

# Baseballs Picking Robot (BPR, pronounced fondly *beeper*) Neelima Valluru

## Functional Specification

This document provides the functional specification of a robot that is designed to pick up baseballs during batting practices on the field. This can be programmed to detect and pick other types of balls too.

## The Idea

I got tired of picking baseball during my brother's on-field batting practices. Searched for a robot like the Roomba to do this chore. There isn't one. Decided to make one myself.

## Requirements

- It needs to pick-up baseballs or any other balls that are used in practice sessions. Like softball, tennis, and golf.
- Needs to be lightweight
- Easy to transport
- Power efficient to last for several batters
- Should be able to traverse terrains like infield, grass, hard-court, clay, or putting greens.
- Should be autonomous, using camera vision
- Collect and carry at least ten balls at a time, a typical practice session of a batter is fifty balls
- Bring the collected balls back to the first base. The charger is at the first base, if we could run a cable to it.

## Electrical Design

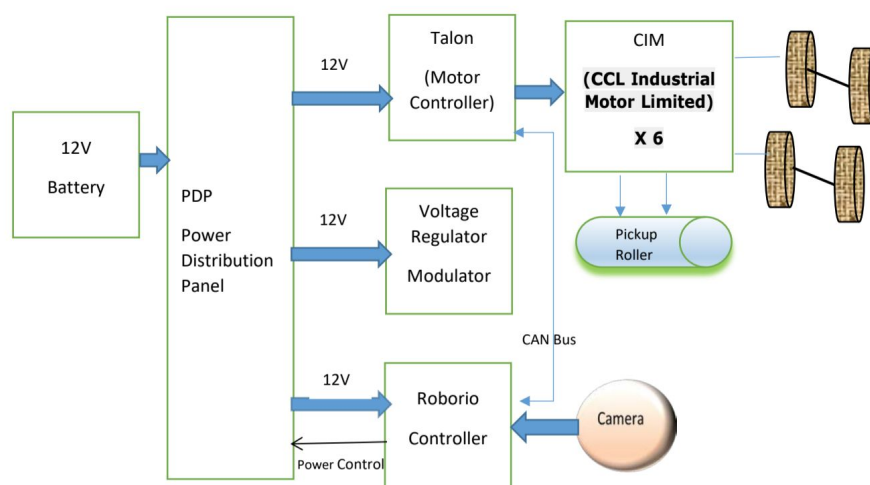


Fig 1. Block Diagram of Baseball Robot

## Mechanical Design

### Control System

- Roborio
- PDP
- Talons

### Drive System

- 4 CIMS
- 

## Controls Subsystems

Roborio is the main robot controller. It runs Linux O.S and runs our application that

- Surveys the baseball field, picks up the baseball, collects ten balls at a time, and brings it back to the first base.
- Receives the visual data from the camera
- Controls the CIM
- Drive train

### Power Distribution Panel

- This receives 12V power from the battery and distributes based on control signals from Roborio.

### Talons

- Receives CANBus signals from Roborio and provides the correct voltage to CIMS.

### CIM

- These are motors coupled to the rollers and the wheels.

### Vision

- Vision is implemented with a regular web camera to get the video streaming of the field.

### Algorithm

- In my application there are two threads running simultaneously. One for the main loop and the second for the vision tracking algorithm.
- The two loops are for the sake of meeting the time constraints of the vision code.
- The main loop controls
  - four CIMS that makeup the drive train system
  - two CIMS that drive the roller pickup system
- The vision loop receives the camera data at 30FPS, decodes it, and analyses it for ball pattern matching.
  - Uses OpenCV libraries to implement the visual functions. Detects color, shape, and a specific pattern to classify before picking or rejecting the object.