# Introduction

## Problem:

People with limited mobility often lose independence in the simplest day-to-day actions such as pouring a glass of water or making a cup of coffee. These aren't special tasks in any way and that's exactly why losing them hits so hard.

When you can't do basic actions on your own, your day becomes shaped around what's physically possible rather than what you want or need at the moment. When those basics require a caregiver, routine living can quietly turn into constant dependence.You need to time your needs around someone else's availability which can lead to hesitance in asking for help or choosing to go without because it feels easier than asking again. Over time, those repeated moments of not being fully in control of your own simple preferences can be discouraging.

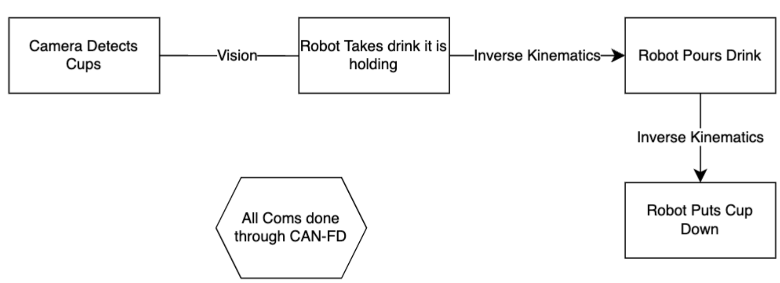
When everyday actions become requests, it can feel like your privacy shrinks, your dignity is compromised, and your autonomy is taken away in small increments. Even with supportive caregivers, the emotional weight of needing help for basics can be draining and can destroy your mentality over time.

## Solution:

What we propose is a robotic arm capable of performing those day-to-day tasks. The demonstration will be for one of those many simple tasks: pouring a drink.

To delve into the details, we propose a 6-DoF robotic arm that is fully actuated. The sensors will include cameras for vision detection, Time of Flight for basic depth detection and strain gauge amplifiers to measure the force applied onto an object. This project will also need a more powerful on-board computer to deal with data processing from our pcb + microcontroller sensor array,vision algorithms, and to accurately create commands through inverse kinematics that drive the robotic arms motors.

## Visual Aid:

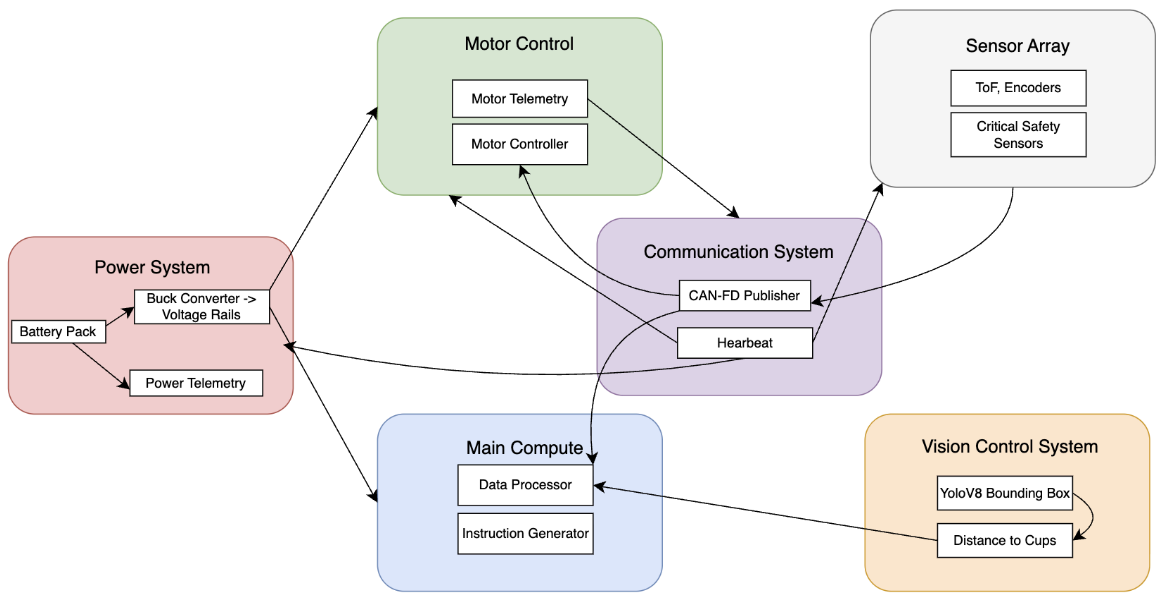


High-Level Requirements List:

- 95% Success rate in pouring a drink in normal indoor lighting conditions
- Robot shall detect human and obstacle collision and stop within 20 ms
- Robot Accuracy is within 10mm in the control loop

# Design

Block Diagram:




Subsystem Overview:

There are multiple subsystems involved in this project.

Vision Subsystem:

- Detect cups and detect glass used to pour into cups.
- Custom tuned YoloV8 model here.
- Helps us take the scene and estimate pose for pouring the water into the cups
- Use stereo cameras to find the difference in perspective for the proper distance to our bounded object.

PCB Sensor Array System:

- Our robotic arm hosts the following electronics: strain gauge amplifiers, ToFs, Load Sensors, as well as magnetic encoders.
- We need a sensor array system that reads these things.
- Distributed pcb components which are identical in function but read from different links of the robotic arm. This gives multiple benefits of isolated information per link + joint and more robust communication to each joint in response to the sensor inputs.

Quick overview of each sensors purpose:

- Strain Gauge Amplifier - Finds the force applied by the end effector to know when to stop clenching our gripper
- ToFs - Useful for aiding our camera through sensor fusion on depth perception as well as measuring how far apart our gripper peices are
- Load Sensors - The load sensors are used to prevent the arm from running into people and hurting them in places which is out of our cameras FoV giving full 360 degree protection.
- Encoders: The position each of our motors are at.

## Communication System:

- The communication system will be implemented via CAN-FD. This is how each component in the robotic arm sends data to each other
- Each joint/link gets a unique ID on the bus for proper communication.
- CAN-FD is ideal here because of its differential pairing which makes it more resistant to EMI that can come from motors.

## Motor Control System:

- These are the motor drivers that physically move the robot and publish data about the motor. Will be an FOC based motor driver.

## Power System:

- Converts the primary input into regulated rails for the robot
- Separates high-current motor bus from low-noise logic rails to prevent brownouts and EMI.
- Feeds the Motor Control System, Main Compute System, and all distributed Sensor PCBs
- Reports voltage & current status to Main Compute so the robot can derate or enter a safe state before power faults cause unsafe behavior.

## Main Compute System:

- Determine the proper instruction to the arm
- Safety signals get preference here.

## Subsystem Requirements:

### Vision Subsystem:

- Must Publish 3D target pose to Main Compute at minimum of 15 Hz with minimum 100 ms latency for smooth closed-loop control
- Depth accuracy of at least +/-10 mm at 0.5 m and more than 0.85 recall on cup classes in the environment. If not the arm cannot reliably align and pour.
- Vision degraded fault within 200 ms of confidence dropping below a specific threshold, sending confidence with the timestamp to the Main Compute so the system can slow down when our perception is not that reliable. If this is not implemented, the vision subsystem may fail to accurately detect cups and cause us to fail the entire process.

### PCB Sensor Array System:

- Must sample with a timestamp for our sensors locally and publish over CAN-FD with bounded timing. We want encoders at least 1 kHz, force and contact sensors at least 500 Hz, and no more than 20 ms worst-case sensor to Main Compute latency. This allows for stable control and fast collision response
- Provide proper signal integrity to detect grip force thresholds and contact events. If this is not done then the gripper may grip too tight and fail the task.

### Communication System:

- The CAN-FD bus delivers safety critical frames in at most 10 ms. If this isn't hit then safety issues will arise.
- Implement a heartbeat with at least 10 Hz per node and raise communication faults if the heartbeat is missing for more than 100 ms. This way we know if a component is not working properly and can either stop the process safely or readjust accordingly.
- Maintain reliable operation in a motor-EMI environment. If not our signals will be corrupted and cause false data to be sent to our computer.

### Motor Control System:

- Implement local FOC control with current loop at least 10 kHz and accept torque,velocity, and position commands with a maximum of 10 ms latency. If we have more than 10ms latency and less than 10kHz of current loop control then our robot wont be able to properly execute the given task because it is too slow to follow what the brain is telling it to do.
- Enforce hardware torque and current limits and execute a safe response. If not, someone may get crushed or the cup will get destroyed.
- Publish Motor states to main compute with at least 100 Hz. If not we can not properly update motor states with relevancy to other data streams.

### Power System:

- Supply regulated rails with defined tolerances. Sensors at 5V, Motor at designated voltage, and designated Main Compute voltage. If this isn't done then our system will fail to work because of insufficient power

- Provide power telemetry to Main Compute at  Main Compute at least 10 Hz. If this rate is not hit then when there are power failures we can not maximize our reaction time.

Main Compute System:

- Must Prioritize safety signals and issue a safety command within a maximum of 20ms. If not, someone may get dangerously hurt.
- Output joint commands over CAN-FD at a minimum of 100 Hz and consume timestamped vision and sensor messages. If this update rate is not hit then stable closed-loop alignment isn't possible.

## Tolerance Analysis:

Variables:

R_allow = Minimum lateral offset at the instant we start pouring.
R_cup = Cup inner radius.
R_stream = The stream radius/uncertainty.
R_margin = The rim safety margin accounting for things like cup movement, buffer or outside environment factors.

Formula:

R_allow = R_cup - R_stream - R_margin.

Some components that can cause us to be outside of R_allow include,
The vision subsystem having inaccuracies, the PID Loop on our motor FOC control being unstable,kinematic error,and EMI interference on signals causing improper judgement.

## Ethics & Safety:

To keep ethics and safety in mind when making this robotic arm,we adhere to the IEEE and ACM codes of ethics. We put safety first and are transparent about the limitations of our arms. We avoid harm through making very conservative design choices as well as sharp limitations. During development, we face a huge risk with the end-effector crushing someone or something, unexpected motion, and electrical faults. To limit the risk factor we make sure to define clear exclusion zones based on the mechanical limits of our arm as well as an emergency stop that can be pressed to immediately kill power to our robot. We align our approach with ISO 10218 and ISO/TS 15066. We also follow lab safety policies for electrical and shop work. To prevent misuse we set sharp limits on torque as well as joint velocity + acceleration to ensure that no one can get hurt even if the robot does have unpredictable behavior.