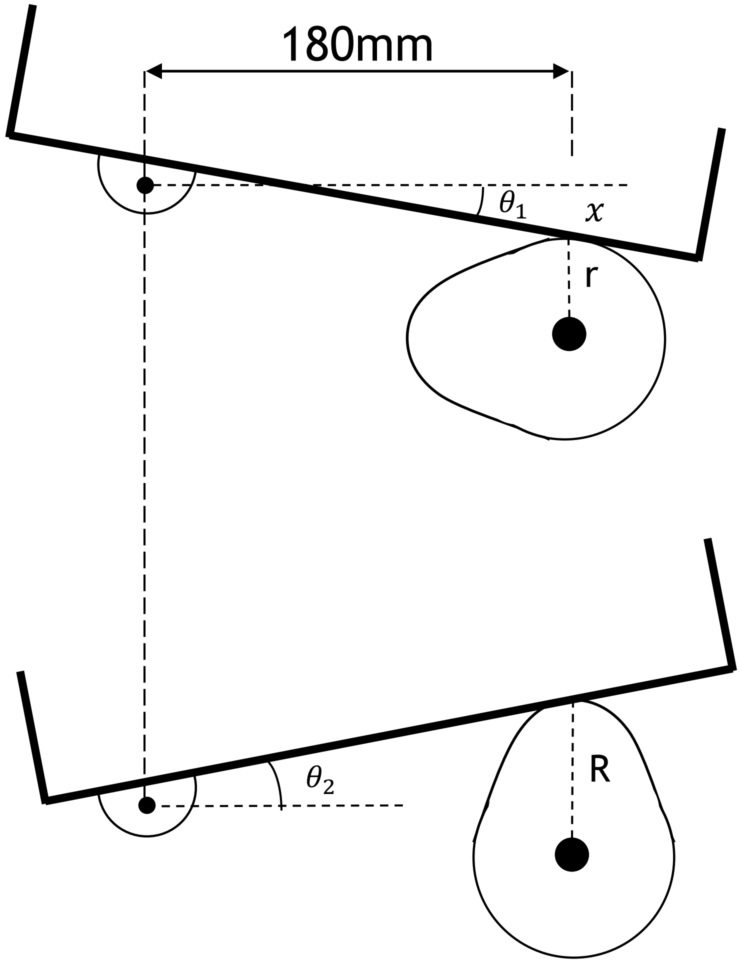
$$\sum T = r f_2 - r f_1$$

$\sum F_x$ is always negative so the ball move to –x direction.
This makes that we can not eat the ball.

Therefore, the ball should not be located outside the wing.

Appendix-2 : Cam & Follower Motion

Before drop



After drop

$$r = 180 \tan\theta_1 - x$$
$$\theta_1 = 7.6°$$

$$r = 30mm,\ x = 6mm$$

$$R = 180 \tan\theta_2 + x + r$$
$$\theta_2 = 7.5°$$

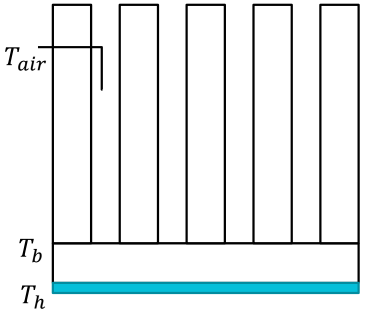
$$R = 70mm$$

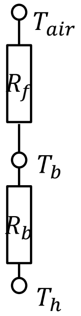
Appendix-3 : Heat Problem

Converter is known as the primary heat source *

NCP1529 – 1 A DC to DC converter release maximum 720mW

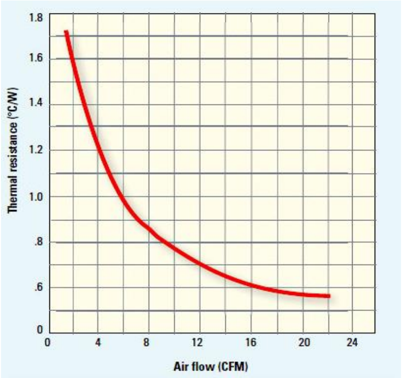
3cm X 3cm Aluminum Heatsink with>16CFM can cover 731mW





Converter

Diagram of heatsink



Air flow – thermal resistance graph for 5’’ X 5’’ Aluminum Heatsink **

Our goal : below 35° C
(room temperature 27° C)

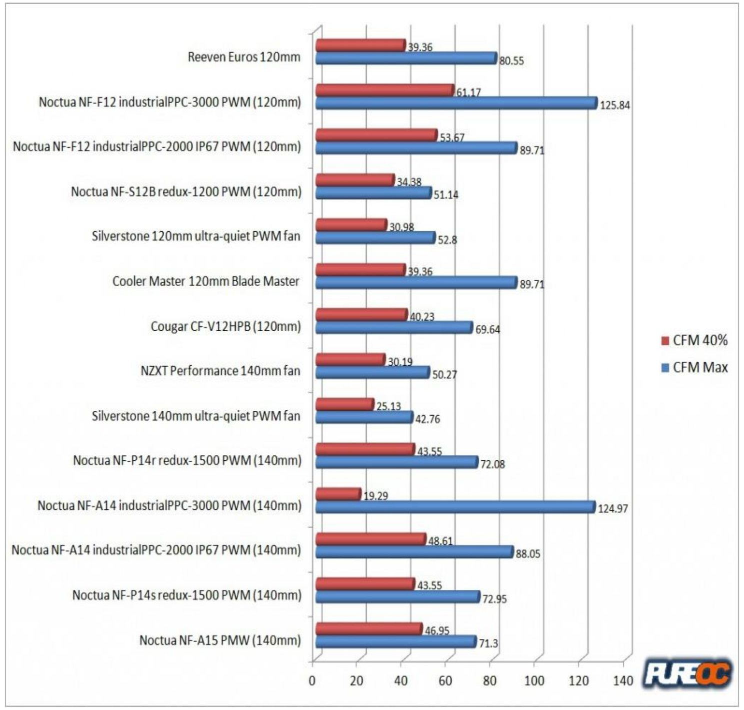
$$R_{total} = \frac{1}{\eta_o h A_t} = \frac{T_h - T_{air}}{q}$$

(where $\eta_0 = 1 - \frac{NA_f}{A_t}(1-\eta_f)$)

$$\eta_f = \frac{tanh mL_c}{mL_c}, mL_c = \left(\frac{hP}{kA_c}\right)^{\frac{1}{2}}$$

Convection coefficient $h = \frac{Nu\, k_{air}}{D}$

(Nu can be determined by air velocity and geometry of heatsink.)

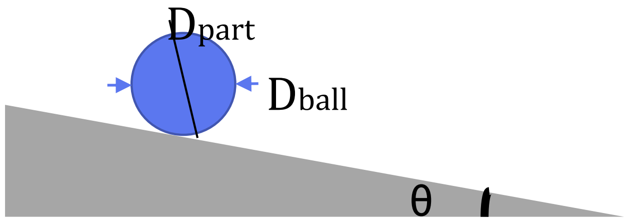


120mm Fan performances (Source : http://www.pureoverclock.com)

Appendix-4 : Vibration feeder

- Simplified version of the theory behind the project An object or particle is placed on a slope. The frictional force(s) prevents the object from rolling/sliding down the slope. When an external force is acted on the slope, it will drop down and back. The object/particle will “fall” onto a higher point on slope. Due to the frictional force(s), the object/particle wouldn’t return to the original point on the slope. Repeating this will cause the object/particle to “climb” up the slope. (Gordon H. Y. Wong)

Appendix-4 : Vibration feeder–How it works



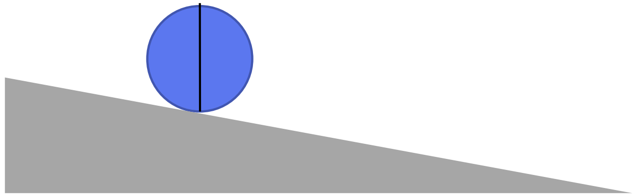
θ=7°, Dball=75mm, Dpart=77mm, g=10m/s^2

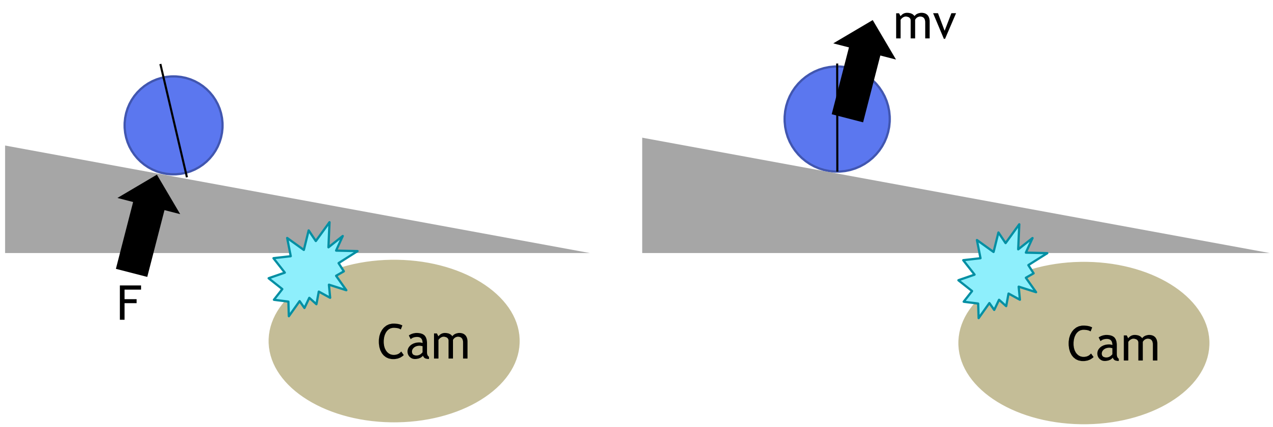
Without movement, We need

θ=$\theta = 1 - \sin\frac{76}{77} = 9.3°$

to overcome the parting line stuck.

That's hard condition





Energy conservation

$$mV_{ball}^2 = mg\Delta h$$

$$V_{ball} = \sqrt{2g\Delta h} = 1.224mm/s^2$$

$$\Delta h = r(1 - \cos\frac{75}{77}) = 0.0612mm$$

$$g = 9.81m/s^2$$

$$V_{con} = V_{ball}l_{sto}/l_{ball} = \omega_{cam}l_{cam}$$

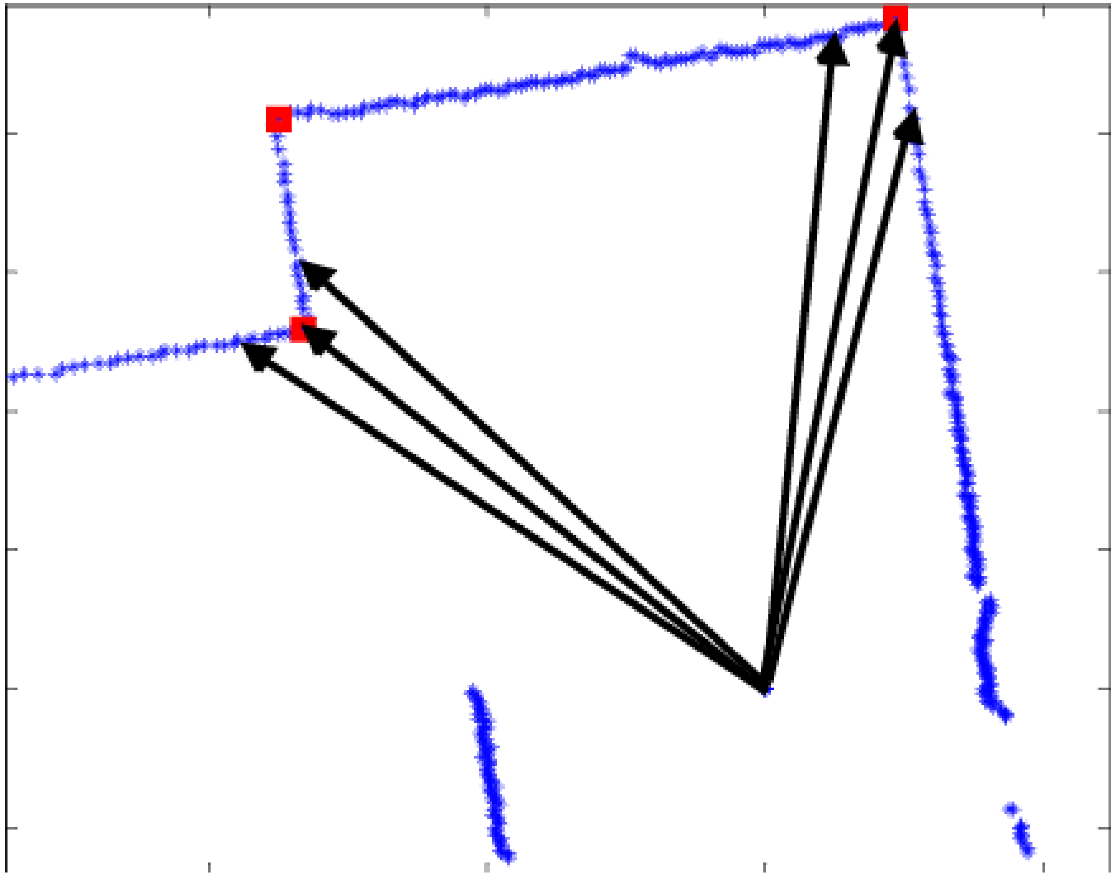
$$\omega_{cam} = \frac{V_{ball}r_{sto}}{l_{cam}l_{ball}} = \frac{150*0.001224}{20*70} = 0.131 \text{ rad/sec}$$

$$N_{cam} = 1.25\ rpm$$

From real experiment, We put practical value

$$N_{cam} = 1.25\ rpm$$

Appendix-5 : Detect Corners with rpLidar



We found the article that suggests good method for detecting corners with lidar.

$$\sum_{j=m-M/2}^{m+M/2} \left| r_m - r_j \right| \geq r_{corner\_threshold}$$

Since we only need to detect concave corners, we modified the algorithm

$$\sum_{j=m-M/2}^{m+M/2} \frac{r_m - r_j}{r_j} \geq r_{corner\_threshold}$$

By modulating the threshold value, we could successfully detect four corners.

Ref) Liu, Shifei & Atia, Mohamed & Gao, Yanbin & Noureldin, Aboelmagd. (2015). Adaptive Covariance Estimation Method for LiDAR-Aided Multi-Sensor Integrated Navigation Systems. Micromachines. 6.