

ME400 창의적 시스템 구현

## Final Report

2018 spring

견지말고기어(박용화 교수님)

20150510 윤상원

## <목차>

### 1. Schedule

### 2. Hardware

2-1)Overview

2-2)Pick up

2-3)Fan and Fin

2-4)Gear

2-5)Frame

2-6)Webcam

2-7)Vibration

### 3. Software

3-1)openCV

3-2)LabVIEW

3-3)ROS

3-4)Integration

### 4. progress

# 1. Schedule

## GANTT CHART

ID	작업명	시작	완료	기간	2018년 04월	2018년 05월
1	Idea Generation	2018-03-19	2018-05-01	6.4w	3-25 4-1 4-8 4-15 4-22 4-29	5-6 5-13 5-20
2	1st Presentation	2018-03-30	2018-03-30	.2w		
3	Motor 구동 및 이동	2018-04-02	2018-04-13	2w		
4	집기 방식 제작	2018-03-19	2018-04-23	5.2w		
5	1차 프로토타입 제작	2018-04-23	2018-05-01	1.4w		
6	2nd Presentation	2018-05-04	2018-05-04	.2w		
7	인식 및 경로 설정 알고리즘	2018-03-30	2018-05-01	4.6w		
8	2차 프로토타입 제작	2018-05-01	2018-05-14	2w		
9	피드백 수집 및 최종 제품 제작	2018-05-14	2018-05-30	2.6w		
10	3rd Presentation	2018-06-01	2018-06-01	.2w		

그림 1 간트차트

프로젝트를 시작했을 때 우선 간트 차트를 작성했다. 3월의 경우 시간이 많이 없어서 각 파트별 소프트웨어를 읽히는데 집중하였고, pick up 모듈 브레인스토밍과 열 및 진동의 설계 등에 대해 논의하였다.

4월 5월부터는 아래와 같이 좀 더 구체적인 계획을 세우며 진행을 했다.

## Plan in April

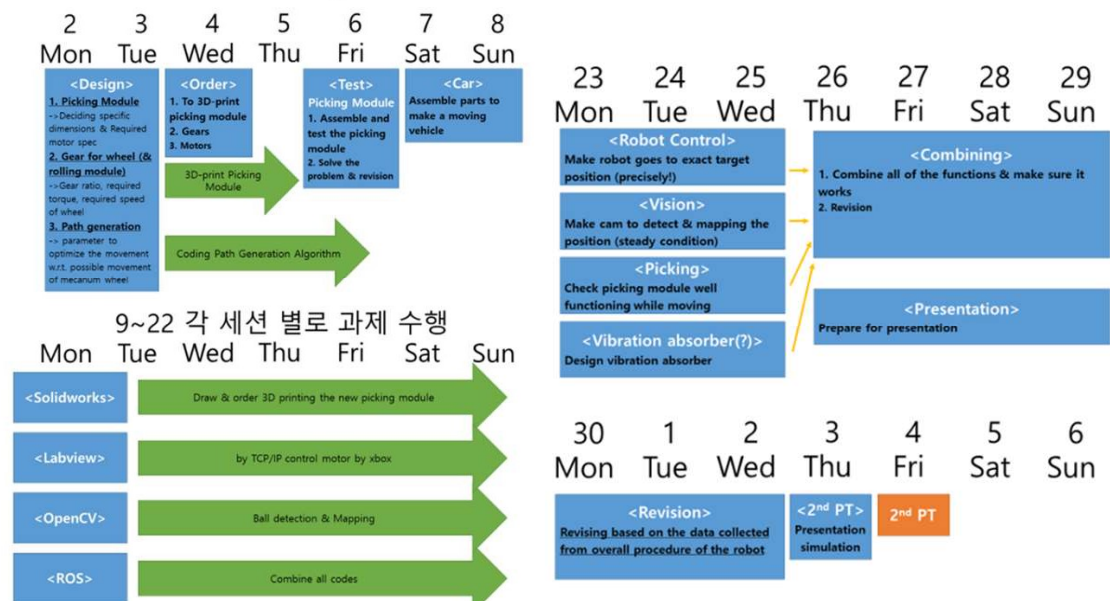


그림 2 4월 계획

## Plan in May



그림 3 5월 계획

## 2. Hardware

### 2-1) Overview

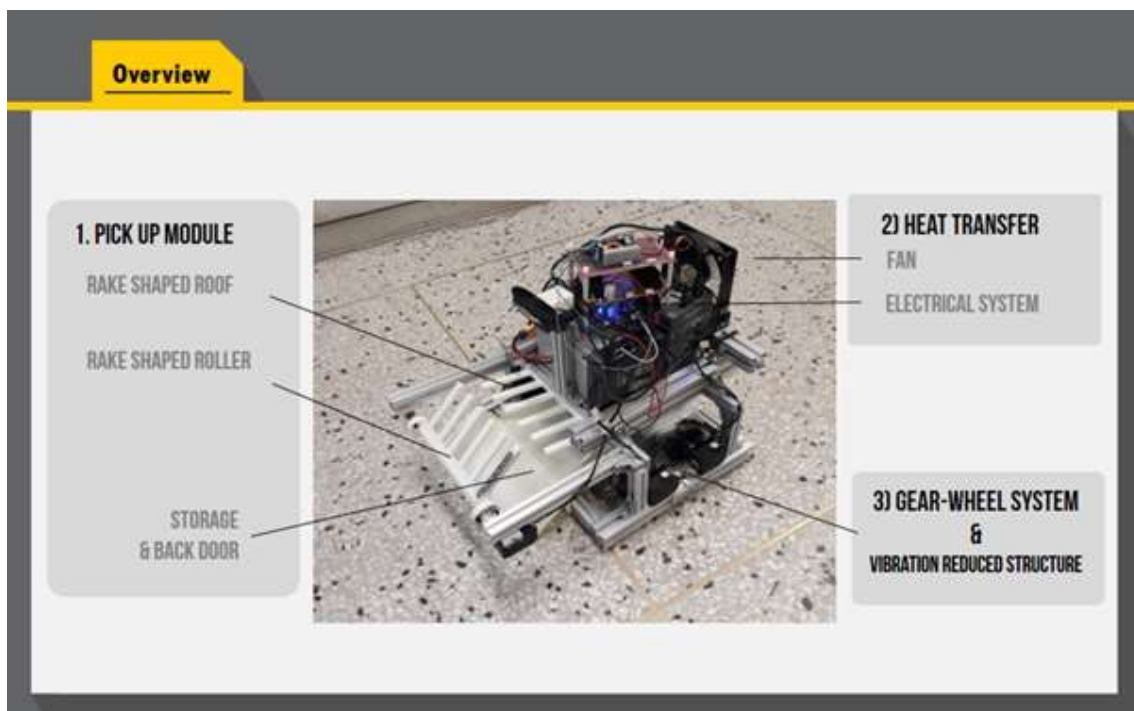


그림 4 전체 모습

우리 제품의 전체 모습은 위와 같다. 핵심 부분인 공을 줍는 방법은 롤러를 이용했다. 공에게 다가간 후 롤러를 돌려 차량 내부 저장고로 공을 저장한 후 목표 지점에서 뒷문을 열어 바구니에 담는 방식이다. 또한 속도의 향상을 위한 기어, 열 방출을 위한 핀과 팬을 장착했다.

## 2-2) Pick up


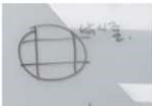
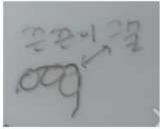

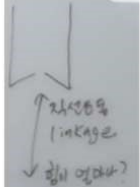

1. 격자형 집게	3. 끈끈이	4. 우선 다 담고 골라내기 (깎때기 형)	
		6. 빨간공 쳐내기	
2. W자형 집게	4. 지게차	8. 벽으로 밀어서 일렬로 만들기	
		9. 바구니까지 이어지는 통로	

그림 5 pick up 브레인스토밍

Pick up 방식을 택하기 위하여 위와 같이 브레인스토밍을 진행했다. Decision matrix를 이용해 우리가 생각하는 최적의 방식인 roller를 선택했고 아래와 같이 제작을 진행했다.

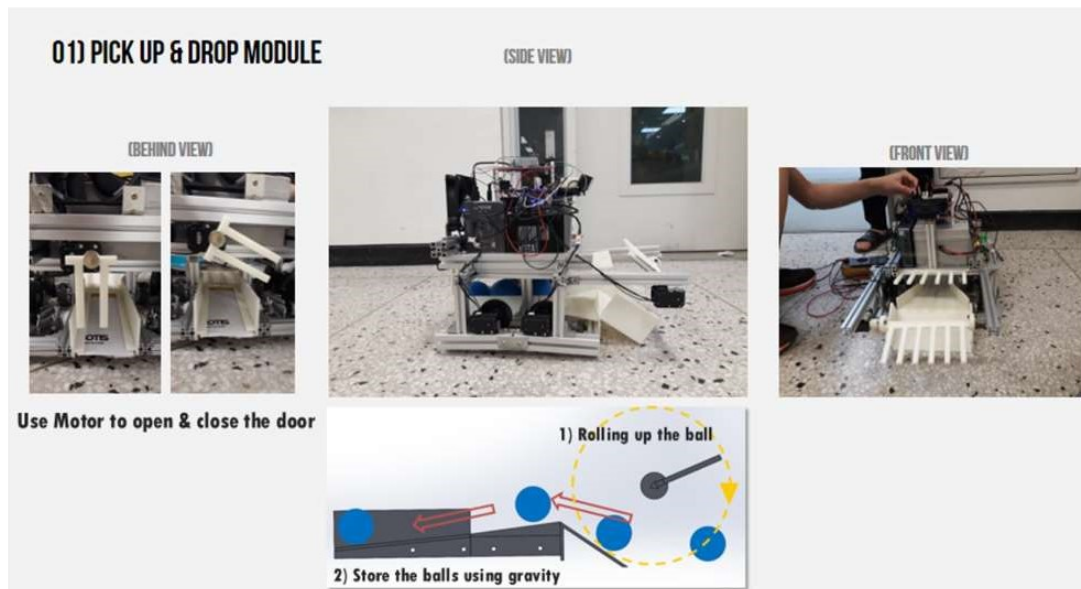
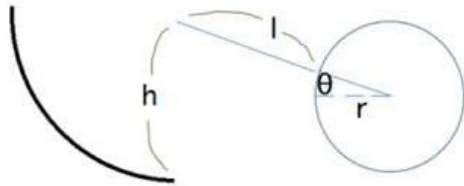


그림 6 pick up 도면

Pick up module은 위 그림과 같이 제작했다. 우선 차체가 공에게 접근한 후 사거리 내에 들어오면 롤러를 앞으로 돌리기 시작한다. 그러면 롤러와 공이 부딪히며 안으로 빨려 들어 오게 되고 차체 중앙 밑에 있는 저장고로 들어간다. 들어간 공은 경사진 저장고 면은 따라 굴러가게 되고 닫힌 뒷문으로 인해 정지한다. 목표물에 도착하면 뒷문이 열리면서 중력에

의해 자연스럽게 안으로 흘러 들어가게 된다.

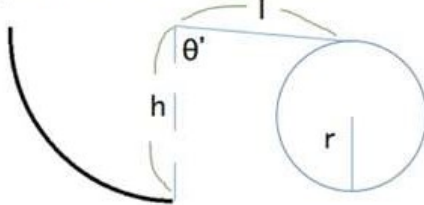
case1) The moment the ball first touches



$$\sin\theta = \frac{h-r}{r+l} \rightarrow \cos\theta = \sqrt{1 - \left(\frac{h-r}{r+l}\right)^2}$$

$$\therefore d = (r+l)\cos\theta = (r+l) \sqrt{1 - \left(\frac{h-r}{r+l}\right)^2}$$

case2) The moment the ball first enters



$$\cos\theta' = \frac{h-2r}{l} \rightarrow \sin\theta' = \sqrt{1 - \left(\frac{h-2r}{l}\right)^2}$$

$$\therefore d' = (l)\sin\theta' = l\sqrt{1 - \left(\frac{h-2r}{l}\right)^2}$$

$$\text{Minimize } d-d' = (r+l) \sqrt{1 - \left(\frac{h-r}{r+l}\right)^2} - (l)\sin\theta' = l\sqrt{1 - \left(\frac{h-2r}{l}\right)^2}$$

$$\therefore h \approx 12\text{cm}, l \approx 9\text{cm}$$

그림 7 optimal dimension of roller

롤러와 공이 부딪히는 상황에 따라 공을 쳐낼 수도 있다. 따라서 그 확률을 최소화 할 수 있는 설계를 위의 상황을 가정해 진행하여 최적의 피벗 높이 및 롤러 길이를 설정했다.

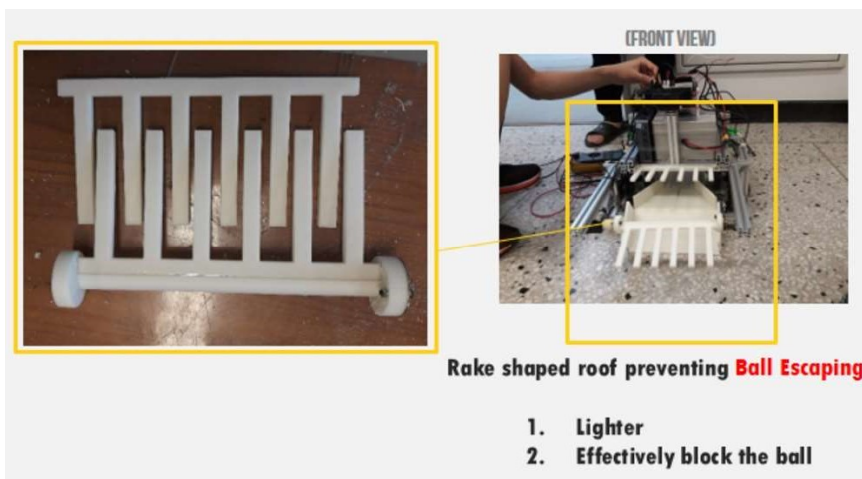


그림 8 rake 모형의 롤러 및 roof

특히 롤러는 차체에 부정적인 영향을 주는 큰 질량을 방지하기 위해 rake 모양을 이용해 큰 효과를 봤다. 또한 공이 저장고로 바로 들어가지 못하고 한 바퀴 회전해서 빠져나갈 상황이 발생할 수 있어 위에 가림막을 설치했다. 재료는 3D 프린터를 이용했다.

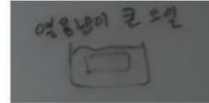


## 2-3) Fan and Fin

1. 드라이아이스  
(혹은 다른 냉매)



3. 열용량이 큰 오일  
에 담그기



2. Pms 위아래로 공간을 두어 이동을 이  
용하여 convection



3. 방열판과 쿨러

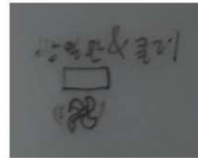


그림 9 heat brainstorming

방출되는 열을 어떻게 줄일 것인가에 대해 다양한 방법을 논의해봤다. 현실적으로 고려해봤을 때 효과도 크고 불확실성도 작은 fin and fan 방식을 사용하기로 했다.

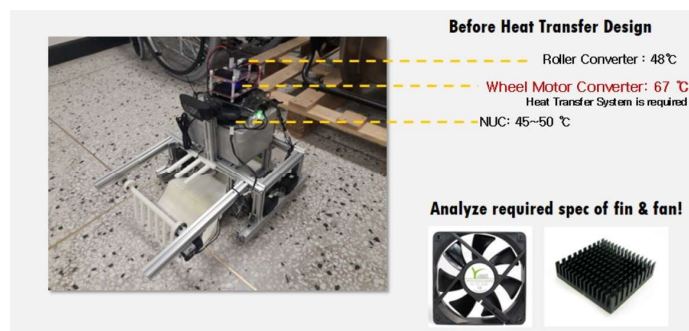


그림 10 온도 측정

왜 열을 줄여야 하는지 알기위해 사전에 각 부품별 온도를 측정해 보았다. 그 결과 motor converter가 67°C까지 올라가며 매우 높은 수치로 나타났다.

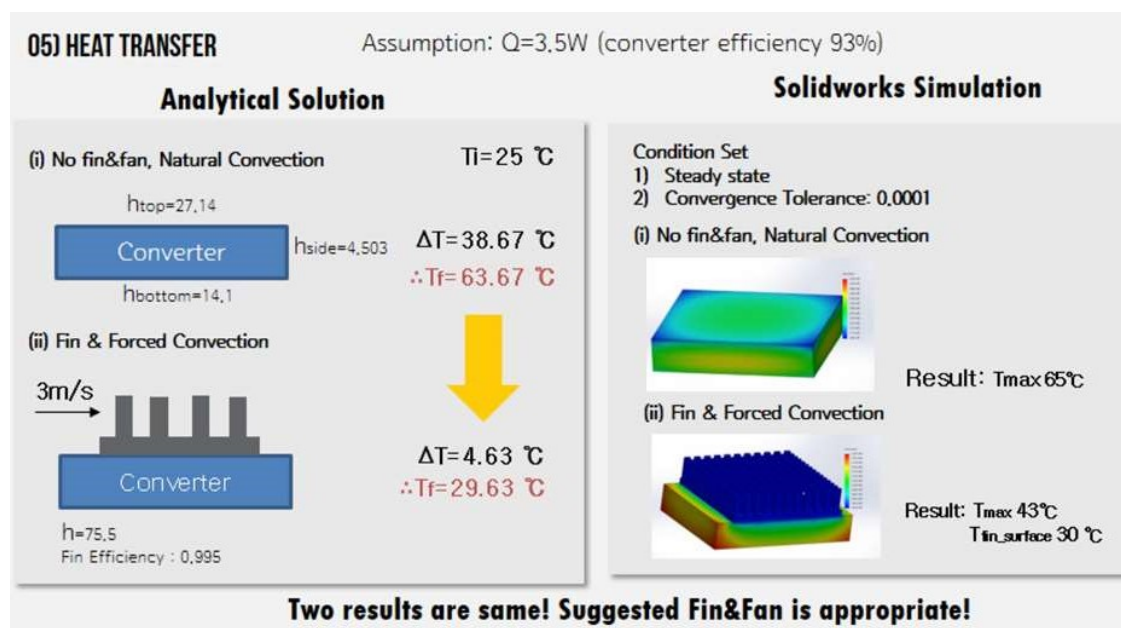


그림 11 온도 분석

가장 열이 많이 나는 모터 컨버터를 집중적으로 생각했다. Conduction or radiation보다 설계가 쉽고 효율적일 것 같은 fin and fan을 이용했다. 해석적으로 또 소프트웨어를 이용해 우리의 fan, fin, converter를 분석해봤고 아래와 같이 20°C이상 감소효과를 볼 수 있었다.

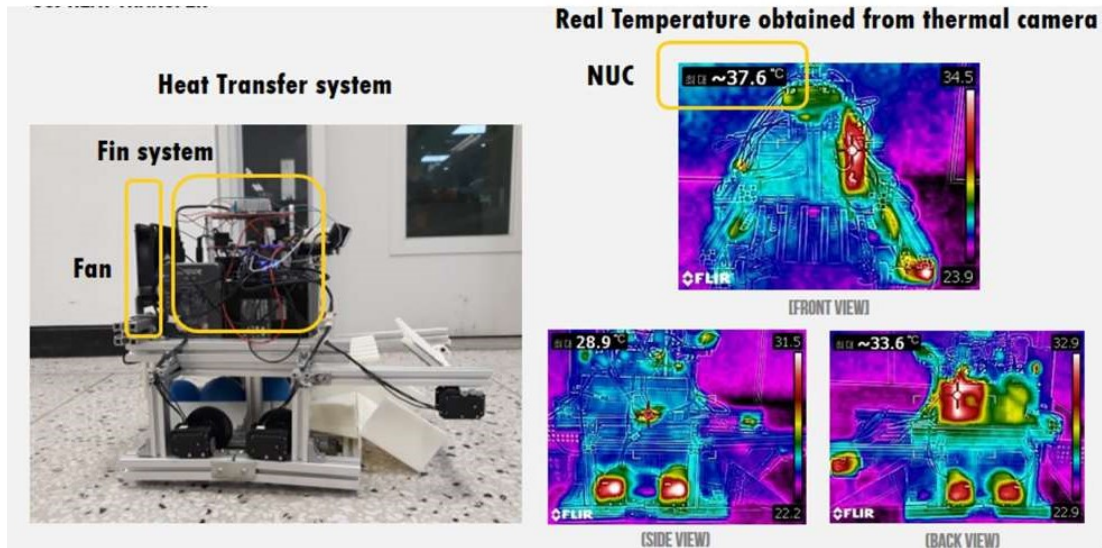


그림 12 제품 열 측정

## 2-4) Gear

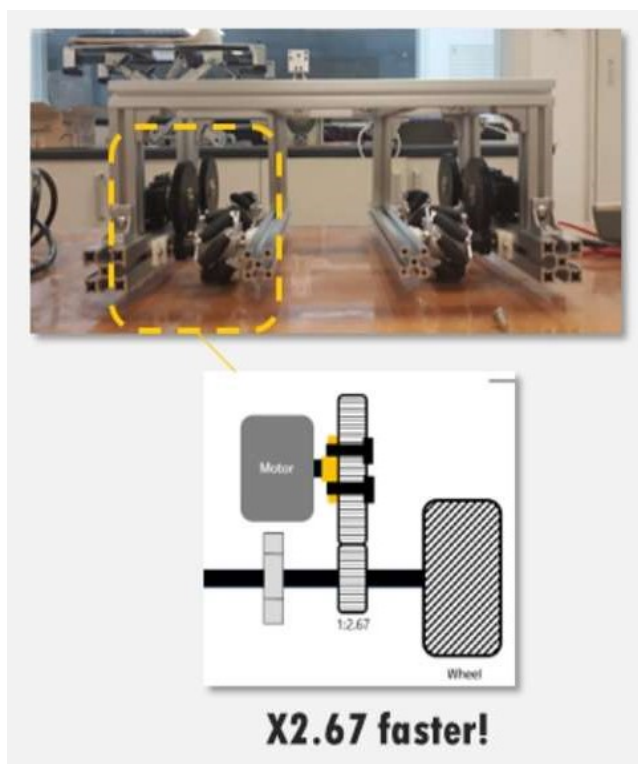


그림 13 Gear 모습

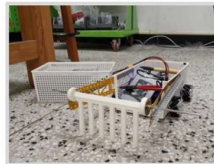
최초 모터인 MX-28AT는 최대 회전 출력이 55rpm이어서 겉보기에도 매우 느렸으며 실제 필드에서 움직일 것을 생각하면 이동시간이 너무 크게 소모되어 속력을 높일 방안으로 기어



를 설계했다. 잇수비는 30과 80으로 기어비는 2.67로써 같은 만큼 속력이 빨라졌다. 기어를 장착함으로써 생긴 뒤틀림, 진동 문제는 바퀴를 두 축으로 연결하며 해결했다.

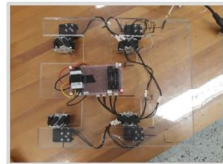
## 2-5) Frame

### MATERIAL



PLASTIC BOX

Large Transformation



ACRYLIC PANEL

Heavy Weight



ALUMINUM

그림 14 Frame material 후보군

플라스틱, 아크린판, 알루미늄 등 다양한 재료를 사용해본 결과 가볍고, 부피 차지도 적고 제작도 쉬운 알루미늄 프레임을 사용하기로 했다.

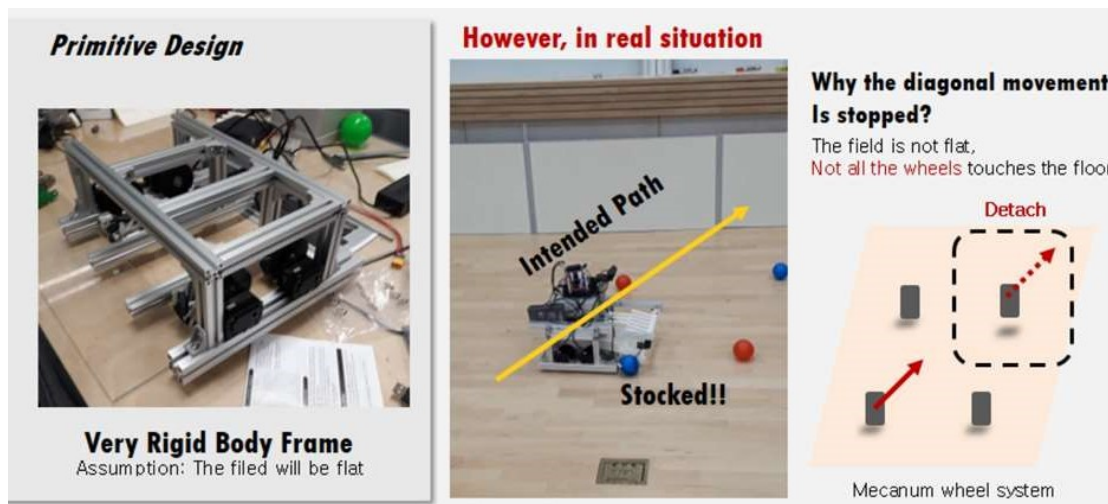


그림 15 Frame 모습 및 대각 방향 문제점

바퀴는 두 축 고정하는 설계를 했고 차체는 전체적으로 rigid했다. 따라서 지면의 형태라는 noise로 인해 모든 바퀴가 땅에 닿지 않는 치명적인 문제가 발생했다. 이로 인해 대각선 방향 움직임에서 바퀴만이 헛도는 상황이 생겼다. 이를 우리는 TRIZ 기법을 이용해 차체에 자유도를 줄 수 있는 방법을 생각하여 차체를 분리하여 회전 자유도를 부여하여 바퀴 축이 약간 움직이면서 대각선 방향의 움직임을 다시 구현했다.

## DYNAMIC INTERPRETATION

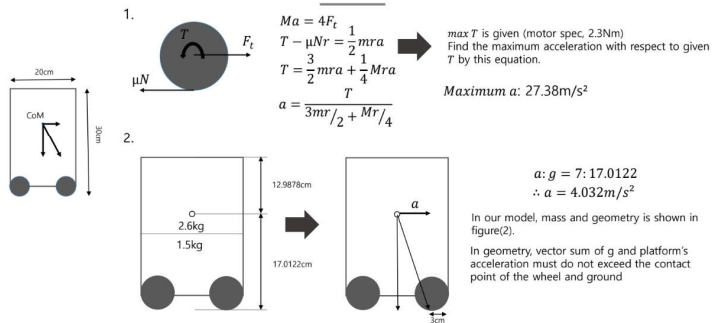


그림 16 제품의 동적 분석

본 차체가 생각보다 높이가 높게 나와서 안정적인지 분석하기 위해 해석을 해보았다. 그 결과 우리가 사용하는 모토 내에서는 절대 불안정하게 될 상황은 발생하지 않는다는 것을 확인했다.

## 2-6) Webcam

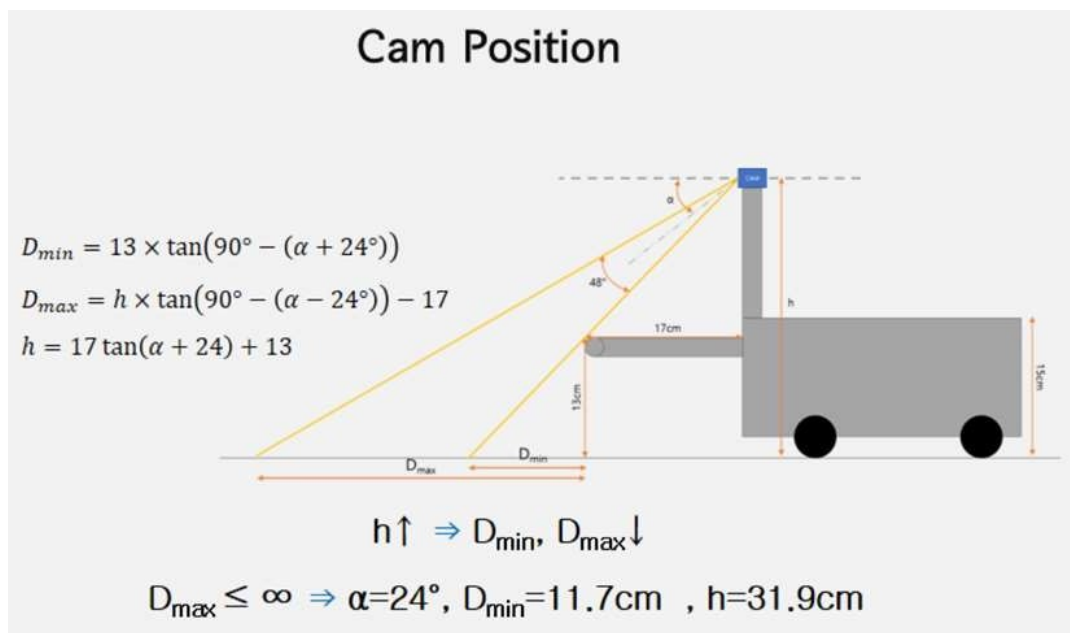


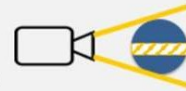
그림 17 최적 카메라 각 분석

카메라를 다는 각도가 매우 중요하다. 최대한 가까운 곳까지 보면서 멀리 있는 공 또한 인식할 수 있도록 웹캠의 위치를 계산하였다. 대각선 시야각이 76°라는 사실을 이용해 가로(66°), 세로(48°)의 시야각을 추정해 계산했다.

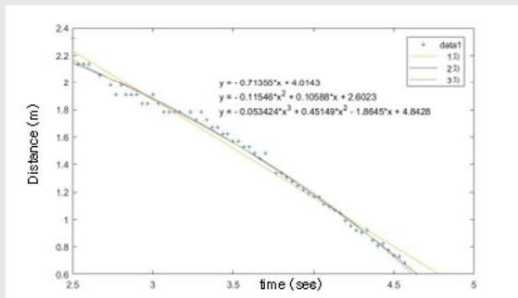
## 2-7) Vibration

### 04) VIBRATION ANALYSIS

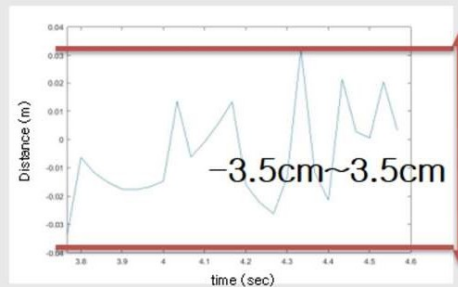
#### Ball Depth Detection



#### Raw Distance Data (AC+DC)



#### Vibration (AC) Distance Data

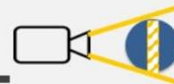


Eliminate 2nd order polynomial & Pixel errors

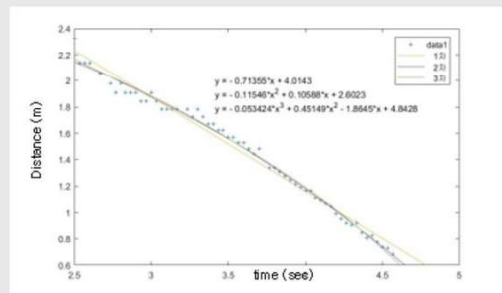
그림 18 직선 방향 raw, ac 데이터

### 04) VIBRATION ANALYSIS

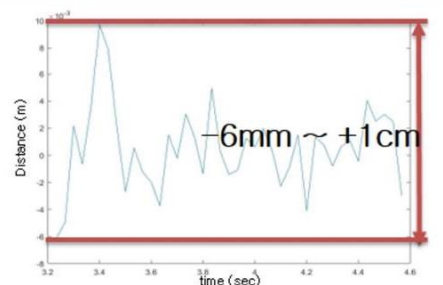
#### Horizontal position Detection



#### Raw Distance Data (AC+DC)



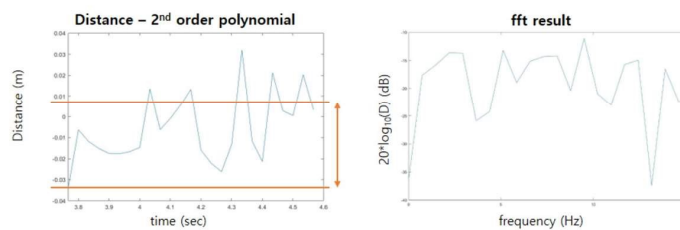
#### Vibration (AC) Distance Data



Eliminate 2nd order polynomial & Pixel errors

그림 19 수평 방향 raw, ac 데이터

## Vibration analysis- fft result (direct motion)



Distance error is affordable!  
No meaningful vibration included!

그림 20 FFT 그래프

차체를 앞 또는 좌우로 움직이며 지정된 공의 위치를 찍고 2차 보간다항식을 빼서 나타내 보았다. FFT도 그려보았지만 의미있는 결과를 얻지 못해 진동수보다는 진폭에 초점을 맞추고 분석했다.

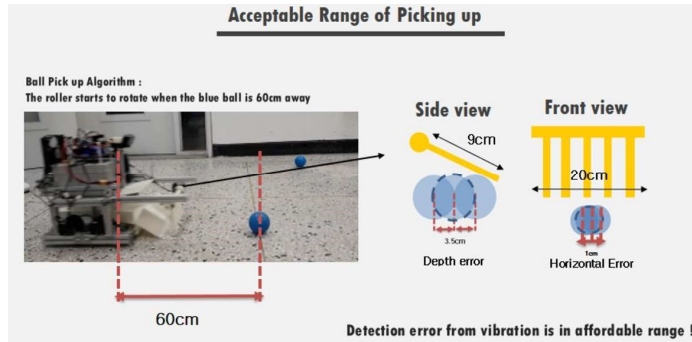


그림 21 진동 문제점 분석

우리 카메라의 진동 진폭이 시스템에 미치는 영향이 없기 때문에 추가적인 진동제어 장치를 추가하지는 않았다.

### 3. software

#### 3-1) openCV

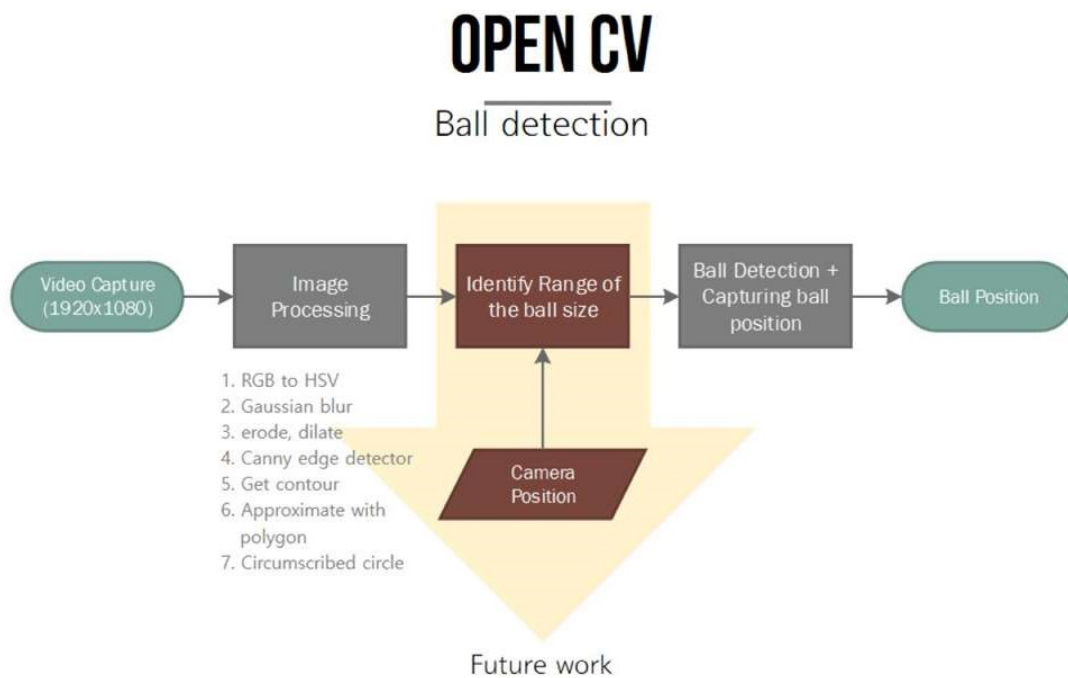


그림 22 openCV 개략도

openCV는 카메라를 이용해 파란 공, 빨간 공, 초록 공을 인식하고 그 위치 데이터를 저장한다.

## 3-2) LabVIEW

### Overview of main server code

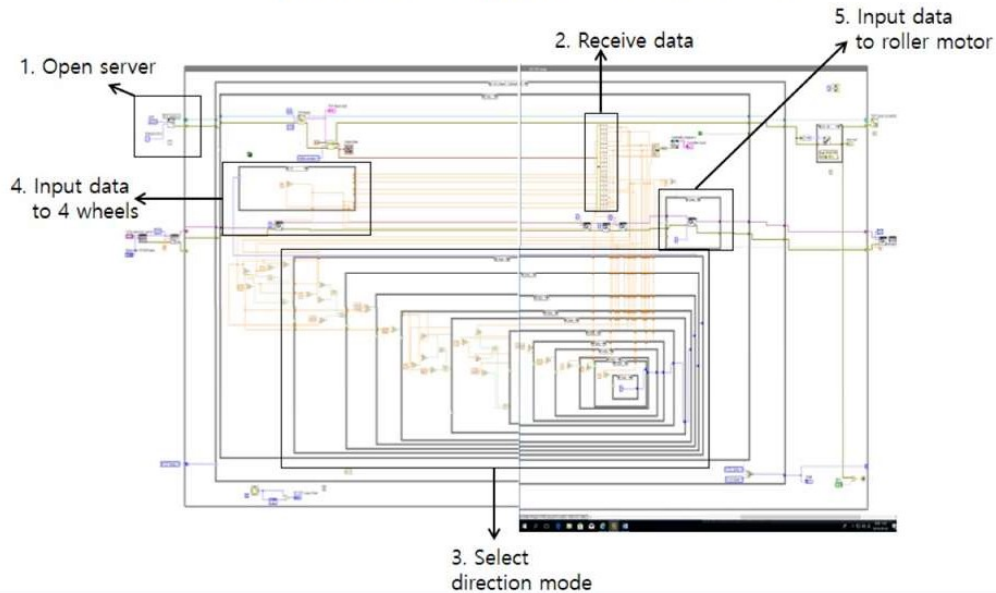


그림 23 LabVIEW 메인 코드

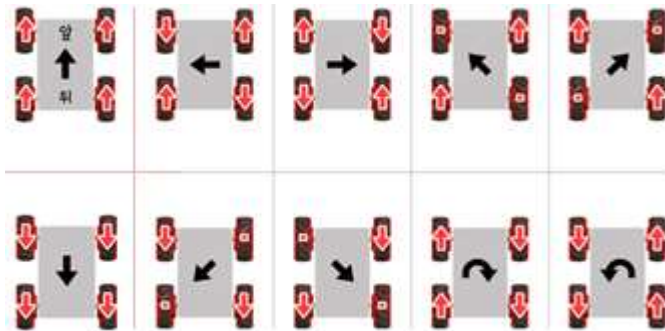


그림 24 10가지 바퀴 모드

LabVIEW의 경우 ROS와의 TCP/IP 통신을 통해 ROS에서 쏜 명령을 받아들이며 위와 같은 10가지 모드에 대한 바퀴 제어를 한다. 또한 롤러 모터와 뒷문 모터도 각각 제어한다.

### 3-3) ROS

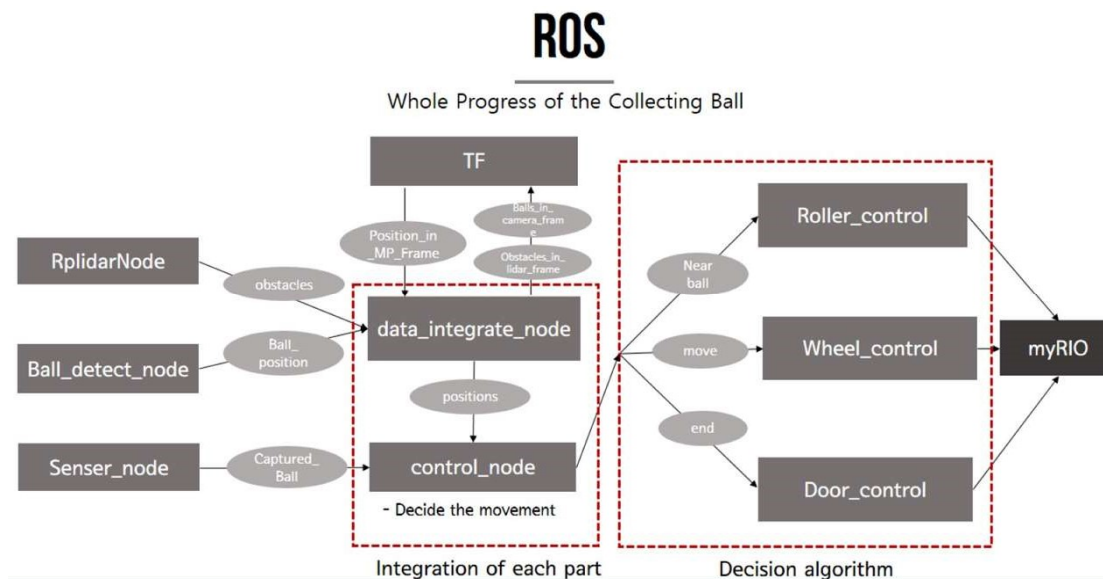


그림 25 ROS nodes

전체 소프트웨어를 총괄하는 ROD는 LabVIEW 및 openCV의 데이터를 이용하여 로봇을 제어한다. 공을 찾는 노드, 바퀴제어 노드 등이 있다.

### 3-4) Itegraion

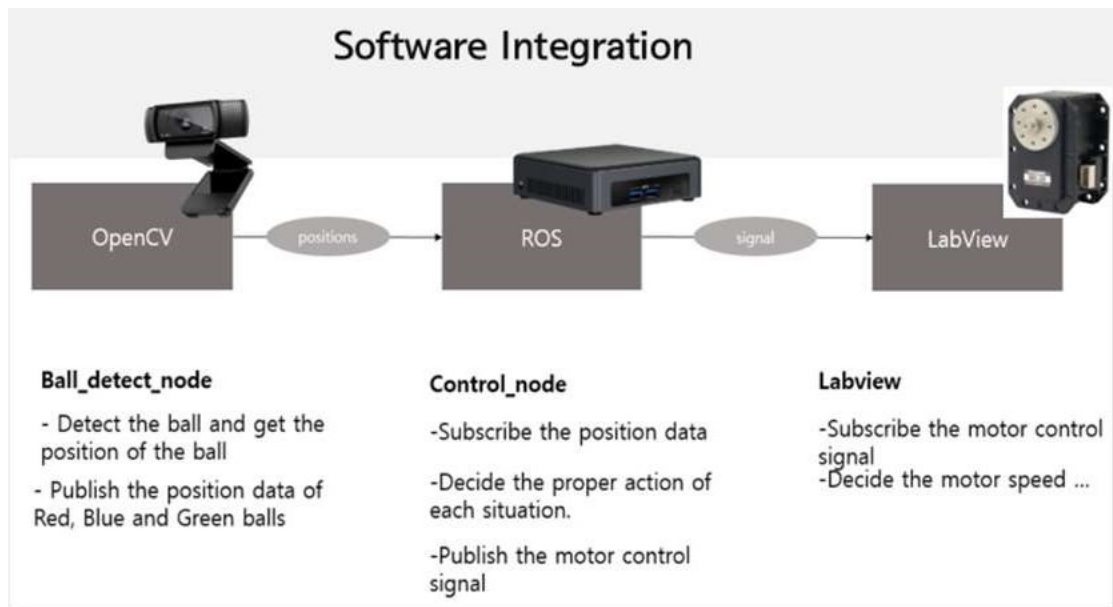


그림 26 Software itegration

software는 크게 카메라의 정보를 처리하는 openCV, 모터를 제어하는 LabVIEW, 소프트웨어들을 통합하고 제어하는 ROS가 있다. openCV는 파란 볼, 초록 볼, 빨간 볼을 인식하고 위치정보를 ROS로 준다. 이러한 공의 위치정보를 받는 ROS 상황마다 로봇을 제어하고, 바퀴



및 롤러 모터를 제어하기 위한 데이터를 LabVIEW로 TCP/IP 통신으로 보내게 되고 LabVIEW는 받은 데이터를 이용해 모터를 구동한다.

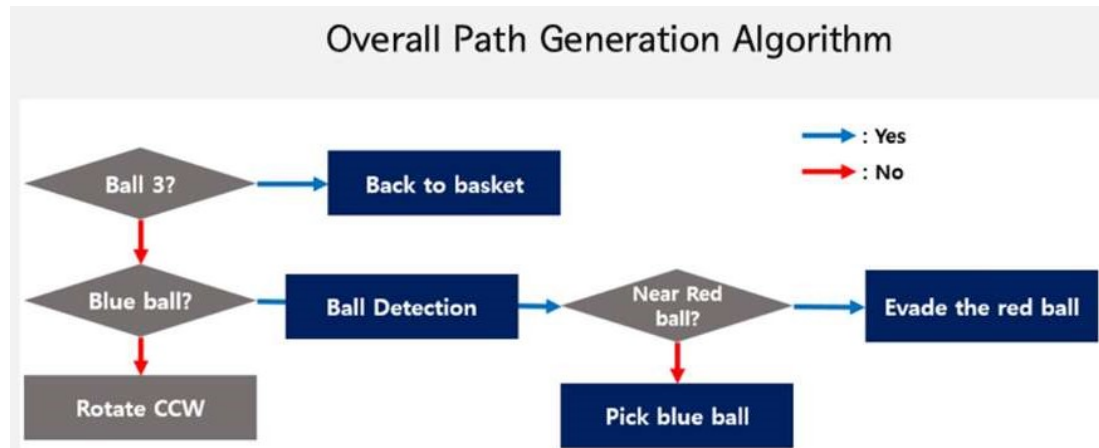


그림 27 Path algorithm

최종적으로 ROS에서의 알고리즘을 도식화 하면 위와 같다. 카메라에 파란 공이 잡히지 않으면 반시계 방향으로 돈다. 그러다 파란 공이 보이면 파란공으로 가고 주변에 빨간공이 있다면 피한다. 이러한 과정으로 파란공 3개를 잡는 행위를 하면 바스켓으로 돌아가 공을 놓는 행위를 한다.

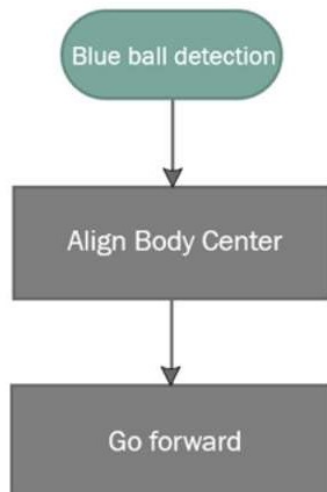


그림 28 Ball detection node

ball detection node는 간단하게 파란 공을 발견하면 파란 공으로 직진할 수 있도록 정렬한 후 직진운동한다.

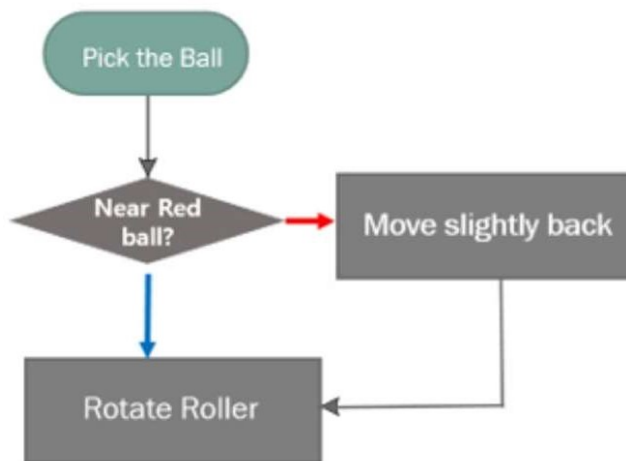


그림 29 Pick up node

pick up의 경우 차체가 공과 일정거리가 되면 롤러를 돌리기 시작하며 파란 공으로 직진해서 공을 잡게 된다.

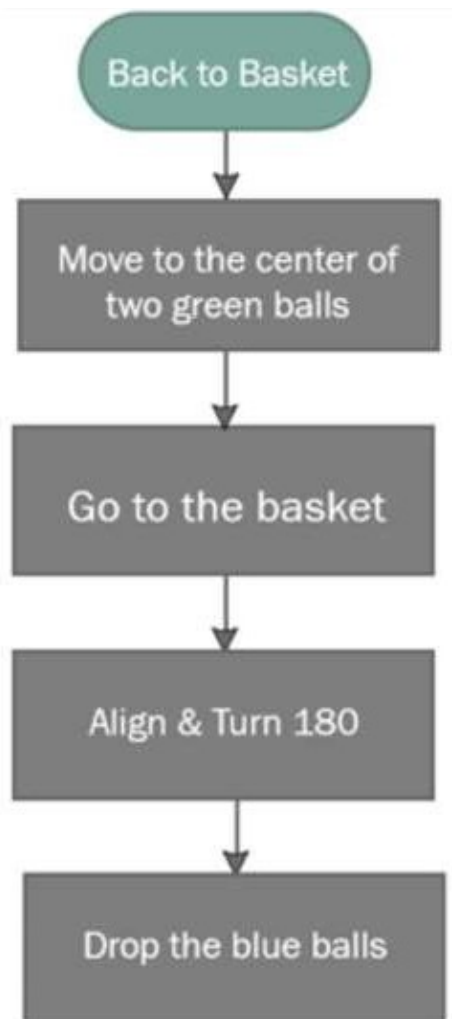


그림 30 Back to basket node

가장 힘든 부분이었던 목표지점으로 돌아가는 node이다. 파란 공을 다 주우면 초록 공을

찾는다. 두 초록 공 중 더 먼 거리에 있는 공 방향으로 계속해서 이동하며 차체를 바구니의 중앙에 오도록 하면서 바구니로 전진한다. 어느 정도 거리 안으로 들어가면 차체가 180° 회전하고 뒤로 이동한다. 일정 거리를 이동한 후 뒷문을 연다.

## 4. progress

간략한 우리 조 및 LabVIEW의 진행 사항을 정리하면 아래와 같다.

### 1주차)

pick up module에 대한 방대한 생각을 했고, 원통형 모양에 고무줄을 박은 1차 프로토타입을 제작함. LabVIEW는 LED를 조금 켜다 껐다 해보는 등의 소프트웨어에 적응하는데 노력을 함.

### 2주차)

기존의 아이디어에서 롤러 모드로 변경함. 현실성, 속력, 오작동, 난이도 등을 고려해 기존의 아이디어가 힘들 것 같다는 생각을 함. 2차 발표 준비를 위해 진동, 열 등 다방면에서의 제품에 대한 문제점 등을 찾고 발전해 나감. LabVIEW의 경우 다이내믹셀을 직접 돌려보는 코드를 진행함.

### 3주차)

1차 발표 피드백을 통해 창의적인 부분을 가미하려 노력함. 1차 프로토타입을 만들어 보려 했음. 컨버터, 케이블타이, 스위치 등 제품을 선정함. 적정한 롤러 모터 스펙을 정함. LabVIEW의 경우 TCP/IP 통신을 이용해 10가지 모드 바퀴 제어를 시도함.

### 4주차)

속력 향상을 위한 기어 설계에 온 힘을 쏟음. 기존에 있는 기어를 사기는 쉽지 않아서 주문 제작하기로 하였음. 대화동에 갔지만 맞는 스펙의 컨버터가 없어 온라인에서 찾아서 주문함. LabVIEW의 경우 바퀴 제어가 되긴 되지만 명령을 한 칸씩 밀려서 작동하는 문제, 속도가 느린 문제, 가끔 명령을 듣지 않는 문제 등의 코드 오류를 수정하는데 노력함.

### 5주차)

주문했던 컨버터 및 모터를 배송 받았음. 납땜 등의 추가 작업을 진행함. LabVIEW의 경우 ROS와의 TCP/IP 통신을 하기 위해 시도함. 하지만 byte 수를 어떻게 읽어야 하는지 몰라서 막힘.

### 6주차)

1차 알루미늄을 이용한 프레임 제작을 진행함. 기어를 달고 싶었지만 기어를 다는 문제가 생각보다 쉽지가 않음. ROS와 TCP/IP 통신 코드를 완료함. 대충 로봇을 움직이는데는 성공함. 하지만 납땜에 문제가 있어 정확하게 돌리지는 못함.

7주차)

어느 정도 차체가 완성되었음. 계속 기어가 물리는 문제가 발생함. 또한 선과 납땜 등의 문제로 계속 진행하기가 어렵기도 했음. 하지만 파란 공 3개, 빨간 공 3개를 놓고 자율 주행 시켜 줍는데는 성공함. 하지만 창의적인 부분은 아직 미흡

8주차)

본격적으로 열 및 진동 분석을 시작함. 열의 경우 fin, fan을 이용하기로 최종결정함. 진동은 우리 시스템에 크게 문제가 없어 왜 문제가 없는 가를 구체적으로 분석하기로 하고 추가 부품을 장착하지 않기로 함. 알고리즘 개선을 통한 시간 단축도 논의함. Lidar를 어떻게 이용할지 의논함. 공 3개를 줍고 돌아가야 하는데 그 판단을 어떤 식으로 할지 논의함. 진동은 공을 이용해 좌표를 찍고 FFT를 그려 분석해 보았음. 차체의 롤러 수치, 경사 등이 잘 맞지 않아 공이 끼이는 현상이 꾸준히 발생해 다시 제작함.

9주차)

롤러 최적 설계, 뒷문, 경사로 최적 설계를 진행함. 뒷문 모터를 배송 받아 달았음. 최종적으로 바구니에 공을 담는 알고리즘을 빨리 개발하려고 노력함. 생각보다 바구니와 라인을 맞추는 게 쉽지가 않았음.

10주차)

공동강의실에서 데모를 진행하였는데, 바퀴가 축이 바닥에 닿지 않는 문제가 발생함. 프레임에 자유도를 주어서 해결하려함. 진동의 가진원이 뭔지 분석함. 주문한 방열판 및 팬이 도착해 바로 부착함. 그리고 열분석을 바로 진행함. 원래 알고리즘에서 왼쪽부터 오른쪽으로 공을 줍고 돌아오는 코드로 변경했지만 오히려 중앙에서 빨간 공을 피하는 시간이 길어져 다시 최단 거리에 있는 공부터 수거하는 방식으로 변경함. 주차문제의 원인이 카메라가 초록 공을 두 개가 아닌 네 개로 인식하고 있다는 점을 발견하고 openCV코드를 바로 수정함

11주차)

공동강의실에서 계속해서 데모를 진행함. 롤러가 공을 정확하게 줍지 못하는 현상이 자주 발생해 좀 더 정확하게 주울 수 있도록 시간, 속도, 각도 등을 제어하려고 노력함. 또한 바구니로 돌아오는 시간도 오래 걸리고 정확성도 조금 떨어지는 점을 계속해서 보완하려함.