Intelligent Picking - Round 3 Template

Team Name: X Ash A-12

Institute Name: Indian Institute of Technology (BHU) Varanasi

Team Member Details

Name	Batch	Achievements
Sirusala Niranth Sai	2022	● 1st position in Pixelate, an image processing and robotics based event in Technex’19 ● Represented the institute in the Inter IIT Techmeet 8.0, in 2 events namely 1. DIC’s Terrace farming robot for hilly areas 2. Engineers’ Conclave(Showcased a Bipedal robot) ● 1st position in Funckit’19, a digital circuit fabrication event in Udyam’19, IIT (BHU) Varanasi ● 3rd position in Robotron’18, IIT (BHU) Varanasi ● 2nd runner up in Mosaic, a Computer Vision and Image Processing based event in Udyam’20
R Lokesh Krishna	2022	Represented the institute in the Inter IIT Techmeet 8.0, in 2 events namely ● Engineer’s Conclave - a project presentation event ● DRDO’s SASE’s Drone fleet challenge Winner of various events in relation to robotics, computer vision and Digital Electronics at our institute level, namely ● Pixelate under Technex 2019 ● Mosaic under Udyam 2019 ● Funkit under Udyam 2019

Team Member Details(Contd.)

Name	Batch	Achievements
Raghav Soni	2023	● Represented the institute in the Inter IIT Techmeet 8.0, in DRDO’s SASE’s Drone fleet challenge. ● Won IIT(BHU)’s 2020 Machine learning and Digital Electronics Events under Udyam Fest ● Participated in Pixelate(Computer Vision Event) under Technex 2020
Ayush Kumar Shaw	2023	● Winner of Mosaic 2020,Machine Learning event held in the Annual fest of the Electronics Engineering society of IIT(BHU) Varanasi. ● Participated in Pixelate(Computer Vision Event) under Technex 2020

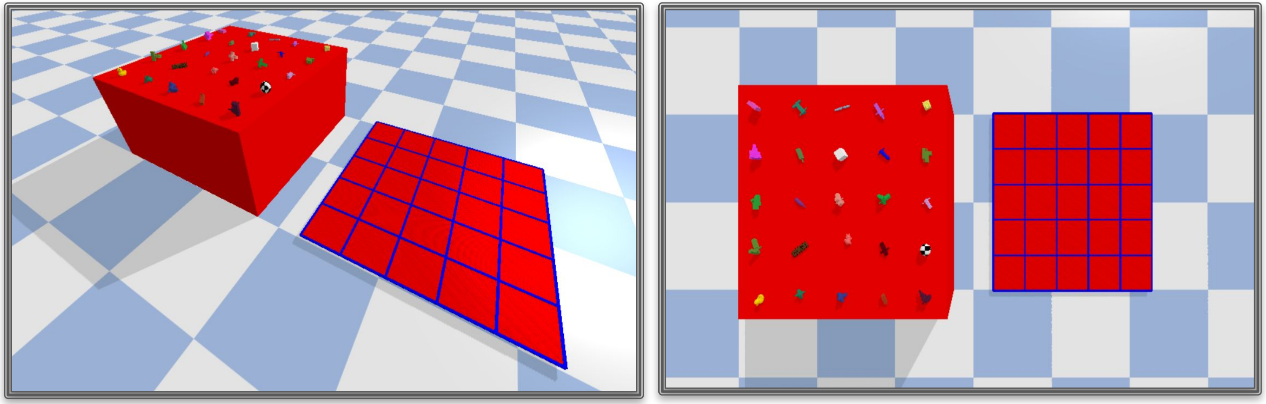
Index:

Introduction

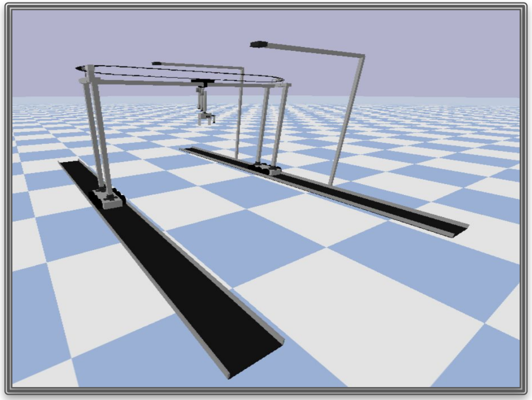
The problem of **pick and place** has been one of the actively studied area and a canonical problem in robotics. The **Amazon Robotics Challenge (ARC)** has a rich tradition for the fabrication of highly robust and competitive warehouse robots that do classify and segregate objects apart from just pick and place. The advent of **Deep Reinforcement Learning** as a reliable alternate for learning robot controllers has greatly increased the dexterity and robustness of these arms.

The given problem statement of **Flipkart Grid 2.0**, is quite unique, unparalleled, challenging, and demands a great amount of customization and design improvements in terms of both hardware and software. The **enormous dimension** of the arena and the relatively **heavier payload** entirely eliminates the possibility of using any pre-existing methodologies. Also fabricating a robot from scratch at the given price point makes the challenge event the more exciting. Thus, we are **sharing a solution** for the above task, with all our experiments and results which according to the best of our knowledge **the most cost-efficient, simplistic yet robust approach**.

Our Hardware Approach



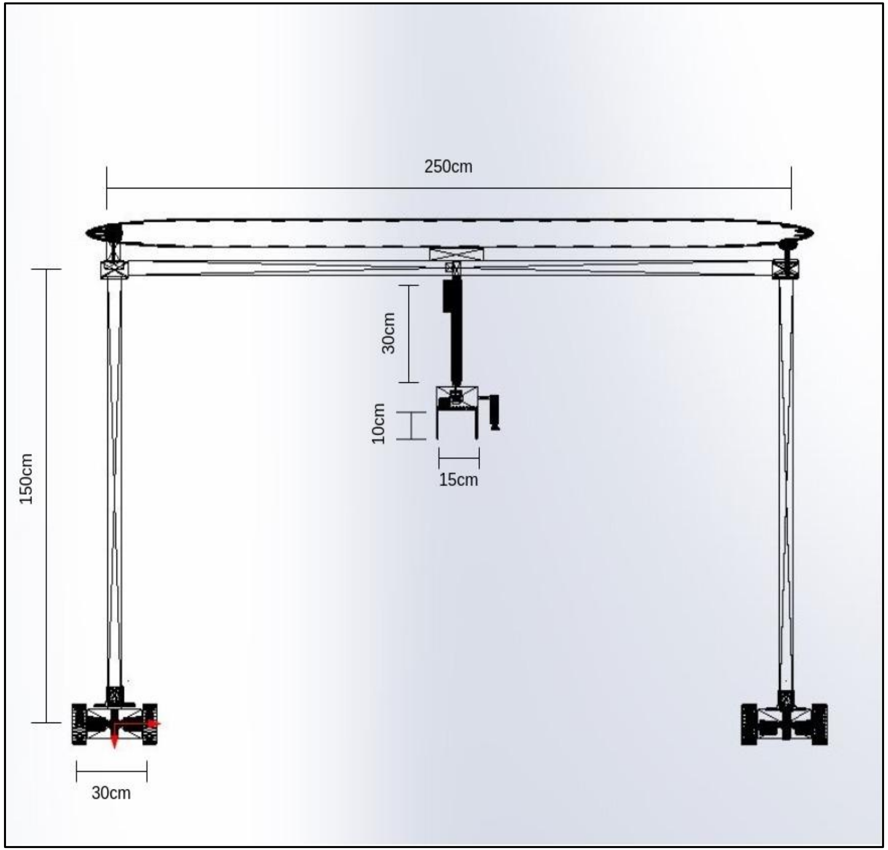
The Arena(Pick Area and Drop Area)



Our Robot

Our robot was designed considering the **enormous dimensions of the arena** and the **relatively greater magnitude of payloads**.Unlike the traditional single 6 Dof, the arm which is greatly suited for workspaces that are relatively smaller and densely packed, the given problem statement poses a new set of challenges. Workspace of **4m X 2m x 0.9m (max)**, makes it almost **impossible to build a single robot arm** that could trace the length and breadth of the arena and carry the payload with realistic torques and forces within the given budget. Having said that we incline to a more **static structure** that is able to transfer the goods with **greater reliability** and has **inherent static stability** while in motion. **The structure and design parameters** of the robot are discussed in detail in upcoming parts.

Hardware Design



Our robot is greatly inspired by cartman, owing to its **cost-efficient cartesian design** which could cover the entire work area in a stable fashion.

This arrangement keeps our **design free of the singularities** that generally arise in complicated robot arms with interdependent joint motions.

The **generic 6 DOF robot arm, requires high torque motors** at every joint to support the payload at the end effector whose costs are around INR 10,000 per unit. However the **torque to be applied per joint** is **drastically decreased** due to **our design** and thus we are unaffected by the above limitation.

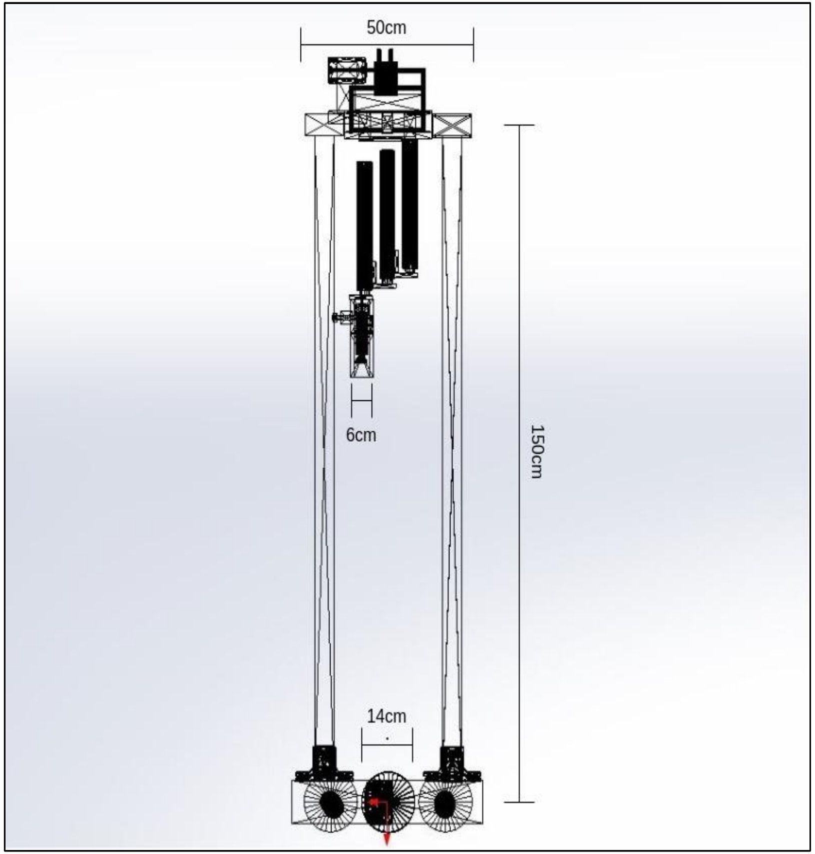
Hardware Design (ContD)

This design further **decomposes the 3D translational motion into two simple steps** namely

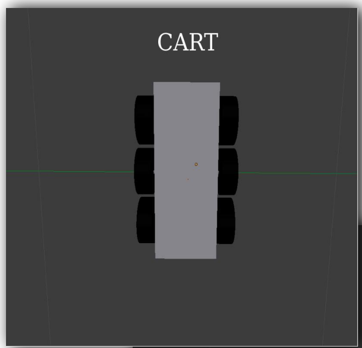
1. **XY plane motions** - realized by the Frame cart and Head motions (explained later).
2. **Z-direction motion** - realized by the 3 stage Linear/Telescopic actuator.

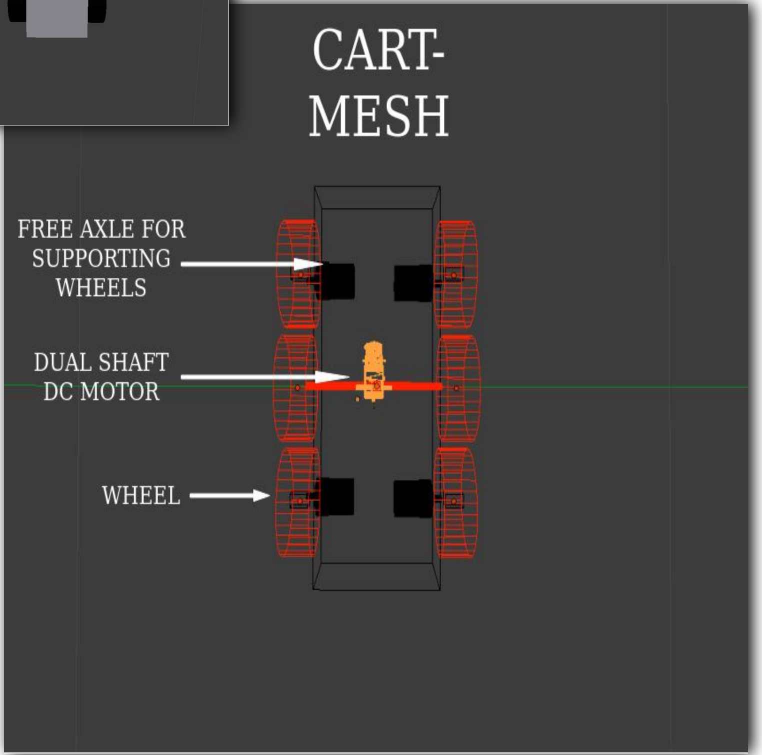
The required flexibility of motion for the **optimal approach of the gripper** is given by the motors local to the grasping unit and possible improvements in this are discussed in later sections.

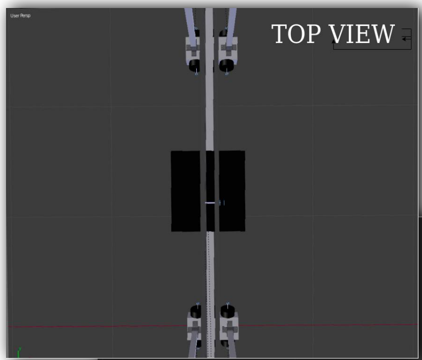
Such a **simplistic and reliable hardware**, allows us to focus and **experiment** with various **planning and control strategies**.

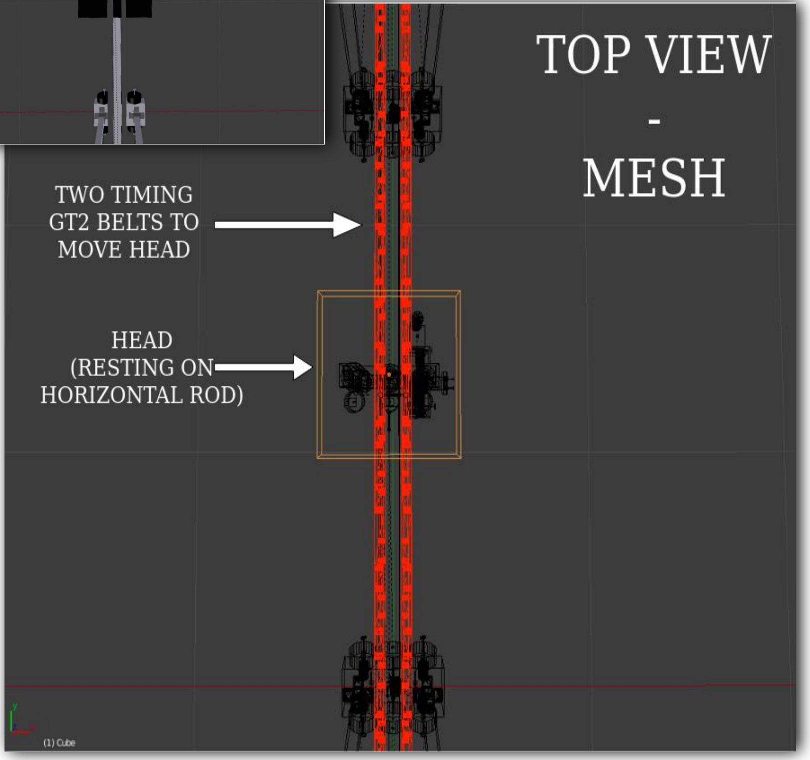


MESH ANALYSIS AND MOTOR PLACEMENTS:

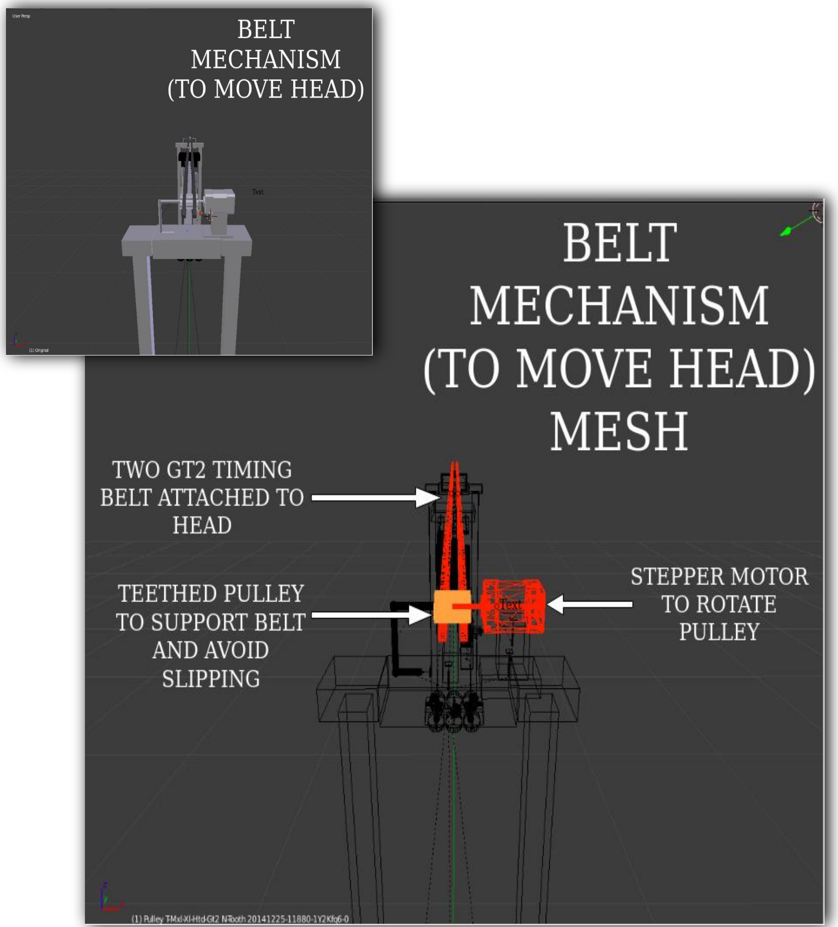


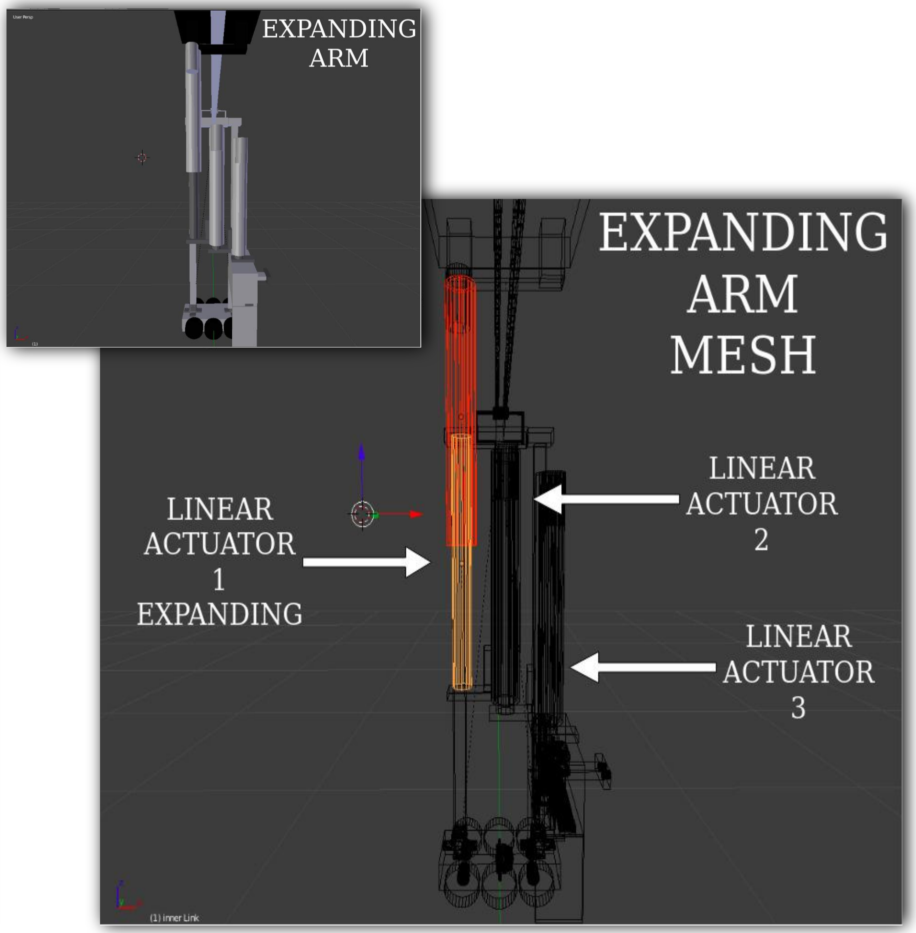






MESH ANALYSIS AND MOTOR PLACEMENTS (ContD):



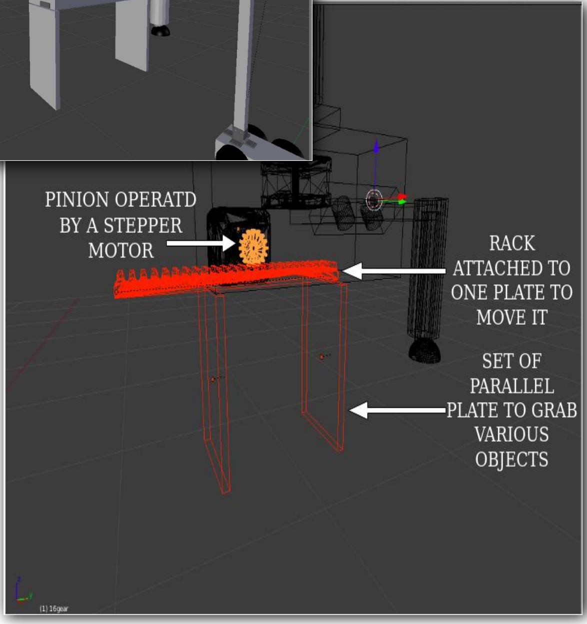


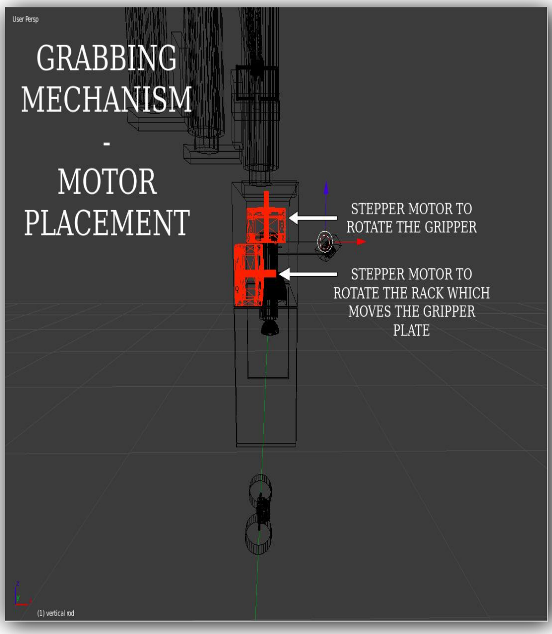
MESH ANALYSIS AND MOTOR PLACEMENTS (ContD):

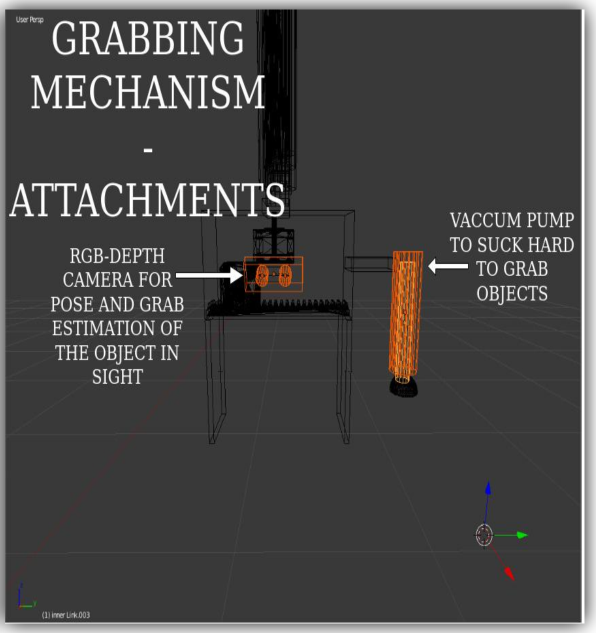
GRABBING MECHANISM-

- Our grabbing mechanism is based on a **rack and pinion** arrangement.

- The pinion is controlled using a **stepper motor** while engaging the rack, which operates to translate **rotational motion** into **linear motion.**

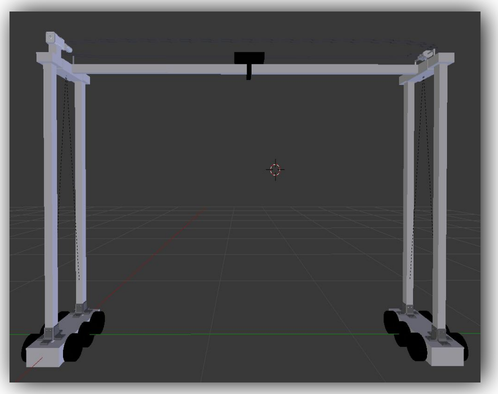






Robot Assembly

Our robot has a **modularized design** and is composed of **three different parts** namely the frame, Arm, and the Grasp unit.



The Frame:

This is perhaps can be called the **pseudo-base** of our robot. The frame with the cart is responsible for motion along the length of the arena. Thus it effectively **slices out a 2D planar region** of interest(YZ plane) from the robot's 3D workspace.

The Arm:

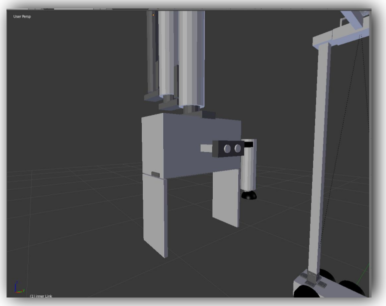
The **elevation of the pick area with respect to the drop area** poses a new challenge and need for an additional degree of freedom to pick and place the payloads accurately. This task is carried over by the Arm unit, which is made of a **3 stage telescopic actuator** with a **compressed length of ~0.3 m** and an **elongated length of ~1.2 m.** This places the object being carried safely inside the grids drawn in the drop area

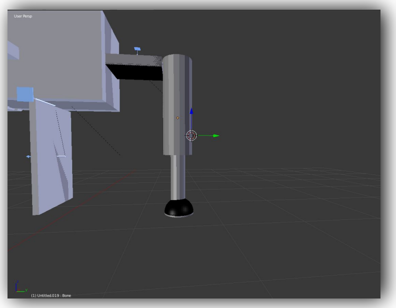
Robot Assembly (contD)

The Gripper:

The pick area is expected to have a **diverse list of household objects** with **varied physical properties** like nature, geometry, texture, structural integrity, etc. However, it is arguably true that it could be drastically simplified into **two different classes**, namely

1. Objects that could be **sucked.**
2. Objects that could be **grabbed.**

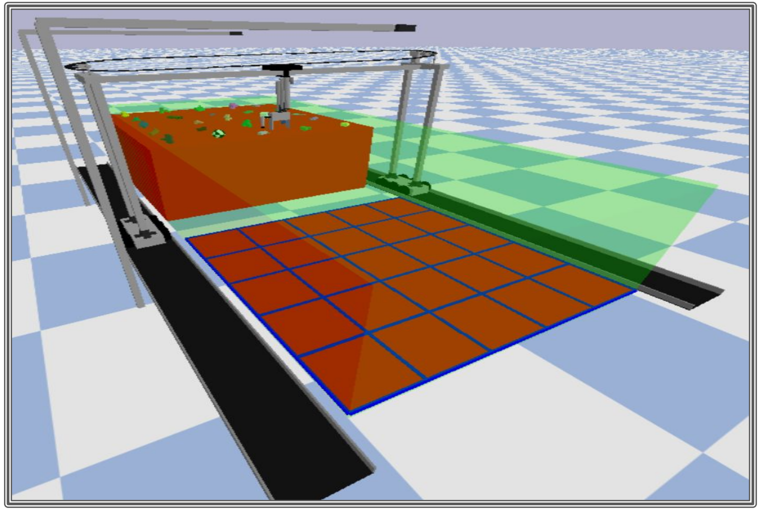
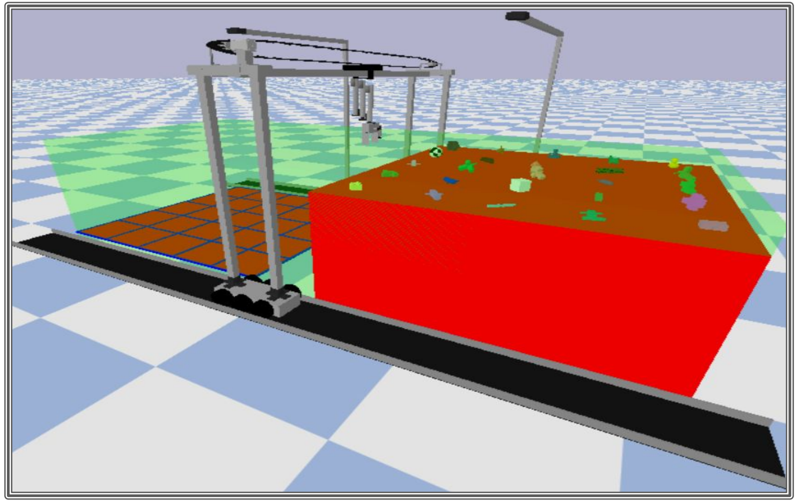




We propose a **parallel plate gripper** with a **rack and pinion** arrangement for grabbing and the suction is driven by **two vacuum pumps in parallel,** each rated with a free airflow rate of `10 CFM and an ultimate **vacuum pressure of 0.3Pa** with silicone cups in the tip of the tubes that could handle even the heaviest of objects (2kg)

Operating voltage of the robot is 12V

Workspace of the robot

3D visualisation of the robot workspace

The **workspace of our robot is a cuboid** with a volume of **4x2x1.2 m^3** (the highlighted green region). Owing to the rectangular shape, we haven't parameterized but rather used the **cartesian coordinate system**, as it is easier to constrain the working envelope. It also makes the **kinematics of the robot straightforward.**

Degree of freedom

By using **Grubler's formula**, the **degree of freedom of our robot** could be easily calculated as follows,

degrees of freedom = (sum of freedoms of the bodies) −(number of independent constraints)

Here,

$$DoF = m(N - 1 - J) + \sum_{j=0}^{j=J} f_j$$

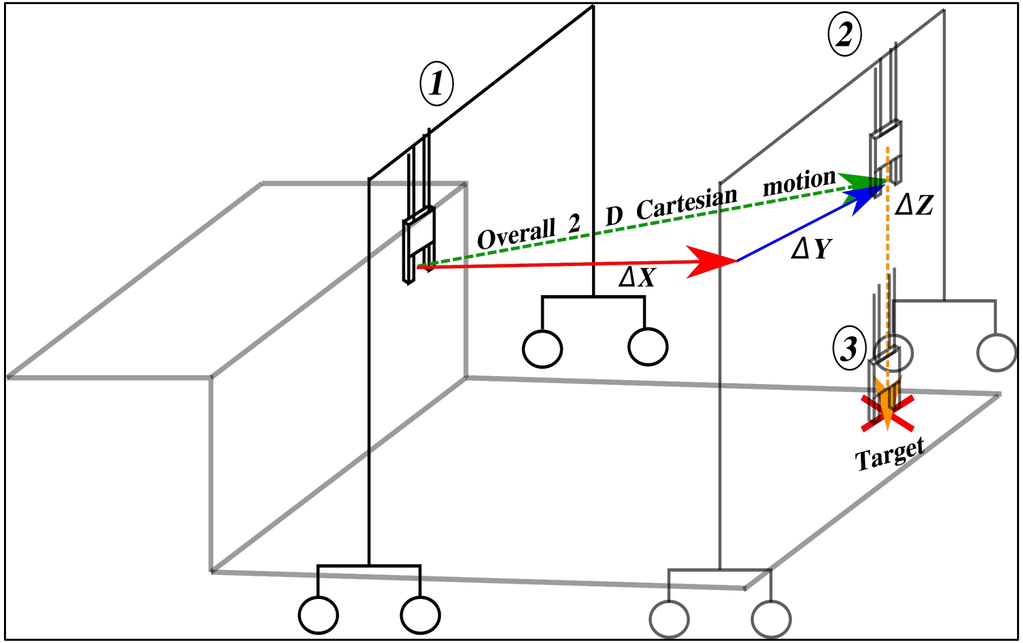
$$m = the\, number\, of\, degrees\, of\, freedom\, of\, a\, rigid\, body$$
$$J = number\, of\, Joints$$
$$N = total\, number\, of\, links$$
$$f_j = number\, of\, degrees\, of\, freedom\, provided\, by\, joint\, j$$

$$\begin{aligned} DoF &= 6 * (7 - 1 - 6) + 6 \\ &= 6 \end{aligned}$$

Note:

- The theoretical model of our robot has a **DoF of 6**, which is calculated by considering **independent joints**.
- The **mathematical model** thus derived is **accurate** in describing all possible motions of our real robot though the **realization of these mechanisms might be a bit different** in the actual robot considering the practical constraints.

Control Strategy:



The **adjoining figure** shows the **control strategy** that has been adopted by us. We have **decomposed the overall locomotion into a 3 step process** as the **displacements along X, Y, and Z direction**. Our design gives us the comfort of **addressing these motions independently** and the individual motors responsible for their motion. The displacement along the three directions is visualized by the 1, 2, and 3 respectively in the figure. We use the **classic PID control** wherein the error terms are taken as shown in the figure($\Delta x, \Delta y,$ *and* Δz).

Our **choice of sensors in implementation** is explained in detail in upcoming sections.

Control Strategy: (contD) PID equations

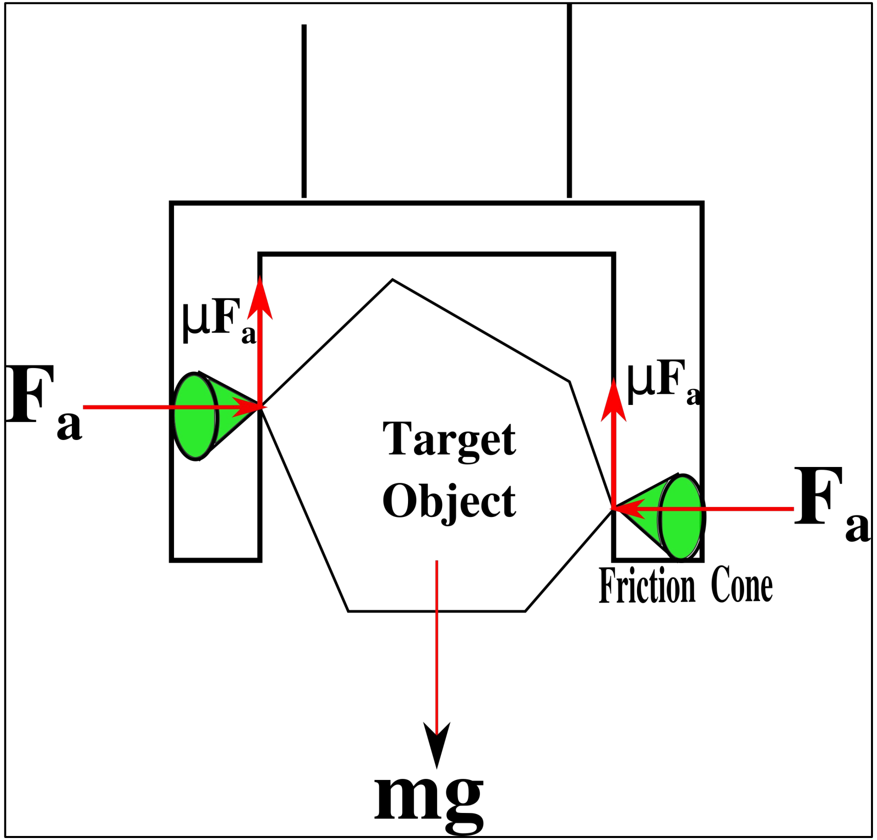
Due to the **symmetrical design of our robot**, multiple motors that contribute the motion along a given axis needs to be applied with the same torque. Thus, the **PID control law** for the entire robot, to calculate the **3 basis torques in matrix form** is written as,

$$\begin{bmatrix}\tau_x\\ \tau_y\\ \tau_z\end{bmatrix} = \begin{bmatrix}K_{p_x} & 0 & 0\\ 0 & K_{p_y} & 0\\ 0 & 0 & K_{p_z}\end{bmatrix}\begin{bmatrix}X^* - X_c\\ Y^* - Y_c\\ Z^* - Z_c\end{bmatrix} + \begin{bmatrix}K_{d_x} & 0 & 0\\ 0 & K_{d_y} & 0\\ 0 & 0 & K_{d_z}\end{bmatrix}\begin{bmatrix}V_x^* - V_{x_c}\\ V_y^* - V_{y_c}\\ V_z^* - V_{z_c}\end{bmatrix} + \begin{bmatrix}K_{i_x} & 0 & 0\\ 0 & K_{i_y} & 0\\ 0 & 0 & 0\end{bmatrix}\begin{bmatrix}\Sigma_{t=0}^{t=t}(X^* - X_c)\\ \Sigma_{t=0}^{t=t}(Y^* - Y_c)\\ \Sigma_{t=0}^{t=t}(Z^* - Z_c)\end{bmatrix}$$

where,

- τ_a , were a = x,y,z represent the **torque to be applied** in the **corresponding motors.**
- K_{p_a} , were a = x,y,z represents the **Proportional constant** of the corresponding **error term** (the units are N)
- X^*, Y^*, Z^* are the **desired positions** along the three axis
- X_c, Y_c, Z_c are the **current positions** along the three axis
- K_{d_a} , were a = x,y,z represents the **Derivative constant** of the corresponding **error term** (the units are Ns)
- V_a^* , were a = x,y,z are the **desirable velocities** along the three axis, which **for our case is equal to 0**
- V_{a_c} , were a = x,y,z are the **current velocities** along the three axis,
- K_{i_a} , were a = x,y,z represents the **Integration constant** of the corresponding **error term** (the units are N)

Payload capacity, FBD's and Force Calculation – Gripper:



At Equilibrium,

$$\mu F_a = \mathbf{mg}/2$$

$$2\mu F_a = \mathbf{mg}$$

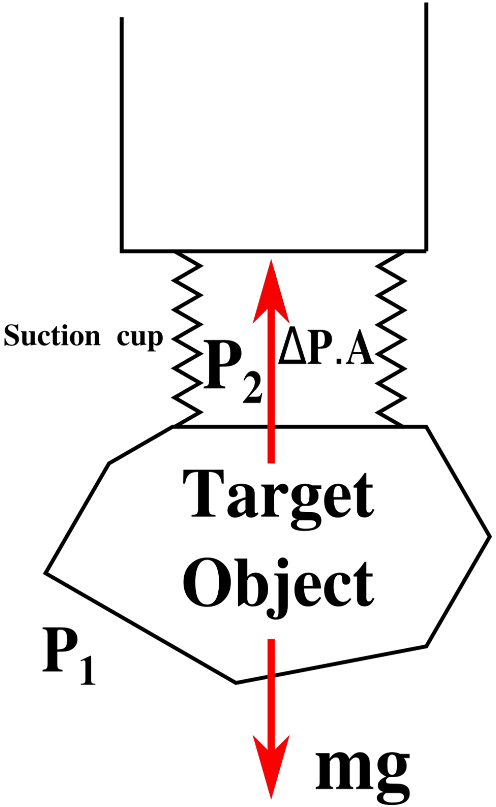
From the above equation, though **μ is a design parameter** it also varies with the nature of the **surface of the payload**, we get the following

$$0.1 < \mu < 0.9$$

$$\Rightarrow \quad 11.2\,N < F_a \leq 100\,N\ ,$$

for $\mathbf{mg} \leq 20\,N$

Payload capacity,FBD's and Calculation - Suction:



At Equilibrium,

$$Upward\,Force = Downward\,Force$$

$$\Delta P.\,A = mg$$

$$where\;\; \Delta P = P_1 - P_2$$

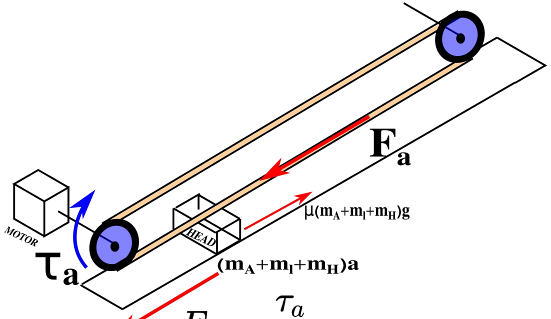
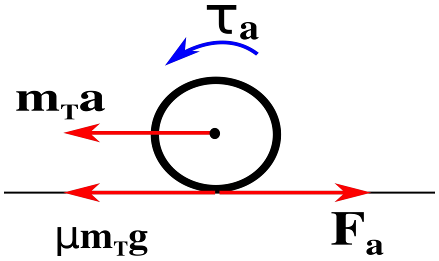
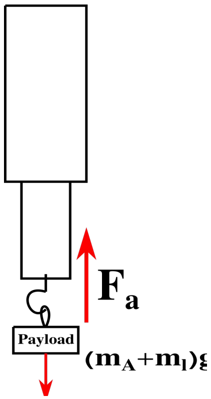
$$P_2 = P_1 - \frac{mg}{A}$$

P_2- Pressure **inside** the suction cup
P_1- **Atmospheric** pressure
A - **area of cross-section** of the suction cup

Note:

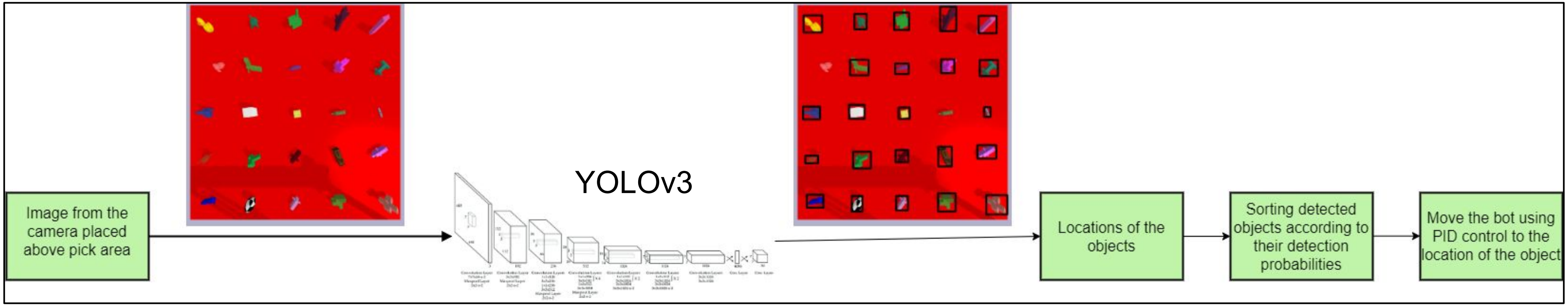
1. **Force due to acceleration of the bot in X and Y** directions is found to be **negligible from experiments in the simulation.**

Actuator Force and Joint Torque Calculations:

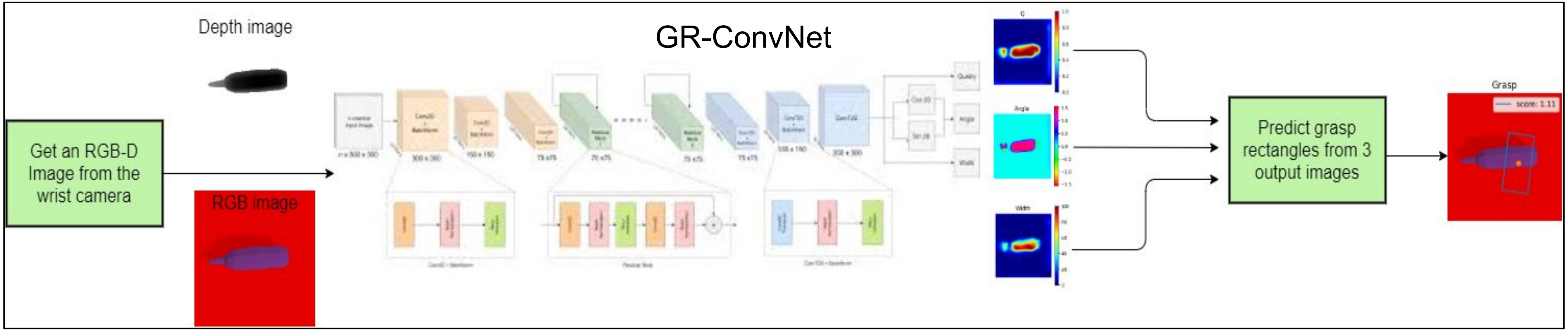
FBD of the **Head Drive** Mechanism (along y axis)	**FBD** of the **Cart Drive** Mechanism (along x axis)	**FBD** of the **Telescopic** Mechanism (along z axis)
 $F_a = \frac{\tau_a}{r}$ $(m_A + m_H + m_l)a = F_a - \mu(m_A + m_H + m_l)g$ $\Rightarrow (m_A + m_H + m_l)a = \frac{\tau_a}{r} - \mu(m_A + m_H + m_l)g$	 At the point of contact for pure rolling, $F_a = \frac{\tau_a}{r}$ $F_a = \mu m_T g$ $\Rightarrow \frac{\tau_a}{r} = \mu m_T g$	 $F_a = (m_A + m_l)g$
F_a- Force **due to belt on head** τ_a- torque by the motor m_H- mass of the **head and arm** m_A- mass of the **gripper** m_l- mass of the **load** r - the **radius of the timing pulley**	τ_a - the Torque **due to the motor in cart** F_a- Force **applied on the ground** due torque **from the motor** μ - coefficient of **friction** m_T- the **total mass** of the **bot** r - the **radius** of the wheel	F_a - Force by the **linear actuator** m_A - Mass of **gripper** m_l - Mass of the **load**

4. Software: Pipeline

Part-1: Object detection

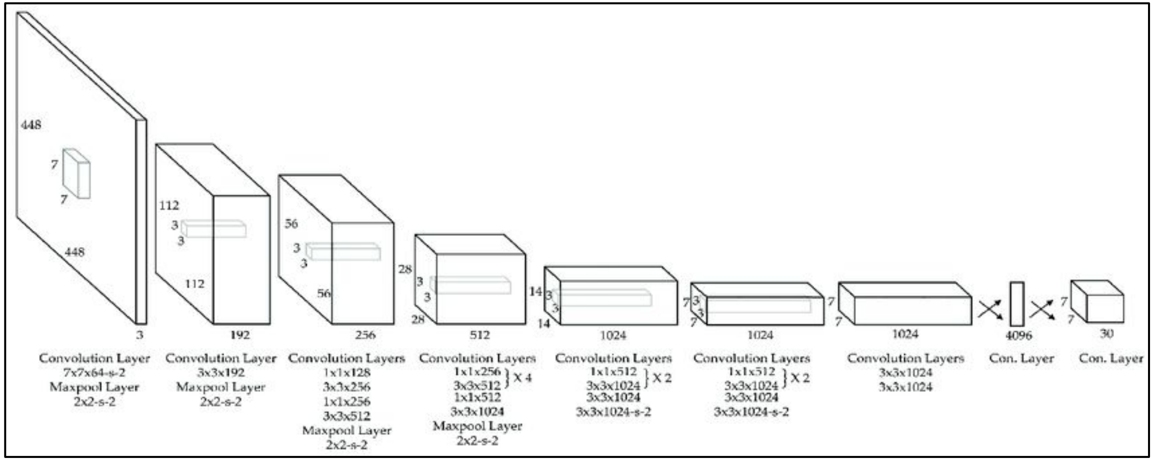


Part-2: Grasp Estimation



4.2 Object Detection

The **first step** of the pipeline is to **detect the positions of the objects** present in the **pick area**. Image from an **overhead camera** placed above the pick area is used for this purpose. Object detection is achieved using the **YOLO(You Only Look Once)** algorithm implemented on a **Convolutional Neural Network**(YOLOv3).



The approach involves a **single deep convolutional neural network** that splits the input into a grid of cells and each cell directly **predicts a bounding box** and **object classification**. The ImageAI library in python was used, which contains the state-of-the-art implementations for various image prediction and object detection models.

Preparing the dataset

The images used for training the object detection model were **generated from the simulation** of pick area in the **PyBullet simulation engine**. Each of the 250 images(resolution 1024*1024) used in the dataset consists of **25 unique objects** kept in the pick area kept at **random orientations**. LabelImg software was used to label these images in **PascalVOC format**.

Model Training

Google Colab was used for all training purposes. Transfer learning was performed using a **pre-trained model trained on the COCO dataset**. The best model had a mean Average **Precision(mAP) of 0.9977** when evaluated with Intersection over **Union(IoU) of 0.5**, **object threshold of 0.2**, and **Non-Max Suppression threshold of 0.5**.

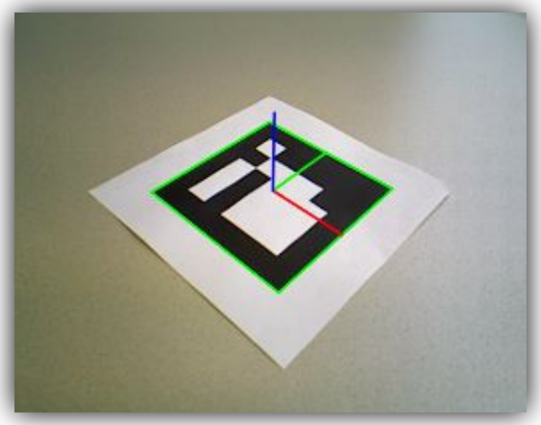
Object Detection in sparse and cluttered environments

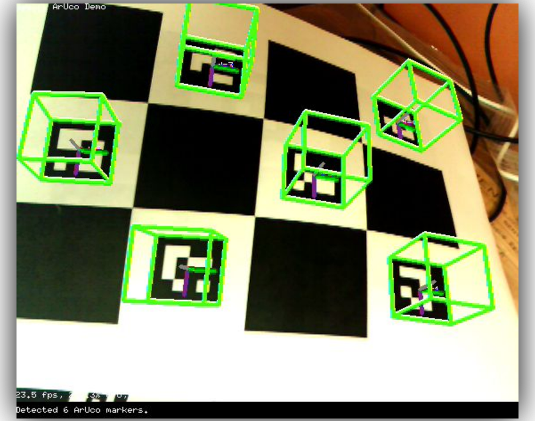
Results

The model showed good results on **novel objects** in the simulator. The model was able to detect **23 images out of 25** on average in the validation set. The model was able to **detect 9 out of 11 objects on average** when placed in a clutter. The **compute requirement** is satisfied by an **average desktop** setup.

4.3 Trajectory planning

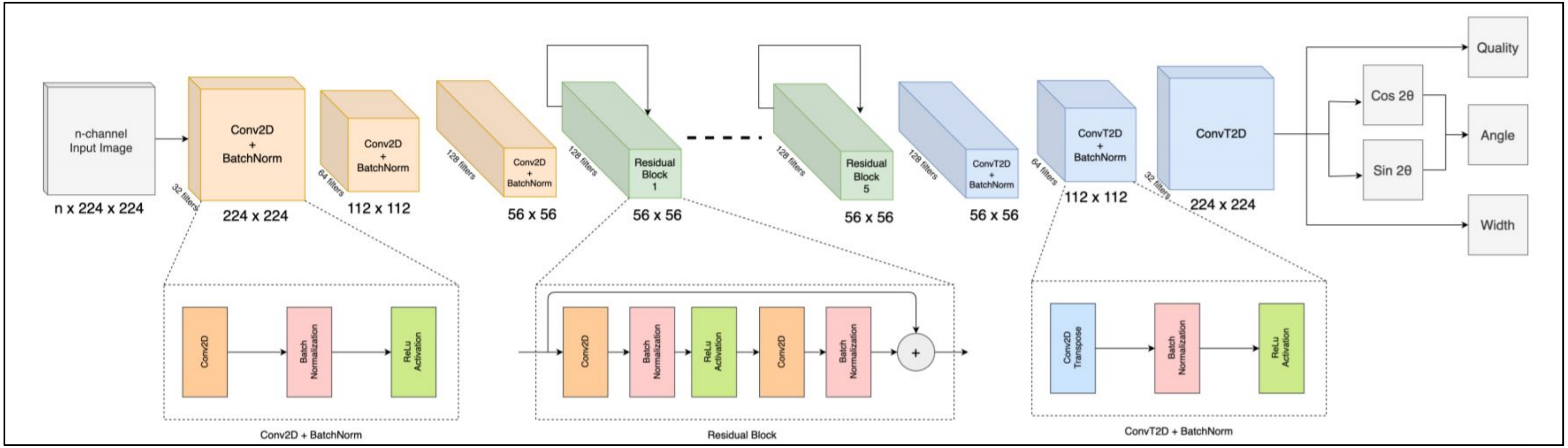
Trajectory planning was based on the **images from cameras** placed **above pick** and **drop areas**. Using the **relative positions** of the objects as detected by the **Object detection model**, the trajectory was computed based on the current position of the cart estimated using an **ArUco marker** placed on the **head** . An ArUco marker is a **synthetic square marker** composed by a **wide black border** and an **inner binary matrix** that **determines its id**.
Based on the **feedback** from cameras **above the pick/drop area**, the arm is made to **move** to the **target position** in the **X and Y** direction.(refer video)





4.4 Grasp estimation

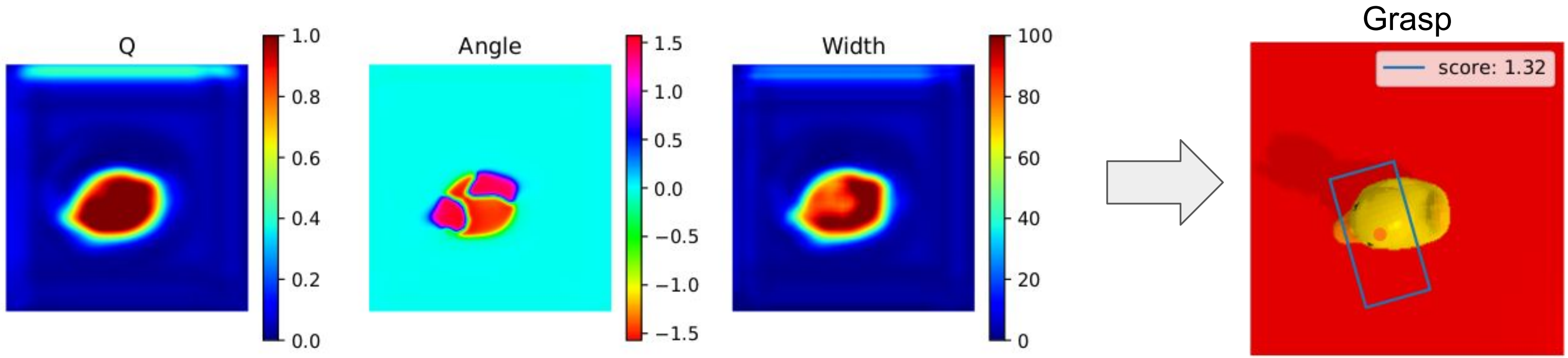
Once the picking position is reached, we need to estimate our **optimal grasp parameters**. For this we used a **Generative Residual Convolutional Neural Network(**GR-ConvNet**)** to generate **robust antipodal grasps** from images of **RGB-D camera** placed in the **wrist**.



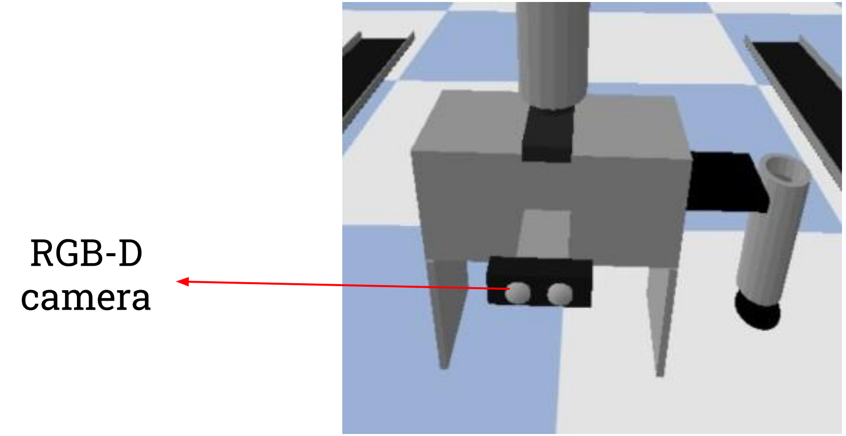
Network architecture

Model Architecture

GR-ConvNet is a **generative architecture** that takes in an **n-channel input image** and **generates pixel-wise grasps** in the form of three images ie ***Q-image***, a ***Width image***, and an ***Angle image***.



The model predicts a grasp as **G=(P,Θ,W,Q)** where,
P=(x,y), is the **most suitable point** for **grasping** the object
Θ , the **angle** which the **oriented grasping rectangle** makes with the **X-axis**, [-π/2,π/2]
W, the **width** of the **oriented rectangle** to the distance to which the plates of the gripper should close to grab the object
Q, the **grasp quality score**
The **Q-image** contains the **grasp quality scores** for **every pixel** in the input image, while the ***Angle image*** contains the **angular rotation** of the oriented grasping rectangle for **every pixel** in the image and the **width image** contains the **width of the grasp** for **all the pixels** in the image.

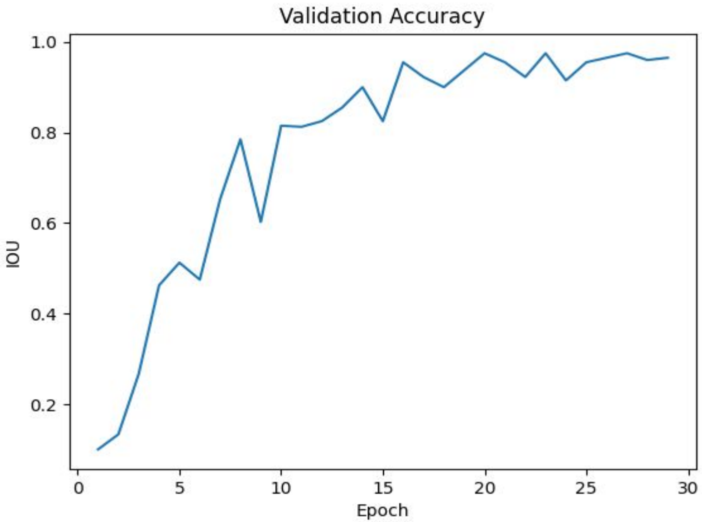


Preparing The dataset

The **dataset for grasp estimation** was again obtained from the **Pybullet simulator**. We generated **800 images(224*224)** of the objects used in training. The grasp annotations were done using labelCode, which was used in making the **Cornell Grasping dataset**. These **800 images** were then **augmented** to make a dataset of **4000 images**, each having **1 object**, containing **1 positive grasp**.

Model Training

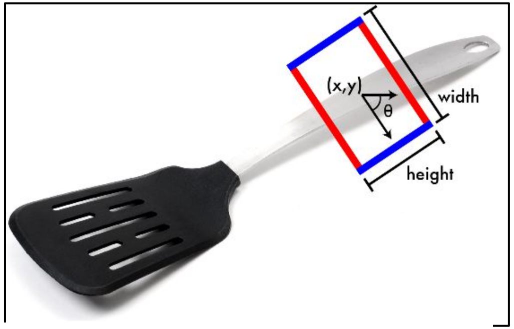
Google Colab was used for all training purposes. A **pre-trained model**, that was trained on the **augmented Cornell Grasping dataset** was utilized. This model was **further trained** on our **custom made dataset**. The model **performed well on** an **evaluation set** with an **accuracy** of nearly **98%.**

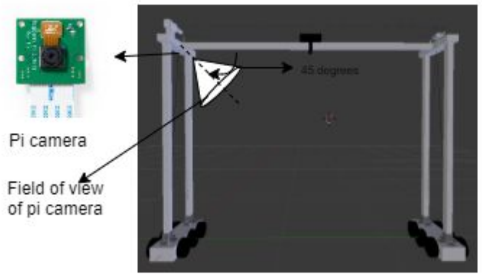


Results

The resultant model was performing well on the **learned as well as novel objects**, generating **optimal grasping points** for objects in random orientations.

4.5 Grasp planning



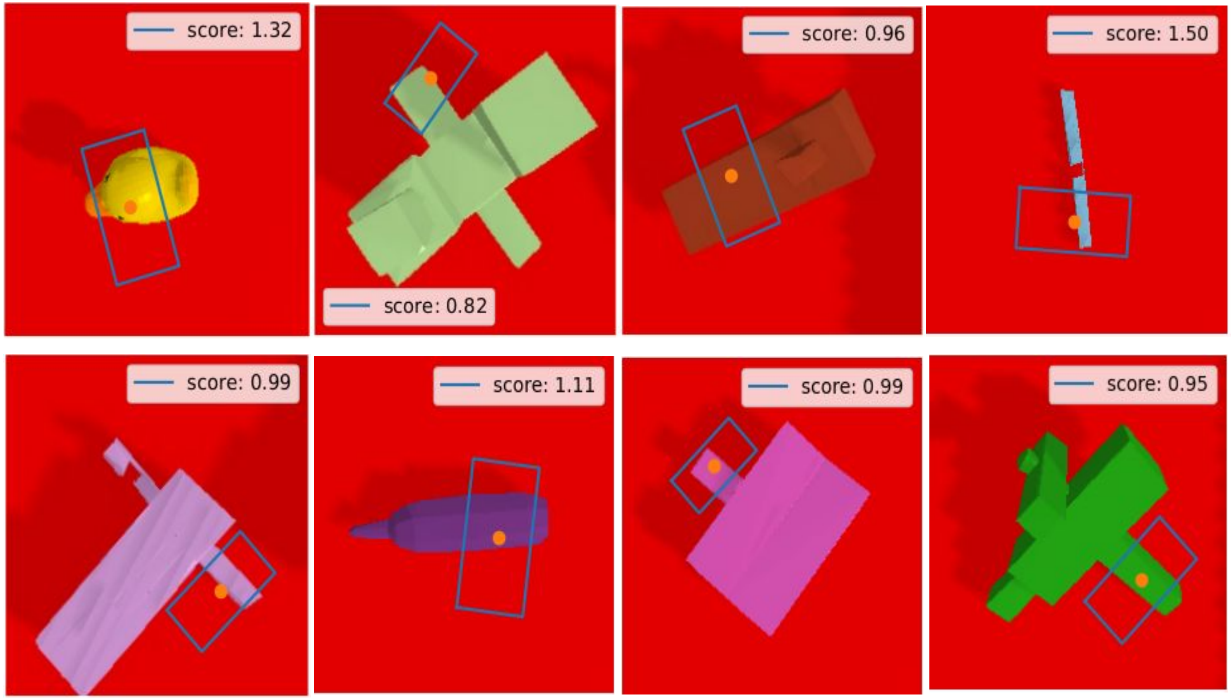


Once we have the grasp estimation, the strategy for **grabbing** becomes fairly **straight forward** as we have implemented a 3D grasping(x,y, and theta). However, when the grasp quality score (Q value) is **less than** a **threshold of 0.9**, it is statistically observed to be a **failed grasp**. Thus, we use the **help of the suction** unit here.Though we use a **depth camera**, there is hardly any information about the mass distribution, coefficient of friction, nature of material, etc. This may easily cause a failed grasp despite a good Q-Value. Once again, **upon a failed grab we rely on suction**. (refer video)

For **suction**, the suction cup is made to move to the **center of the contour** of the object.
The arm is tracked in the **Z-direction** using an **ArUco marker** attached near the gripper.

Results of our models

The grasp estimation of various oblique objects with unconventional geometry was put to test. In all of the test samples, it is observed that the gripper's **predicted** approach **rectangles** are quite **realistic** and perhaps employ the most **strategic pick orientation**. It always goes for the **relatively thinner part** of the trunk, at the same time keeping the grab position(orange dot) close to the COF to get a stable grasp. This is quite remarkable as the model has no details of the mass distribution of the target object, yet is able to generalize enough.



The **successful grabs and suctions are documented** in the attached video file in Simulations/Videos slide

Simulation/Videos

A compilation of all the demo and experimental results conducted is available in the unlisted video link shared below.Please find the attached link to get a more detailed picture of our approaches and methodologies.

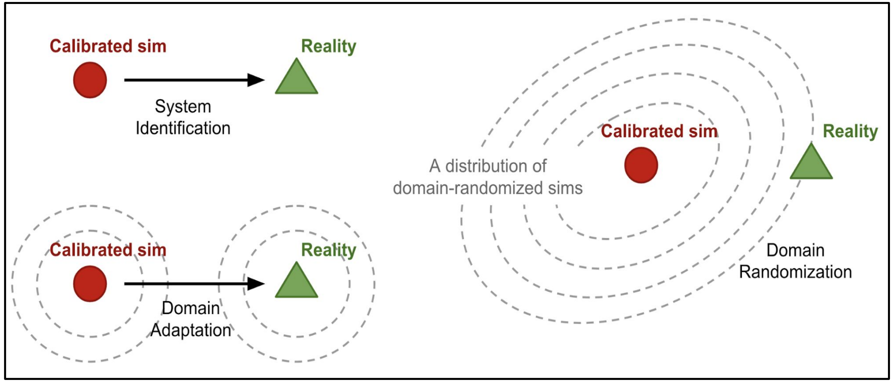
VIDEO LINK: Flipkart Grid 2.0 Round 3 Submission, Team Name : X Ash A-12

Simulation to reality

One of the **inevitable challenges** that need to be addressed in the domain of robotics is the **Sim to Real transfer**. It is loosely defined as the problems arisen due to **inaccurate modeling** of the robot, inability to capture real-world physics in the simulation, undesirable noise added to the sensory data, etc. There have been quite a few successful approaches in addressing this problem, namely

1. Domain Adaptation:

Domain adaptation (DA) refers to a set of **transfer learning techniques** developed to update the data distribution in the sim to match the real one through mapping or regularization enforced by the task model.



2. Domain Randomization:

With **domain randomization** (DR), we can create a **variety of simulated environments** with **randomized properties** and train a model that works across all of them. Likely this model can adapt to the real-world environment, as the **real system** is expected to be **one sample** in that **rich distribution of training variations.**

Simulation to reality (contD)

3.System identification:

System identification is to build a mathematical model for a physical system that required careful **calibration is necessary**. Unfortunately, calibration is expensive. Furthermore, many physical parameters of the same machine might vary significantly due to **temperature, humidity, positioning**, or **wear-and-tear in time**.

The fancier they may sound, there is no clear cut way to decide which among them works for a specific problem. Owing to the independency of data from the real world hardware, we have implemented **Domain Randomization** in our learning loop to bridge the **simulation to the reality gap**. We sampled random objects with **distinct geometry, color, mass, and texture** and created a **rich dataset of images with diverse objects** for training both the object detection and Grasp estimation models. Thus, we got robust enough models that works **fairly well even in real-world images**. However, we are also interested in **improving the robustness of our model** event the more by adding **minimalistic data(few-shot learning)** from the real world robot when we have one.Our hardware could be easily scaled up, which could be observed in our video results.

Possible Improvements

Though, we have **full fledged working solution for the problem statement in simulation**, we predicted a list of possible improvements that could enhance the performance of our hardware greatly and also help in bridging the simulation to reality gap.

* Usage of **weight estimate of the payload** with **loading cells and Force Sensing Resistors (FSR's)** placed in the fingertips of the gripper **to detect slipping and enable accurate grasping**.

* **Sensor Fusion of Motor encoders and cheap Inertial Measurement Units(like MPU-9250) along with the overhead camera** to have a **greater accuracy** and **decrease noise** in the robot's position estimate.

* Adding a simple **2 DOF wrist mechanism** to the arm that could increase the possible **grasp approach angles, if we moved on to 6D based Grasp Estimation techniques**.

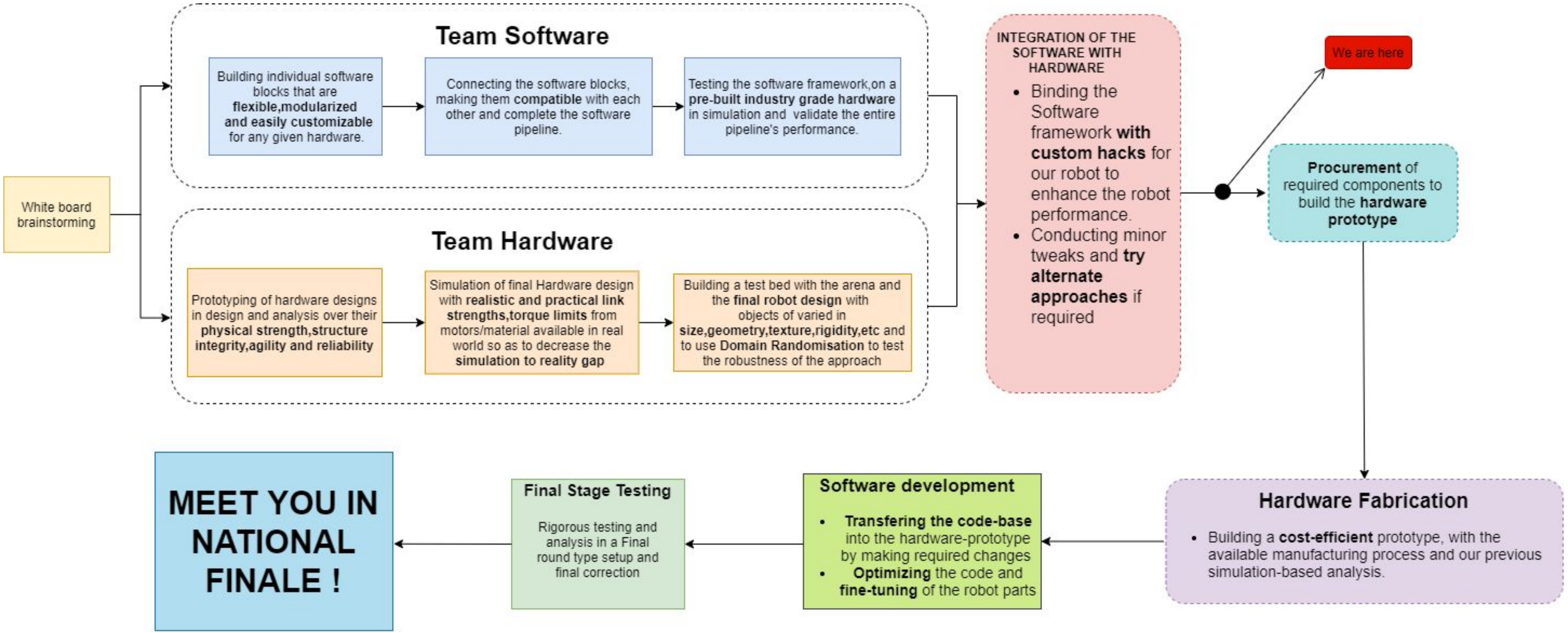
Components details, sources, price & total cost

SERIAL NO.	PART NAME	Quantity	PRICE	LINKS
1	Rack and pinion (42CRMo4)	1	340x1 = 340	LINK
2	High Torque Servo (RKI-1211)	1	500x1 = 500	LINK
3	Stepper Motor (REES52)	1	900x1 = 900	LINK
4	Linear Actuators (MACHIFIT 200N)	3	4200x3 = 12600	LINK
5	Depth Camera (INTEL REALSENSE SR305)	1	10400x1 = 10400	LINK
6	High Torque Stepper Motor (NEMA 23)	1	1300x1 = 1300	LINK
7	High Torque DC motors (12V - 21000RPM)	4	550x4 = 2200	LINK
8	Over Head Webcams (MI WIRELESS WIDE ANGLE)	2	1700x2 = 3400	LINK
9	Raspberry Pi (MODEL B - 2GB RAM)	1	2700x1 = 2700	LINK
10	Aluminium Frame (2020 T-SHAPED)	1	200 per meter 200x8.5 = 1700	LINK

Components details, sources, price & total cost (contD)

SERIAL NO.	PART NAME	Quantity	PRICE	LINKS
11	Heavy Duty Anti-Slip Wheels (100mm - 20kg/wheel)	12	300x12 = 3600	LINK
12	Vacuum pump motor (32PSI - SPARKFUN 12V)	1	1800x1 = 1800	LINK
13	Arduino Nano boards (NANO V3.0)	2	190X2 = 380	LINK
14	Motor Controllers (L298 H-BRIDGE)	2	170x2 = 340	LINK
15	GT2 Timing Belt (10M-GT2)	1	1000x1 = 1000	LINK
16	Pulleys for GT2 Belt (20 TEETH TIMING PULLEY)	4	120x4 = 480	LINK
17	Pi Cam (OMNIVISION 5647)	1	390x1 = 390	LINK
18	Suction cup (PSUB2.5)	1	300x1 = 300	LINK
20	Fabrication Costs	-	3000x1 = 3000	-
TOTAL			47,330 INR	

Execution plan with timelines



Current Progress

1. As showcased in the video, we have a **robust solution** for the given problem statement which has been modelled and **validated in the Pybullet,** a python simulation module built around the Bullet physics SDK.Thus, addressing the minor tweaks that could be raised by Sim2Real transfer, we are confident of fabricating a real world robot that is highly cost efficient and reliable.

2. The flow chart in slide 35 elaborates our **exact execution plan and current standing**.**The proof of concept in simulation and the attached CAD files** leave us in a position successfully to **start the fabrication process**.

3. Due to the current pandemic situation, we are quite unclear about the starting the fabrication process and are eagerly waiting for the **guidance of the mentors from Flipkart as instructed by them in the webinar session**.However we have done some design changes owing to the availability of the required components and have attached the rough price estimate for the same.(refer slide 33 and 34).

4. We are ready to submit the code base of our solution for your perusal if required.

References

- Cartman: The low-cost Cartesian Manipulator that won the Amazon Robotics Challenge
- Design of a Multi-Modal End-Effector and Grasping System
- YOLOv3: An Incremental Improvement
- Antipodal Robotic Grasping using GR-ConvNet
- Real Time Grasp Detection using Convolutional Neural Networks
- ImageAI: A python library for object detection and image predictions
- LabelImg: A graphical image annotation tool
- LabelCode: Annotation tool for grasping dataset
- Robotic Pick-and-Place of Novel Objects in Clutter with Multi-Affordance Grasping and Cross-Domain Image Matching
- Vision-based Robotic Grasp Detection From Object Localization, Object Pose Estimation To Grasp Estimation: A Review
- Robotic Pick-and-Place of Novel Objects in Clutter with Multi-Affordance Grasping and Cross-Domain Image Matching
- Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World