Preliminary Design Review

Range Extending System to Complement Underground Exploration **(RESCUE)**

Customer: Prof. Eric Frew

University Of Colorado Boulder

Table of Contents

Project Overview

Main Objective:	Improving subterranean unmanned ground vehicles' ability to sense locations that the vehicle cannot travel to or are obstructed from the onboard sensors' field of view or range.
Specific Application:	MARBLE's Clearpath Husky UGV being used in DARPA's Subterranean Challenge.
Proposed Solution:	A folding robotic arm utilizing an RGBD camera, CO_2 Sensor, and AHRS Sensor that is mounted to the top of MARBLE's Husky.

Figure 1: Clearpath Husky

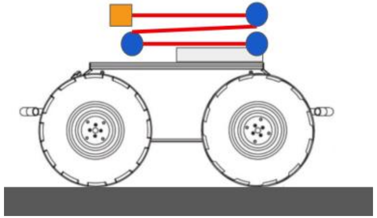


Figure 2: Side view of clearpath husky with arm stowed

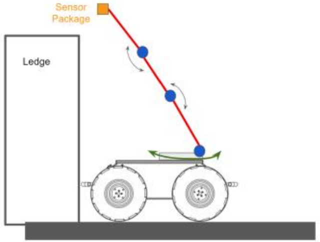


Figure 3: Side view of clearpath husky with arm extended

Project Motivation

Problem:

- MARBLE’s UGV has **limited sensing** capabilities in comparison to other competitors in DARPA’s Subterranean Challenge.
- UGVs offer greater endurance than UAVs, however, **field of view and mobility are limited**.
- Ledges, crevices, holes, and elevated platforms are currently **impassible** and/or out of the FOV of the UGV.

Existing Solutions:

- UAVs that are either independent or dock on a UGV
- Telescoping sensor systems
- Crawling/climbing drones

Project Mission Statement

RESCUE's mission is to expand the sensing capabilities of UGVs by developing a system which can be utilized not only by MARBLE in DARPA's Subterranean Challenge but also by other UGVs that share the common goal of identifying objects in difficult to reach locations.

CONOPS

Range Extending System to Complement Underground Exploration (RESCUE)
Mission Concept Of Operation

CONOPS

Range Extending System to Complement Underground Exploration (RESCUE) Mission Concept Of Operation

0. RESCUE traveling with MARBLE's HUSKY in“standby mode

CONFIRMED ARTIFACTS:

LEVEL1: HIGH-VISIBILITY SIGNATURES
LEVEL2: LOW-VISIBILITY SIGNATURES
LEVEL3: NON-VISIBILITY SIGNATURES

* Levels reflect artifacts with similar signatures not difficulty

CONOPS

Range Extending System to Complement Underground Exploration (RESCUE) Mission Concept Of Operation

0. RESCUE traveling with MARBLE's HUSKY in"standby mode

1. MARBLE team sends a firing command to RESCUE to deploy to a location relative to the HUSKY

CONFIRMED ARTIFACTS:

LEVEL1: HIGH-VISIBILITY SIGNATURES
LEVEL2: LOW-VISIBILITY SIGNATURES
LEVEL3: NON-VISIBILITY SIGNATURES
* Levels reflect artifacts with similar signatures not difficulty

CONOPS

Range Extending System to Complement Underground Exploration (RESCUE)
Mission Concept Of Operation

0. RESCUE traveling with MARBLE's HUSKY in"standby mode
1. MARBLE team sends a firing command to RESCUE to deploy to a location relative to the HUSKY
2. RESCUE starts deployment process and switches to active mode

CONFIRMED ARTIFACTS:

LEVEL1: HIGH-VISIBILITY SIGNATURES
LEVEL2: LOW-VISIBILITY SIGNATURES
LEVEL3: NON-VISIBILITY SIGNATURES
* Levels reflect artifacts with similar signatures not difficulty

CONOPS

Range Extending System to Complement Underground Exploration (RESCUE) Mission Concept Of Operation

0. RESCUE traveling with MARBLE's HUSKY in "standby mode
1. MARBLE team sends a firing command to RESCUE to deploy to a location relative to the HUSKY
2. RESCUE starts deployment process and switches to active mode
3. RESCUE collects sensory data for potential artifacts

CONFIRMED ARTIFACTS:

LEVEL1: HIGH-VISIBILITY SIGNATURES
LEVEL2: LOW-VISIBILITY SIGNATURES
LEVEL3: NON-VISIBILITY SIGNATURES
* Levels reflect artifacts with similar signatures not difficulty

CONOPS

Range Extending System to Complement Underground Exploration (RESCUE) Mission Concept Of Operation

0. RESCUE traveling with MARBLE's HUSKY in"standby mode
1. MARBLE team sends a firing command to RESCUE to deploy to a location relative to the HUSKY
2. RESCUE starts deployment process and switches to active mode
3. RESCUE collects sensory data for potential artifacts
4. RESCUE transmits data to the HUSKY for processing

Confirmed Artifacts:

Level1: High-Visibility Signatures
Level2: Low-Visibility Signatures
Level3: Non-Visibility Signatures
* Levels reflect artifacts with similar signatures not difficulty

CONOPS

Range Extending System to Complement Underground Exploration (RESCUE) Mission Concept Of Operation

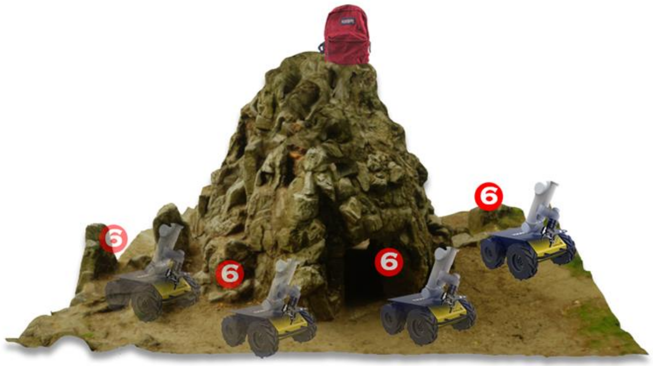
0. RESCUE traveling with MARBLE's HUSKY in"standby mode
1. MARBLE team sends a firing command to RESCUE to deploy to a location relative to the HUSKY
2. RESCUE starts deployment process and switches to active mode
3. RESCUE collects sensory data for potential artifacts
4. RESCUE transmits data to the HUSKY for processing
5. If RESCUE recieves no further commands, RESCUE returns to standy mode

Confirmed Artifacts:

Level1: High-Visibility Signatures
Level2: Low-Visibility Signatures
Level3: Non-Visibility Signatures
* Levels reflect artifacts with similar signatures not difficulty

CONOPS

Range Extending System to Complement Underground Exploration (RESCUE) Mission Concept Of Operation



0. RESCUE traveling with MARBLE's HUSKY in"standby mode
1. MARBLE team sends a firing command to RESCUE to deploy to a location relative to the HUSKY
2. RESCUE starts deployment process and switches to active mode
3. RESCUE collects sensory data for potential artifacts
4. RESCUE transmits data to the HUSKY for processing
5. If RESCUE recieves no further commands, RESCUE returns to standy mode
6. Back to stage 0

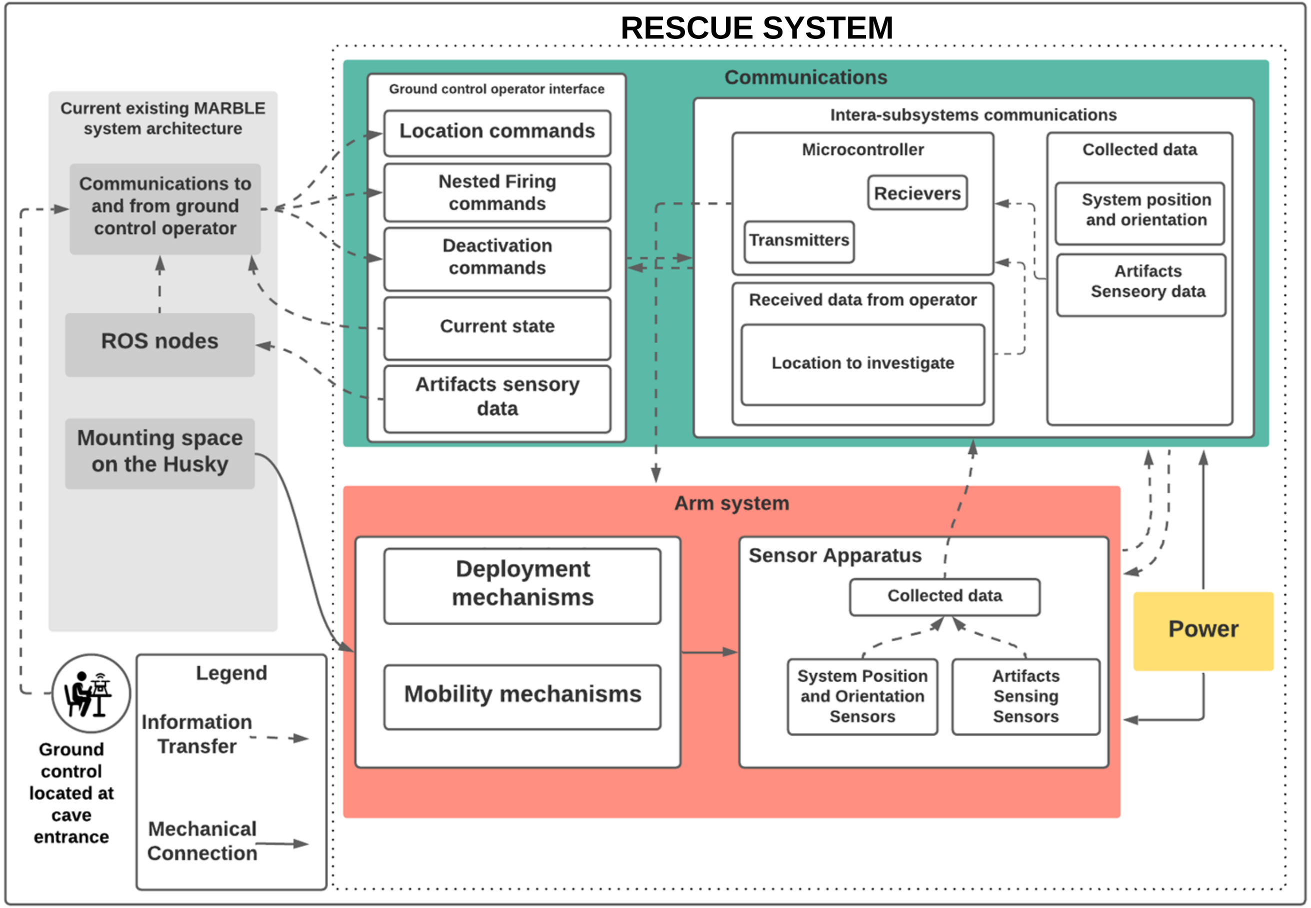
CONFIRMED ARTIFACTS:

LEVEL1: HIGH-VISIBILITY SIGNATURES
LEVEL2: LOW-VISIBILITY SIGNATURES
LEVEL3: NON-VISIBILITY SIGNATURES
* Levels reflect artifacts with similar signatures not difficulty

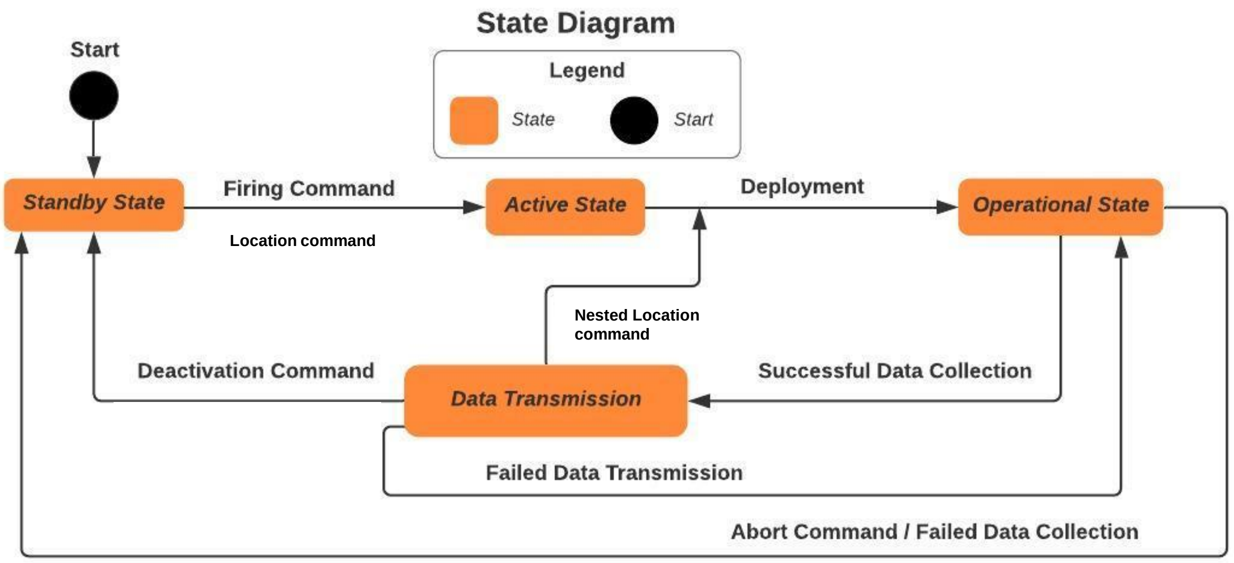
CONOPS

Range Extending System to Complement Underground Exploration (RESCUE)
Mission Concept Of Operation

FBD


RESCUE SYSTEM
Current existing MARBLE system architecture
Communications to and from ground control operator
ROS nodes
Mounting space on the Husky
Communications
Ground control operator interface
Location commands
Nested Firing commands
Deactivation commands
Current state
Artifacts sensory data
Intera-subsystems communications
Microcontroller
Recievers
Transmitters
Received data from operator
Location to investigate
Collected data
System position and orientation
Artifacts Senseory data
Arm system
Deployment mechanisms
Mobility mechanisms
Sensor Apparatus
Collected data
System Position and Orientation Sensors
Artifacts Sensing Sensors
Power
Ground control located at cave entrance
Legend
Information Transfer
Mechanical Connection

State Diagram



Project Definition: Functional Requirements (MOST RELEVANT)

FR1.1	At minimum, RESCUE shall have the ability to physically reach a location that is at least 1 meter but not more than 5 meters from the apparatus' mounted position on the MARBLE Clearpath Husky.	Physical reach
FR2.1	The sensing system shall be able to sense DARPA subterranean challenge competition artifacts.	Artifacts sensing
FR3.1	RESCUE shall determine and report its location and orientation relative to the ground robot.	System Position and Orientation
FR4.1	When in its standby configuration, RESCUE shall be compatible with the MARBLE team's Clearpath Husky.	Deployment: constraints
FR6.1	RESCUE shall communicate its sensed data with MARBLE and this process shall not interfere with MARBLE's communication systems. RESCUE shall be able to receive firing commands, nested firing commands, and deactivation commands from MARBLE's team.	Communication

Baseline Design

Baseline Design: Overview

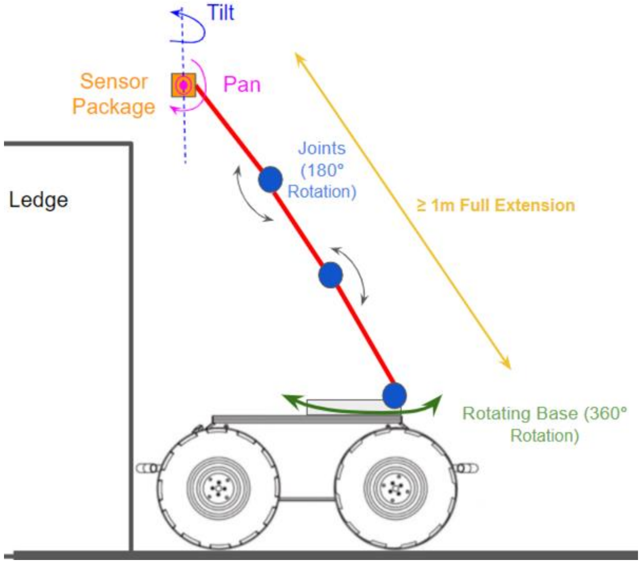


Figure 1: Side view of Deployed arm

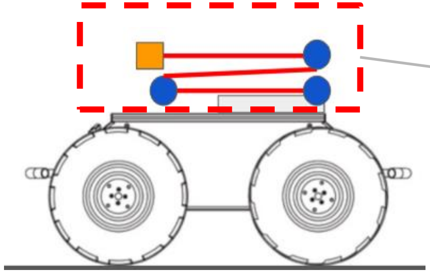


Figure 2: Sideview of Stowed arm

Sensors:
-Intel Realsense D435
-Sensirion SCD30 CO2
- 3-Space™ Watertight USB/RS232 AHRS

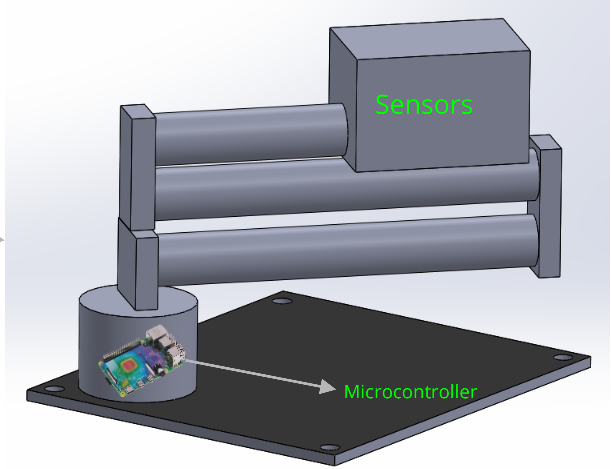


Figure 3: Sideview of Stowed arm

Baseline Design: Overview

These are the only discussed elements in baseline design, everything else available in Backups

- Arm Design
- Joints Design
- Wrist Design
- Base/Waist Design
- Sensor packaging

Mechanical

- Visual Sensor
- CO_2 Sensor
- Attitude determination
- Attitude controls
- Microcontroller
- Communications

Software

Most Relevant

- Arm Design
- Wrist Design
- Joint Design

Mechanical

- Visual Sensor
- Attitude determination
- Attitude controls
- Communications

Software

Baseline Design: **Mechanical Structure**

Baseline Design: Mechanical Structure

Trade Options

- **Robotic Arm**, Flying Drone, Sensor Launcher

Purpose

- **Physically moves sensor package to Husky inaccessible location**

Capabilities

- At least 1m extended reach (TDR1.1.1)
- 90° sensor package rotation about at least 1 axis (TDR1.1.3)
- Directional change, 45° or more about at least 1 axis (1.1.2)
- Meet MARBLE's storage requirements (FR 4.1)
  - 10kg
  - 38x45x30 cm^3 stowed

FR1.1	At minimum, RESCUE shall have the ability to physically reach a location that is **at least 1 meter** but not more than 5 meters from the apparatus' mounted position on the MARBLE Clearpath Husky.	Physical Reach

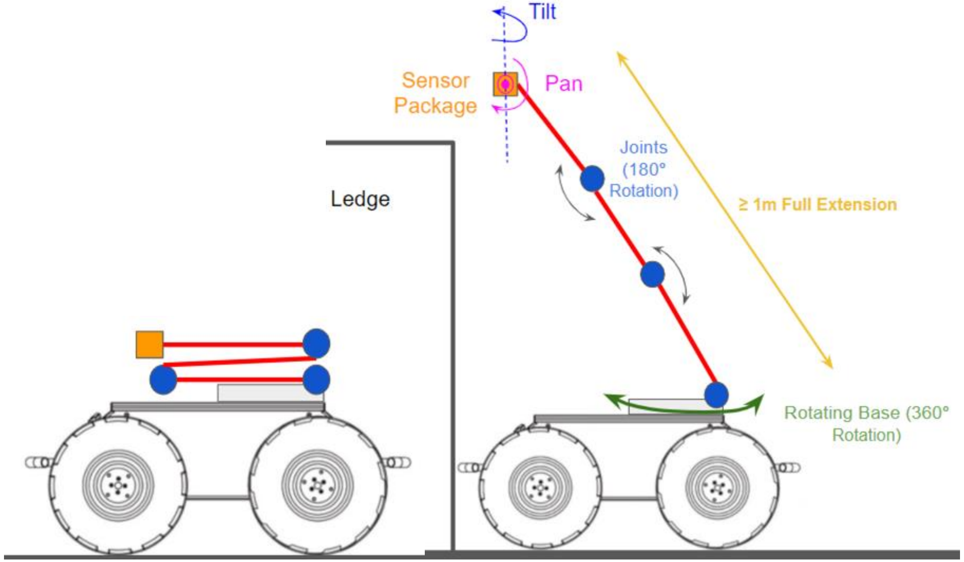


Figure 1: Stowed Arm Configuration

Figure 2: Deployed Arm Configuration

Baseline Design: Arm Design

3 Member Arm Design

- Stowed diagonally to maximize member lengths within given area restriction
- Three members + Joints + Wrist -> **~1.31 m physical reach**
- Members- cylindrical tubing with wires inside
- Total arm mass ~ 1.15 kg

FR4.1	When in its standby configuration, RESCUE **shall be compatible** with the MARBLE team's Clearpath Husky.	Deployment: constraints

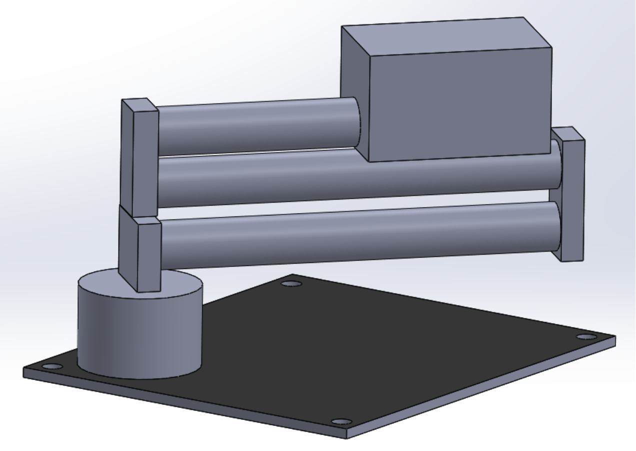


Figure 1: 3 member arm in stowed configuration

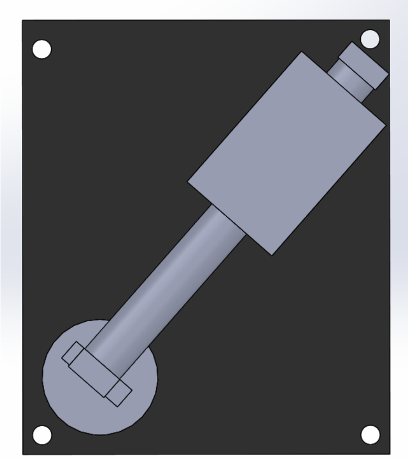


Figure 2: Top-down view of diagonal stowing

Baseline Design: Joint Design

180 degree rotation for all elbow joints

FR1.1	At minimum, RESCUE shall have the ability to physically reach a location that is **at least 1 meter** but not more than 5 meters from the apparatus' mounted position on the MARBLE Clearpath Husky.	Physical Reach

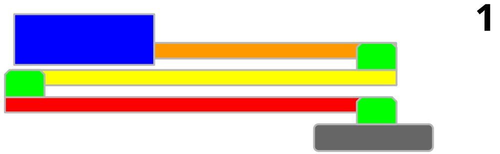


Figure 1: Fully collapsed

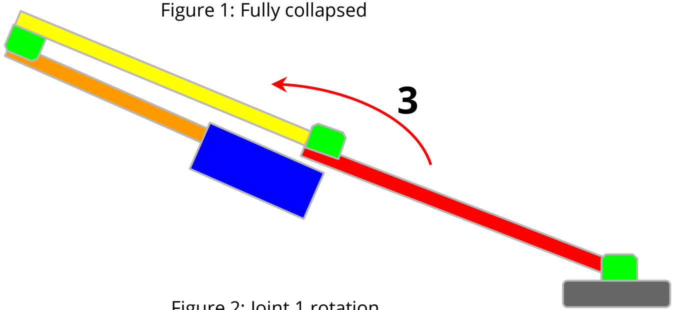


Figure 2: Joint 1 rotation

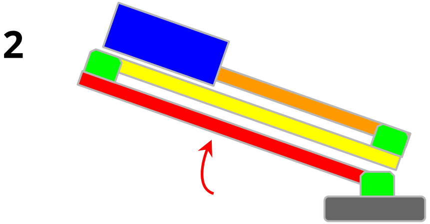


Figure 3: Base joint rotation

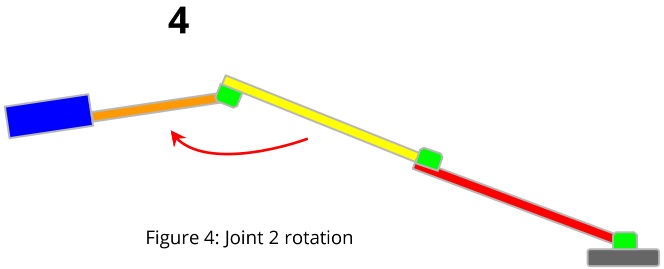


Figure 4: Joint 2 rotation

Baseline Design: **Software Structure**

Baseline Design: Software Structure

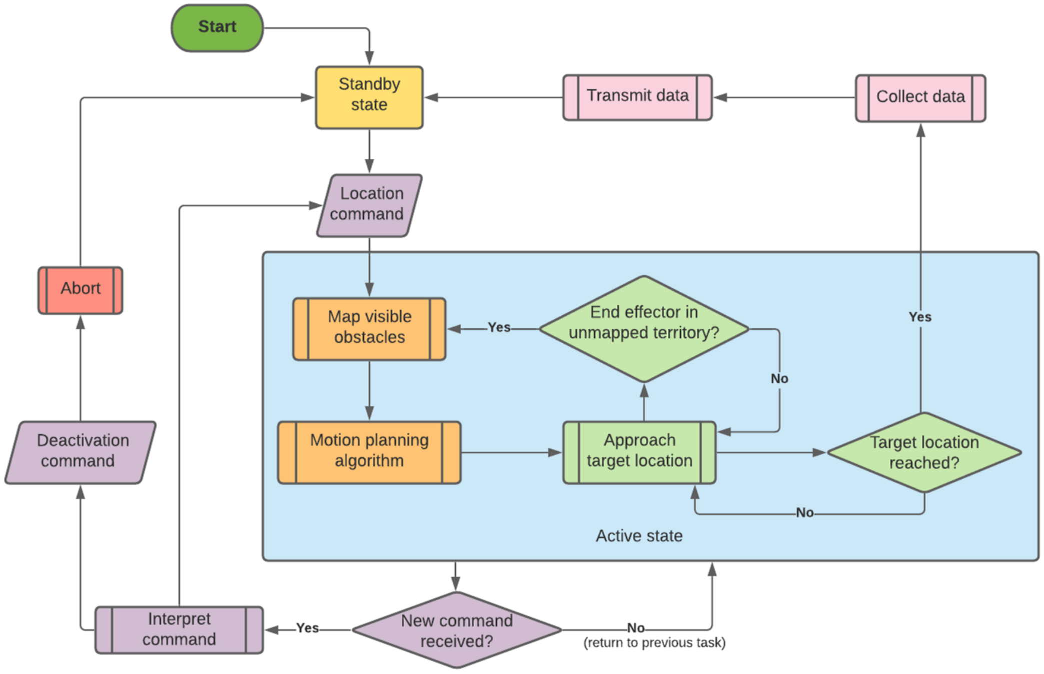
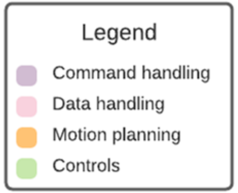


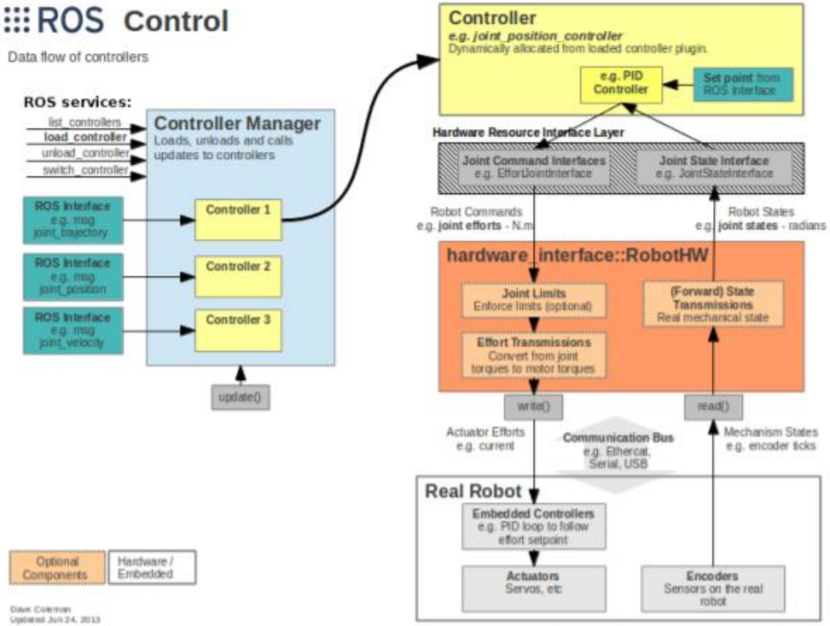
Figure 1: Software simplified flowchart

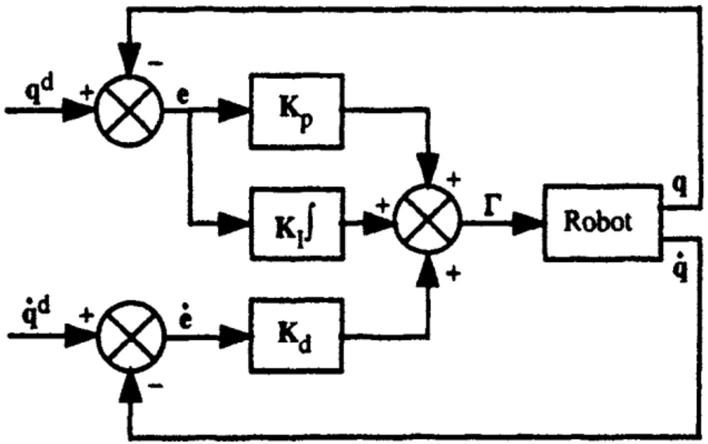
FR3.1	RESCUE shall **determine and report its location** and orientation relative to the ground robot.	System Position and Orientation
FR6.1	RESCUE shall **communicate its sensed data** with MARBLE and this process shall not interfere with MARBLE's communication systems. RESCUE shall be able to receive firing commands, nested firing commands,and deactivation commands from MARBLE's team.	Communication



<u>MME</u> Approach: Mapping ---> Motion Planning ---> Executing

Baseline Design: Attitude Control





$$\Gamma = K_p (q^d - q) + K_d (\dot{q}^d - \dot{q}) + K_I \int_{t0}^{t} (q^d - q) d\tau$$

Figure 1: ros_control node process flowchart

Chitta et al. , *ros_control: A generic and simple control framework for ROS.* 2017

Figure 2: PID controller in the joint space

Khalil, *Modeling, Identification & Control of Robots*. 1999

Baseline Design: Visual Sensor

<u>Intel RealSense D435:</u>
90x25x25mm
Depth Resolution: 1280x720
RGB Resolution: 1920x1080
Diagonal FOV: 70 degrees
Range: 0.3m to 10m

Connection: USB-C 3.1 Gen 1

<u>Decision Drivers:</u>
-The smallest option considered
-Meets the required 5m range
-Compatible with microcontroller
-Easily integrated with MARBLE

FR2.1	The sensing system shall be **able to sense** DARPA subterranean challenge competition artifacts.	Artifact Sensing

Only required to take images, **not** analyze the images or determine what artifacts are in FOV

Figure 1: Intel RealSense D435

Baseline Design: Attitude Determination

FR3.1	RESCUE shall determine and **report its location and orientation** relative to the ground robot.	System Position and Orientation

- Attitude and Heading Reference System (AHRS)
- Orientation resolution: <0.08°
- Accelerometer Sensitivity: ±2g/±4g/±8g selectable
- Gyro Sensitivity:
  0.00833°/sec/digit for ±250°/sec
  0.06667°/sec/digit for ±2000°/sec

- Supply Voltage: +5V USB, +3.3-6.0V for RS232
- Temperature range: -40 - 85°C
- Connection: USB 2.0 or RS232

Figure 1: YOST 3-Space™ Watertight USB/RS232

Baseline Design: Connections

FR6.1	RESCUE shall **communicate its sensed data** with MARBLE and this process shall not interfere with MARBLE's communication systems. RESCUE shall be able to receive firing commands, nested firing commands, and deactivation commands from MARBLE's team.	Communication

Wiring Connections

- Gigabit Ethernet from MARBLE to Raspberry Pi (RasPi)
- USB-A 3.0 to USB-C 3.0 from RasPi to RealSense camera
    - Continuous-flex-and-twist cable.
- USB 2.0 Connection to YOST AHRS
- The additional lighting and C02 sensor will be connected using a direct serial connections and a EK-Cable Y-Splitter

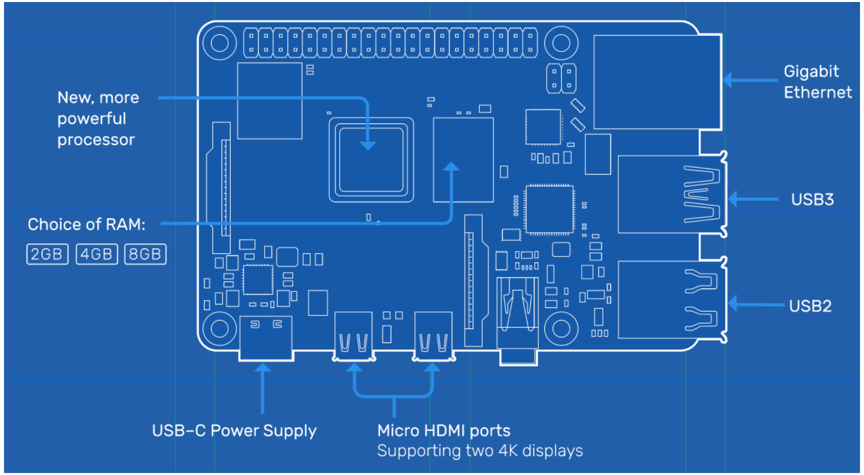


Figure 1: Raspberry Pi 4 Tech Sheet

Feasibility Analysis

Critical Project Elements

- Arm Design
- Wrist Design
- Joint Design

Mechanical

- Visual Sensor
- Attitude determination
- Attitude controls

Software

In the feasibility we will talk about how it is relative to MME (Mapping, Motion planning, and Execution) and not to Sensing. Sensing feasibility is available in Backup slides.

Mechanical Systems Feasibility Analysis

Arm Feasibility, Joint Feasibility

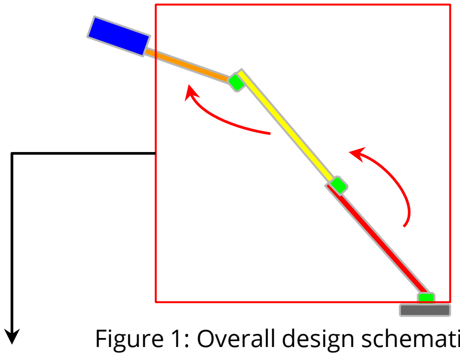


Figure 1: Overall design schematic

Designed for “worst case scenario” - full horizontal extension

- Static analysis of the arm performed
  - Base vertical reaction force ≈ 11.3 N
  - Base reaction moment ≈ 118.6 kg-cm

- Torque per joint
  - 1st joint torque ≈ 72.3 kg-cm
  - 2nd joint torque ≈ 26.1 kg-cm

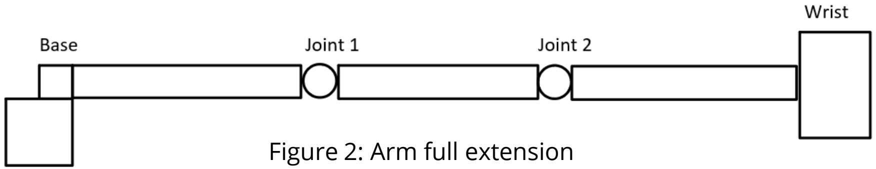


Figure 2: Arm full extension

- Arm member structure - carbon fiber tubing
  - 4.76 cm OD, 4.44 cm ID (1.875 inch and 1.75 inch)
  - Very lightweight - 22 grams/meter
  - Flexural strength - 0.61 GPa
  - Maximum deflection of single member - 6.73 * 10^{-6} meter

Servo Motor

- Most servos can support the wrist of the arm and the joints
- The base of the arm may end up with a different configuration due to its torque requirements

Figure 1: JX Servo C70 HV

Possible Servo Choices

More on: Servos Stepper Motors

	Weight [g]	Stall Torque [kg-cm]	Size [㎣]	Max Power [W]
ZOSKAY Digital Servo	60	35	40x20x38.5	17
CYS S0650	202	55	66x30x56	99.2
JX Servo C70 HV	87	72	40x20x43.3	73.2
TAL9150	580	158	96.4x44x85	80

Arm "Wrist" Feasibility

Feasibility Items

- Can lightweight servos supply adequate torque to rotate the sensor package?
- Approximate "wrist" mass?
- 90∘ sensor package rotation about at least one axis (TDR1.1.3)

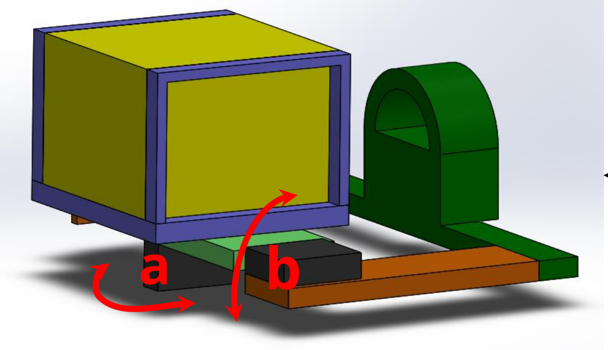


Figure 2: Wrist Design Estimate

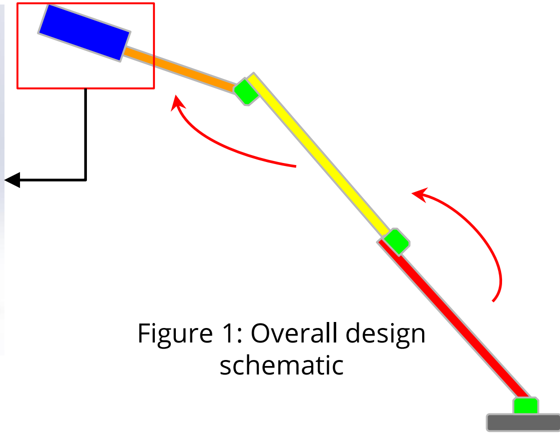


Figure 1: Overall design schematic

Baseline Assumptions

- π rad/s^2 acceleration
- Sensor package: D435i, SCD30, Yost Labs 3-Space AHRS, 2 Streamlight MicroStream lights **(worst case)**
- Solid PLA (1.24 g/cm^3) mounting components **(worst case)**

Required Torque Estimates

Servo	a	b
Required Torque [kg-cm]	**0.031**	**0.12**

Wrist Mass Estimate:
≈ 830 g

Arm "Wrist" Feasibility: Servo Selection

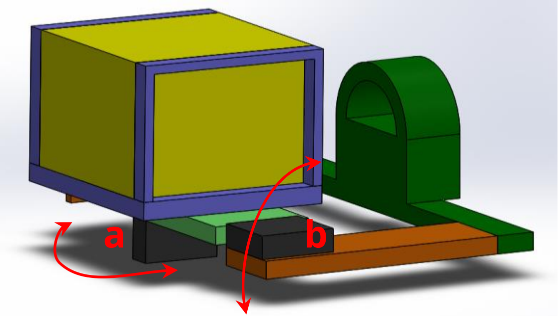


Figure 1: Wirst Design Estimate

Required Torque Estimates

Servo	a	b
Required Torque [kg-cm]	**0.031**	**0.12**

Possible Servo Solutions:

Possible Servo	**Max Torque [kg-cm]**	**Speed [sec/60°]**	**size [㎣]**	**Mass [g]**	**Cost [$]**
Hitec HS-85MG	**0.547**	0.14	29x13x30	21.9	29.95
Sparkfun Sub Micro	**0.232**	0.10	31.8x11.7x29	9	8.95

Joints Feasibility

- Multiple simple existing design options
- Do not require gearing
- Minimal wire interference
- Example cases fit similar mission profiles

Fig. 4 Cylindrical Sections, Single Servo External Joint

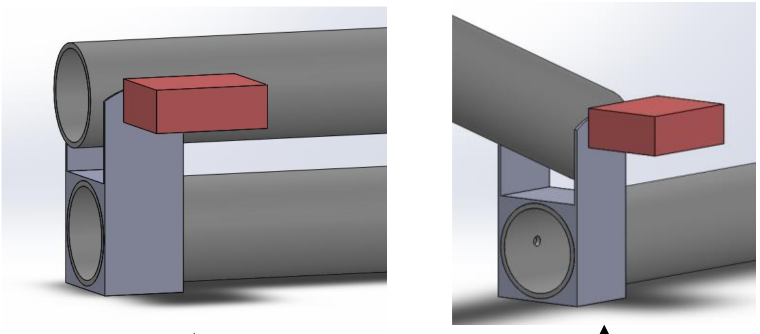


Fig 2. Dual Servo, External Joint: (Trossen Robotics WidowX 250)

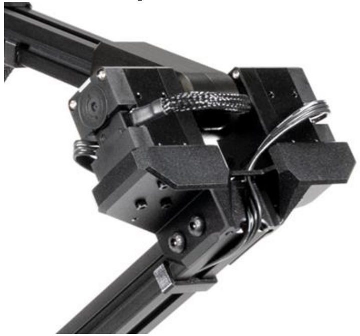


Fig. 3 Single Servo (Servosila/Trossen)

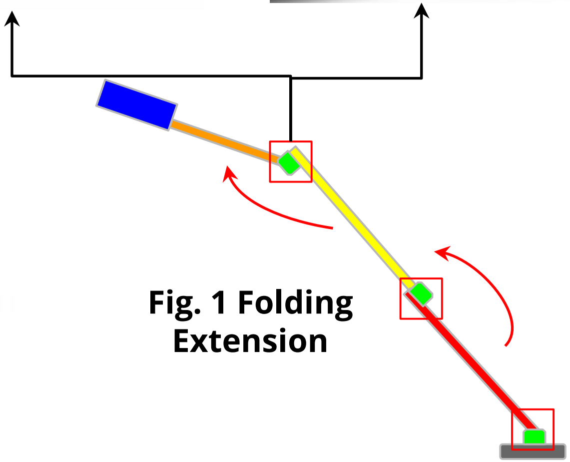


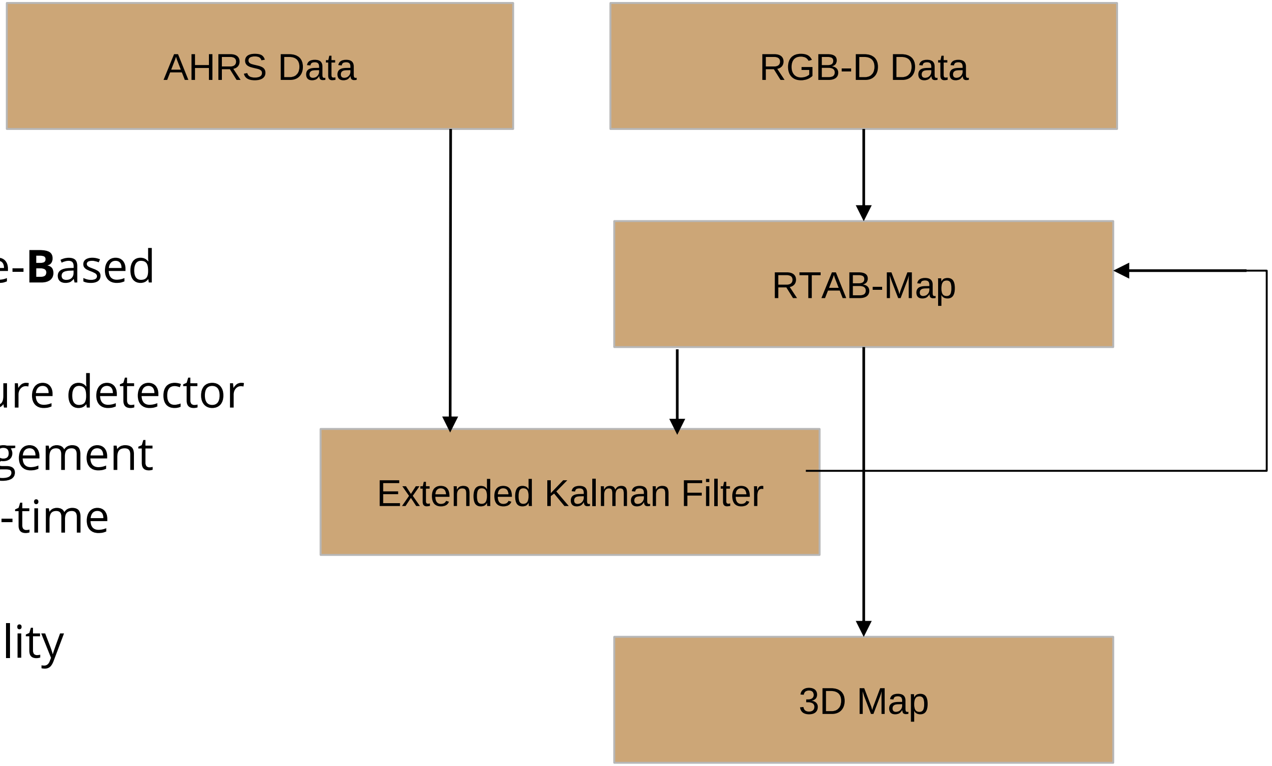
Fig. 1 Folding Extension

Software Systems Feasibility Analysis

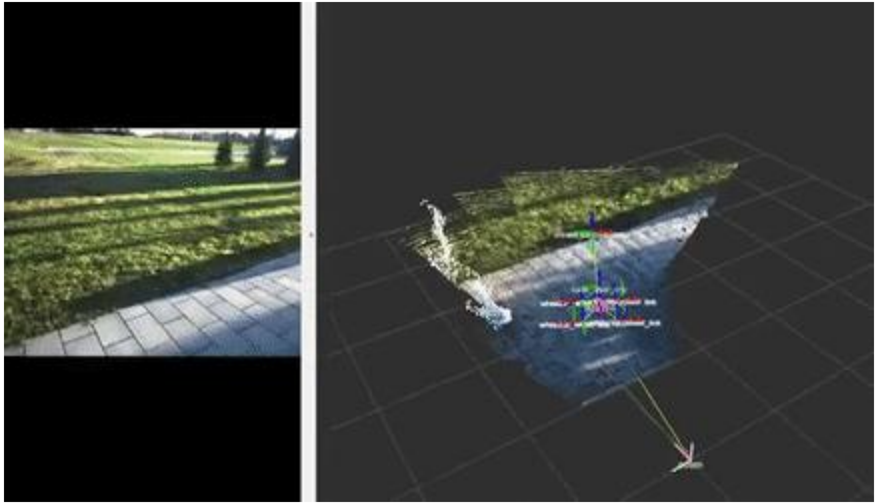
Obstacle Mapping Feasibility

RTAB-Map

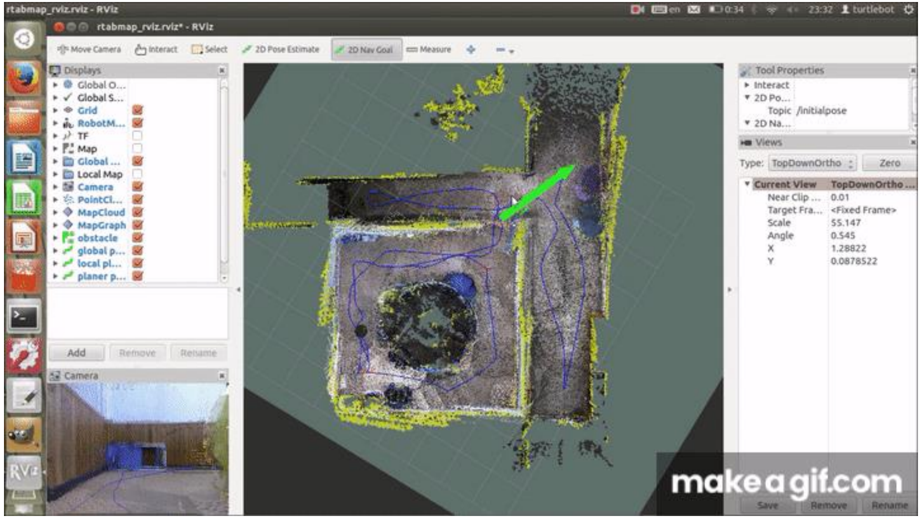
- **R**eal-**T**ime **A**ppearance-**B**ased **Map**ping
- Incremental loop closure detector
- Uses a memory management approach to meet real-time constraints
- Built-in ROS compatibility



Obstacle Mapping Feasibility (Example)



GIF 1: RTabMap being used outdoors



GIF 2: Rapid development using the RTabMap GUI

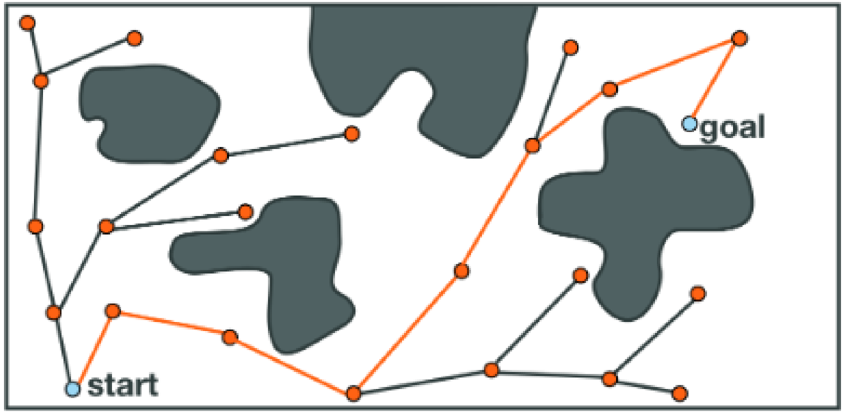
Motion Planning Feasibility (Implementation)

Open Motion Planning Library (OMPL)

- Sampling-based motion planning
- Includes several MP algorithms
- Based in C++
- Built-in ROS compatibility

Alternative: C++ Robotics Library

- Multi-purpose C++ library
- Motion planning, inverse kinematics, hardware abstraction
- No built-in ROS compatibility



OMPL sampling-based planner using a tree structure to generate a path from starting state to target state (courtesy of Rice University Kavraki Lab)

Figure 1: Example of a motion planning algorithm in the OMPL Library

Motion Planning Feasibility (Simulation, 2D).

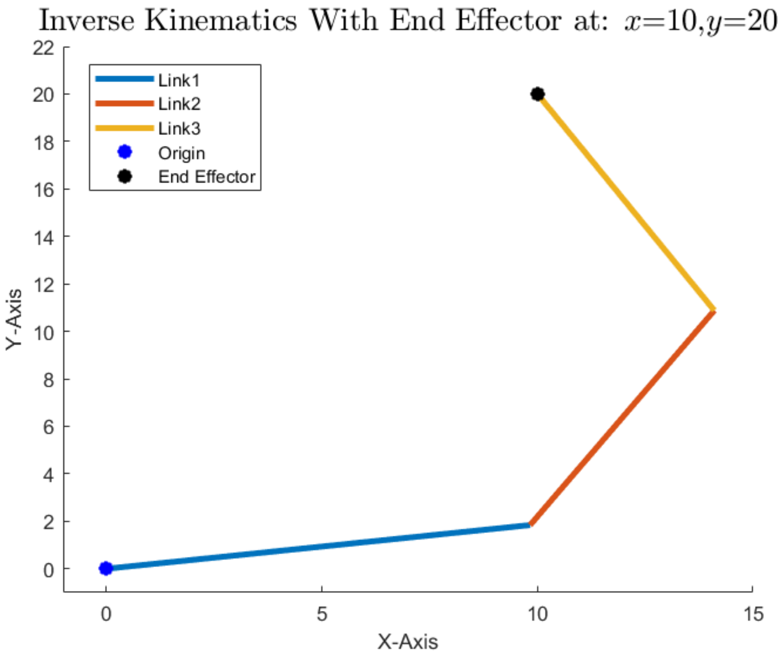


Figure 1: 3 link simulation of inverse kinematics

Inverse Kinematics:

(1) $$x = a_1 \cos\theta_1 + a_2 \cos(\theta_1 + \theta_2)$$

(2) $$y = a_1 \sin\theta_1 + a_2 \sin(\theta_1 + \theta_2)$$

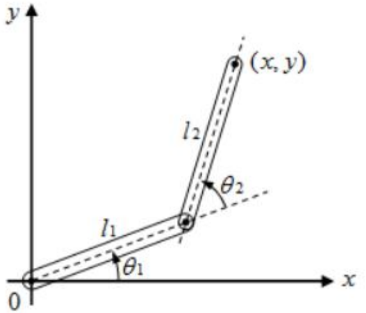


Figure 2: 2 link manipulator

$$\boxed{\cos\theta_2 = \frac{1}{2a_1a_2}\left((x^2 + y^2) - (a_1^2 + a_2^2)\right)}$$

$$\boxed{\sin\theta_2 = \pm\sqrt{1 - \cos^2\theta_2}}$$

$$\boxed{\cos\theta_1 = \frac{1}{x^2 + y^2}\left(x\,(a_1 + a_2\cos\theta_2) \pm y\,a_2\sqrt{1 - \cos^2\theta_2}\right)}$$

$$\boxed{\sin\theta_1 = \frac{1}{x^2 + y^2}\left(y\,(a_1 + a_2\cos\theta_2) \mp x\,a_2\sqrt{1 - \cos^2\theta_2}\right)}$$

Formulas from: Morteza Lahijanian (5519 Lec 5)

Motion Planning (3D)

MATHWORKS Robotics System Toolbox

Rapidly design, simulate, and test robotics applications

Motion Planning (3D)

Two methods:

Task space
Vs
Joint Space

Discussed in details in backup slides.

Motion Planning (3D):

For the Kinova Gen3, a complex robotic arm with features like RESCUE, MATLAB simulations are feasible.

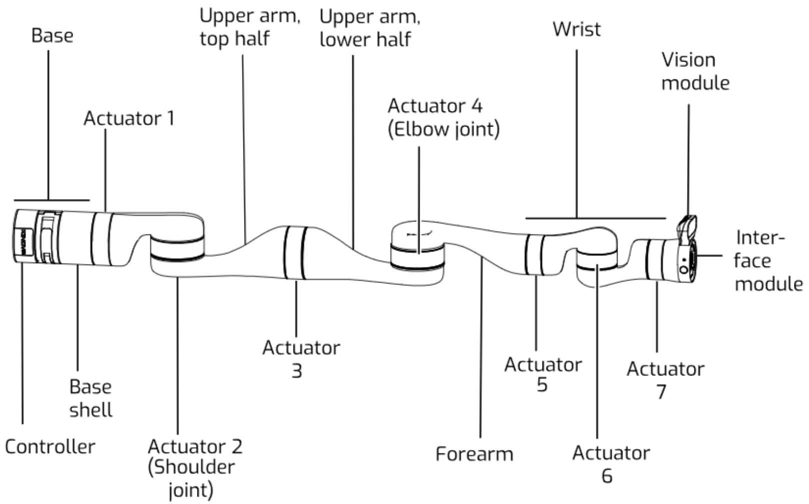
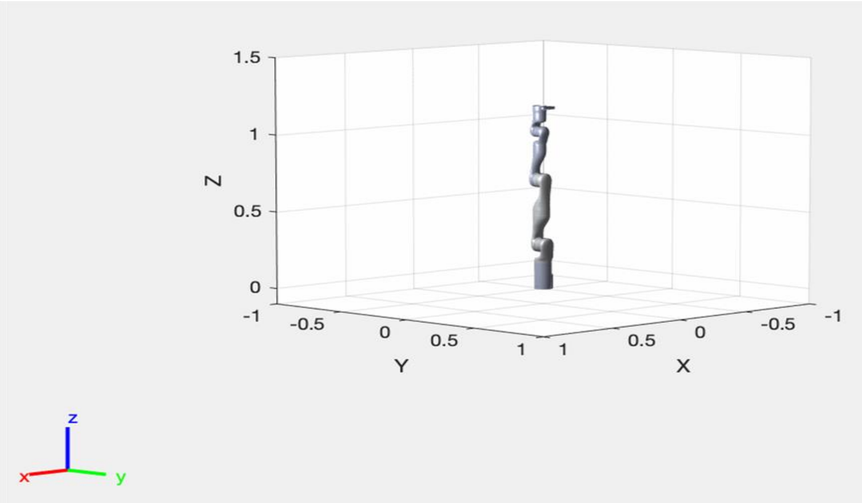


Figure 1: Schematic of Kinova Gen 3

Figure 2: Kinova Gen 3 robotic arm

Example 1:

Motion Planning Feasibility (Simulation, 3D, PD control, Task space).



GIF 1: 3D simulation of robotic arm moving in Task space

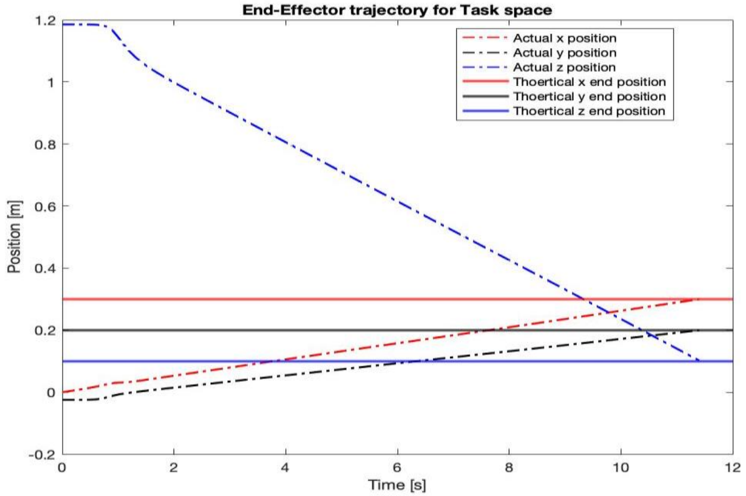
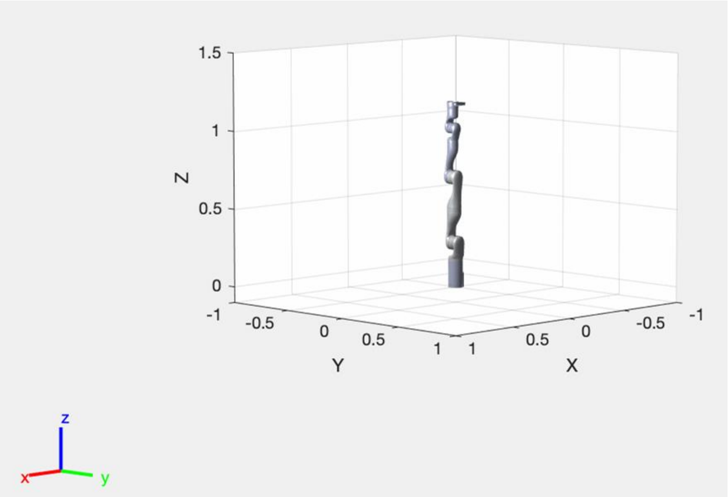


Figure 1: End effector position in Task space

Motion Planning Feasibility (Simulation, 3D, PD control, Joint Space).



GIF 1: 3D simulation of robotic arm moving in Joint space

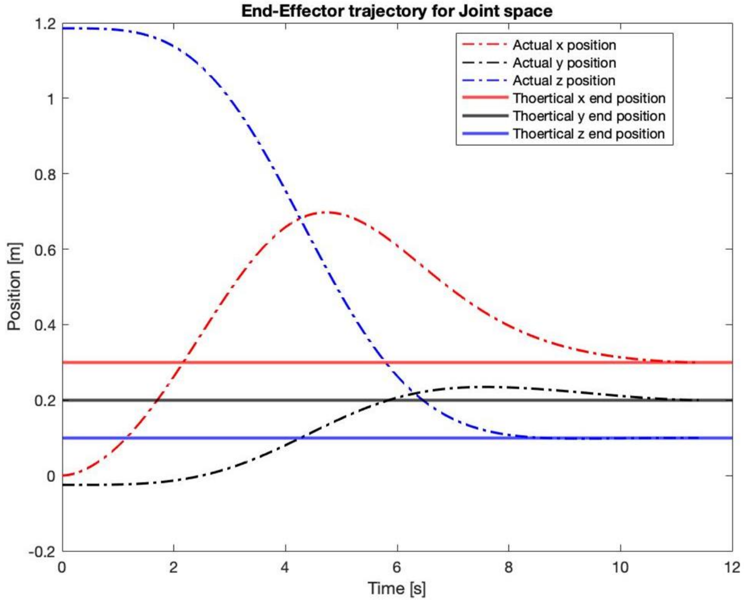


Figure 1: End effector position in Joint space

Example 2:
Comparing Task space vs Joint space

[Workspace vs. C-Space]

Motion Planning Feasibility (Simulation, 3D, Joint Space Vs Task Space).

Suppose we want our end effector to follow the following trajectory:

By how much should we rotate each joint at a given time?

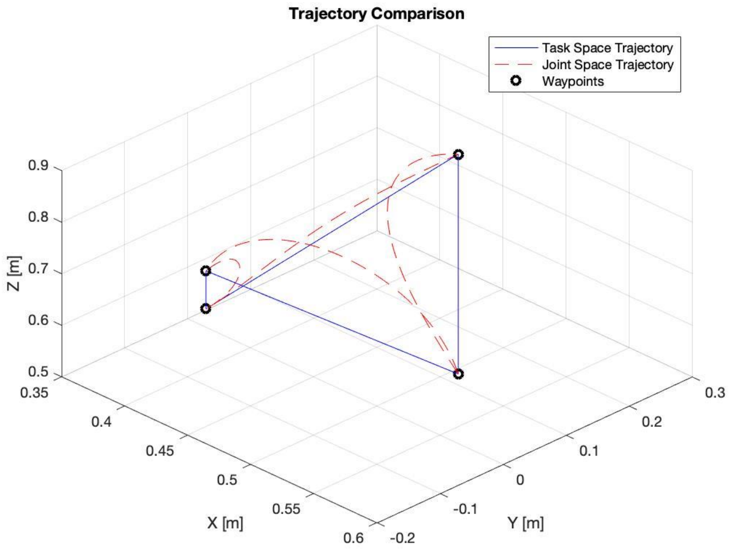
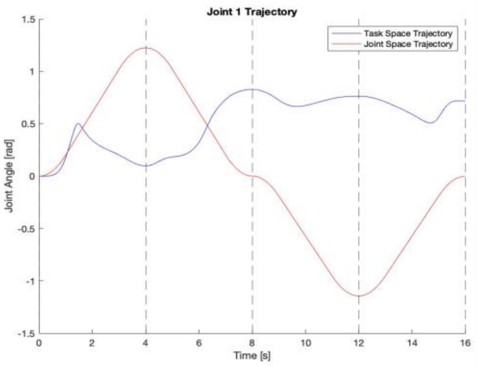
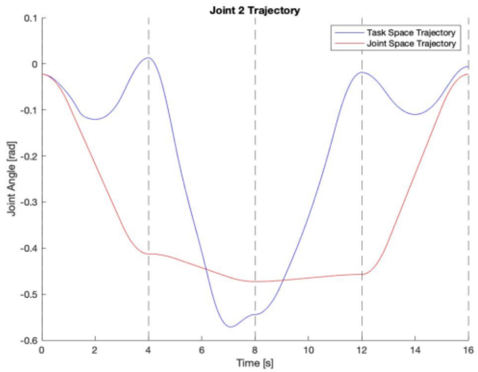
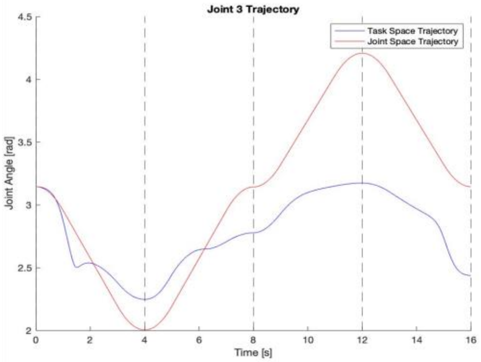


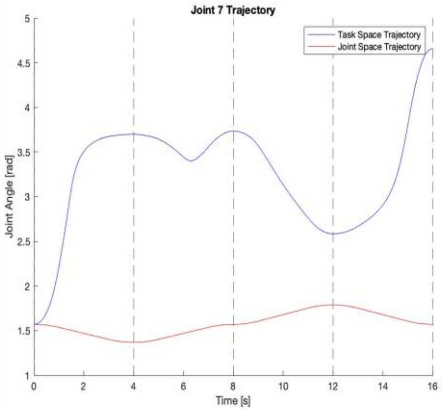
Figure 1: Hypothetical trajectory in 3D space

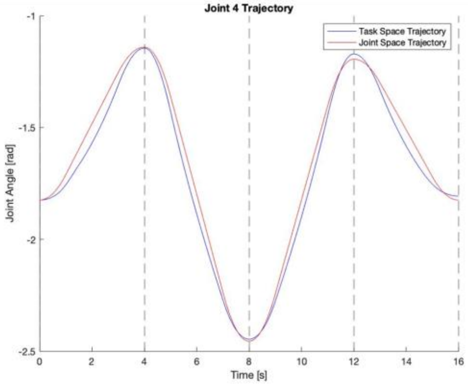
Motion Planning Feasibility (Simulation, 3D, Joint Space Vs Task Space).

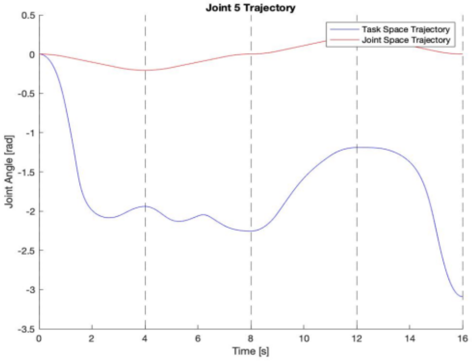


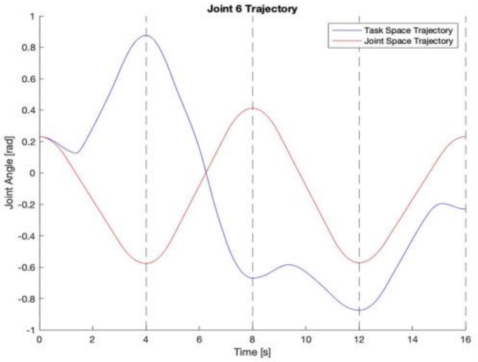








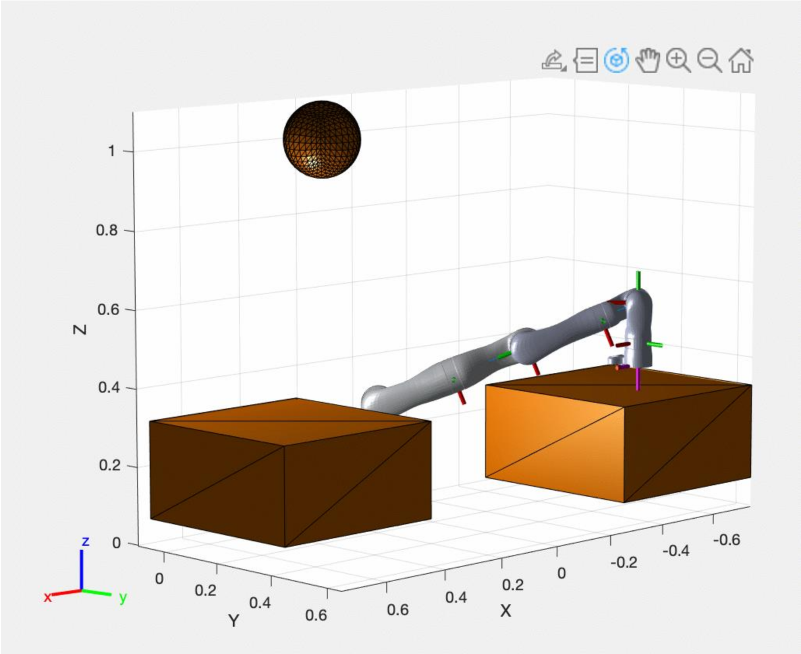




Figures 1-7: Joints rotations in radians at given time

Example 3:
Collision-free Trajectories
Given a map of the obstacles

Motion Planning Feasibility (Simulation, 3D, Collision-free, MATLAB).



Gif 1: 3D simulation of robotic arm moving in collision-free trajectory

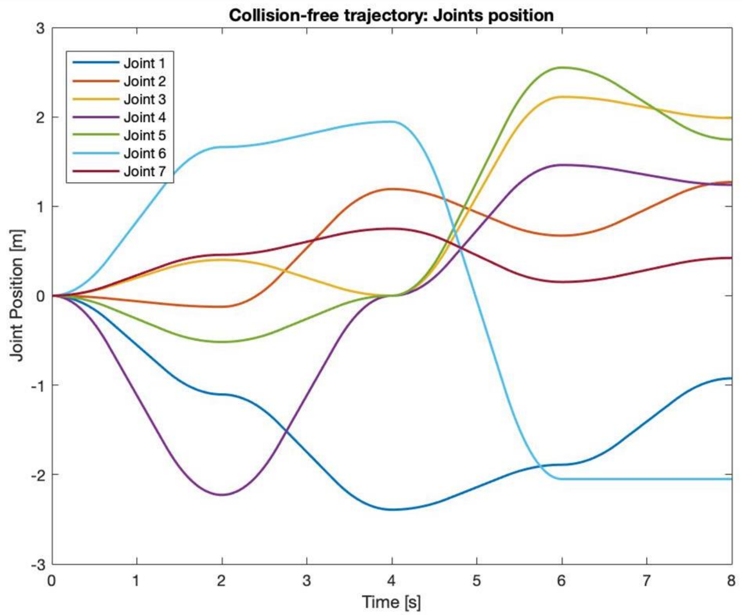


Figure 1: Joints location as a function of time for collision-free trajectory

Controls Feasibility

<u>ROS MoveIt Library</u>:

- Robotic arm controls library
- Python, C++ interface
- Utilizes OMPL for motion planning
- Plans trajectories for individual joints
- Handles inverse kinematics, 3D perception pipeline, collision checking
- Includes 3D interactive visualizer

<u>Options considered</u>: MoveIt, Simulink, MATLAB Robotics and Control Toolboxes

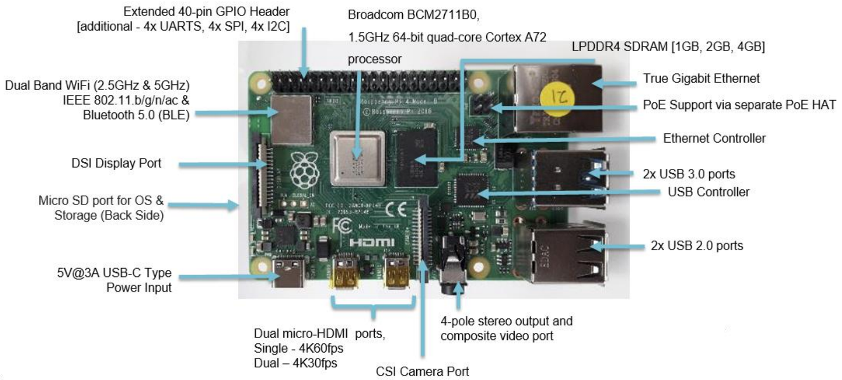


Figure 1: Raspberry Pi 4 input breakdown

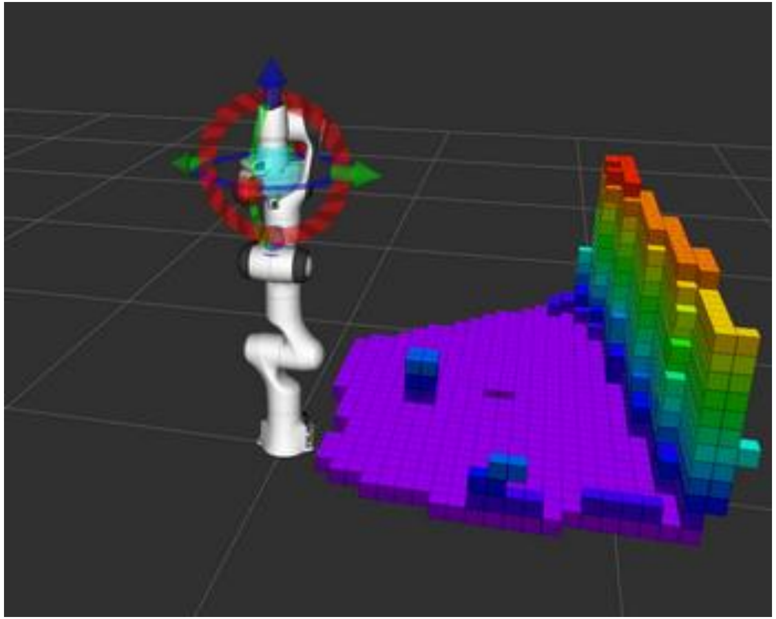
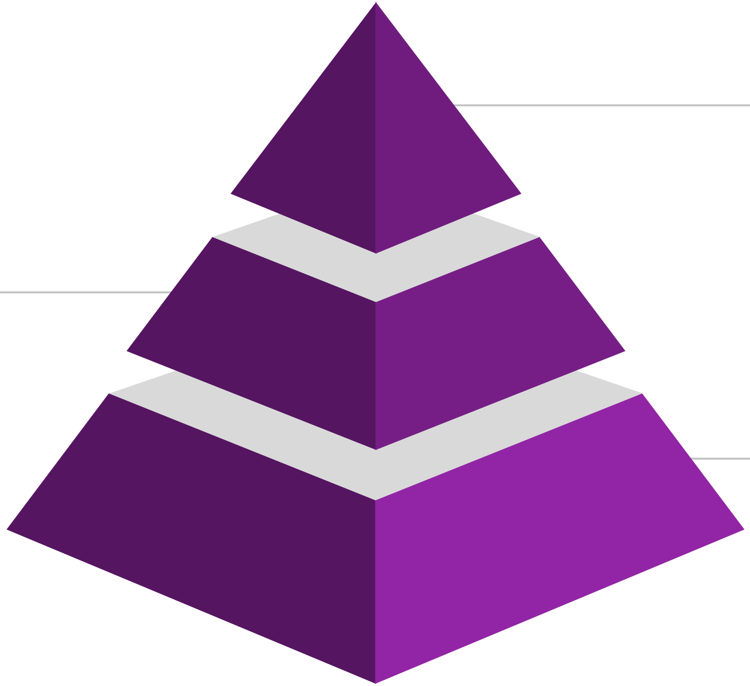


Figure 2: Perception pipeline visualization of point-cloud data (courtesy of MoveIt)

Summary: Subsystems overview



Communications

1 Communications will serve as the link between MARBLE’s operator and RESCUE. In addition, it will connect the Software structure with the mechanical one.

Software structure

2 Software deals with Mapping and Motion Planning. Through the software structure the mechanical joints and arms will be controlled and the sensors will be activated and its data will be used if necessary.

Mechanical structure

3 Capable of providing structural integrity needed to house the sensors and move RESCUE.

***Subsystems detailed breakdown in backup slides.**

Summary: Feasibility

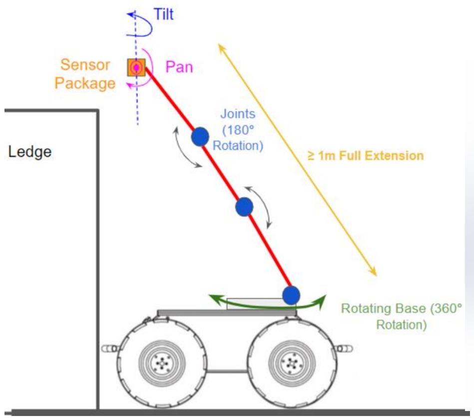


Figure 1: Side view Deployed arm

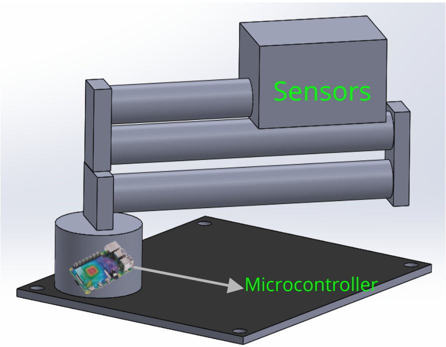


Figure 2: Side view stowed arm

Design aspect	Feasibility
Arm Design	Feasible; Structural integrity and torques needed could be achieved.
Wrist Design	Feasible; Structural integrity and torques needed could be achieved.
Joint Design	Feasible; Required torques could be achieved. Additional studies required for joint volume constraints.
Visual Sensor	Feasible; Sensor available off-the-shelf and compatible with microcontroller.
Attitude determination	Feasible; Sensor available off-the-shelf and compatible with microcontroller.
Attitude controls	Feasible; Open source libraries and available toolboxes.

Summary: Future studies

Mechanical:

- Finalizing joint design/volume fitting
- FEA simulation for stress and torsion
- Von Mises/Mohr-Coulomb stress analysis
- Lighting solution mass reduction (see backup slide)

Software:

- Choose between task space or joint space
- Build a ROS node map to aid in development

References

Mechanical Systems

- https://www.streamlight.com/products/detail/index/microstream
- https://en.wikipedia.org/wiki/Polylactic_acid
- https://yostlabs.com/product/3-space-watertight-usbrs232/
- https://www.sparkfun.com/products/11887
- https://www.sparkfun.com/products/9065
- https://clearpathrobotics.com/husky-unmanned-ground-vehicle-robot/
- https://www.instructables.com/Complete-Motor-Guide-for-Robotics/
- Spong, Mark W., Seth Hutchinson, and Mathukumalli Vidyasagar. *Robot Modeling and Control*. Hoboken, NJ: Wiley, 2006.
- https://www.servosila.com/en/robotic-arms/index.shtml
- https://www.trossenrobotics.com/widowx-250-mobile-robot-arm-6dof.aspx
- https://mechanicalc.com/reference/beam-analysis
- https://www.mcmaster.com/5287T25-5287T253/
- https://www.mcmaster.com/hinges/t-slotted-framing-pivots/
- https://www.universal-robots.com/plus/urplus-components/accessories/bn-uni-kit-for-ur5-ur10/
- https://simxperience.com/Community/SimXperienceDevelopersBlog/TabId/783/ArtMID/1674/ArticleID/13/Direct-Drive-vs-Belt-Drive-vs-Gear-Drive.aspx
- https://www.servocity.com/servo-gearboxes/
- https://en.wikipedia.org/wiki/List_of_moments_of_inertia

References (cont.)

Sensing/Electronic Systems

- https://www.raspberrypi.org/products/raspberry-pi-4-model-b/specifications/
- https://www.newnex.com/usb-3-high-flex-cables.php
- https://yostlabs.com/product/3-space-watertight-usbrs232/
- https://github.com/sparkfun/SparkFun_SCD30_Arduino_Library
- https://www.sensirion.com/en/environmental-sensors/evaluation-kit-sek-environmental-sensing/

Software Systems

- https://introlab.github.io/rtabmap/
- Ioan A. Șucan, Mark Moll, Lydia E. Kavraki, The Open Motion Planning Library, *IEEE Robotics & Automation Magazine*, 19(4):72–82, December 2012. https://ompl.kavrakilab.org
- https://moveit.ros.org/
- https://www.mathworks.com/products/robotics.html

Questions?

Backup Slides

Functional Requirements	Kalman Filter Overview	Mechanical Joints	PD Control
Technical Design Requirements	Numerical Inverse Kinematics	Cable Routing	Microcontroller
Detailed Software Flowchart	Budget Tracking	Sensor Package Fit	AHRS Sensor Feasibility
Workspace vs. C-Space	DARPA Subterranean Challenge	Base Design	Communication Feasibility
CO_2 Sensor	MARBLE	Tipping Analysis	Interfacing w/ MARBLE
RGB-D Output Filtering	Attitude Determination	Motor Design	Subsystems Breakdown
Camera Lighting	Arm Calculations	Water/Dust Proofing	Embedded System
Power Budget Tracking	Wrist Calculations	Task Space vs. Joint Space	

Backup Slide: Functional Requirements

FR1.1	At minimum, RESCUE shall have the ability to physically reach a location that is at least 1 meter but not more than 5 meters from the apparatus' mounted position on the MARBLE Clearpath Husky.
FR2.1	RESCUE shall be able to sense DARPA subterranean challenge competition artifacts.
FR3.1	RESCUE shall determine and report its location and orientation relative to the ground robot.
FR4.1	When in its standby configuration, RESCUE shall be compatible with the MARBLE team's Clearpath Husky.
FR4.2	When in operation, RESCUE shall not interfere with the MARBLE team's Clearpath Husky's operations.
FR4.3	RESCUE's deployment operations shall be rapid enough to incur a minimal time cost to MARBLE's total mission time.

Back

Backup Slide: Functional Requirements

FR5.1	RESCUE despite the splash exposure to mud, water, and dust expected in the DARPA Subterranean Challenge circuit environment. RESCUE shall withstand the thermal environment of the DARPA subterranean challenge.
FR5.2	RESCUE shall have enough electrical power to maintain standby, active, and operational states fitting the MARBLE team's mission expectations.
FR5.3	RESCUE shall withstand repeated deployments.
FR6.1	RESCUE shall communicate its sensed data with MARBLE and this process shall not interfere with MARBLE's communication systems. RESCUE shall be able to receive firing commands, nested firing commands, and deactivation commands from MARBLE's team.

Back

Backup Slide: Technical Design Requirements

FR1.1	TDR1.1.1	RESCUE shall have the ability to physically reach a location that is at least 1 meter but not more than 5 meters along any unobstructed direction from the mounted position of the apparatus of the MARBLE ClearPath Husky.
	TDR1.1.2	RESCUE shall have the ability to make at least one directional change of 45 or more about at least one axis from the starting extended location to physically reach a location that is at least 1 meter but not more than 5 meters and not within a clear line of sight from the mounted position of the apparatus on the MARBLE Clearpath Husky.
	TDR1.1.3	Once RESCUE is re-positioned, the mechanical mount for the visual artifact signature sensor shall be capable of rotating at least 90 or more about at least one axis.
FR2.1	TDR2.1.1	The sensing apparatus shall have the capability to visually sense the following brightly colored artifacts: human survivor, backpack, fire extinguisher, and rope. The visual sensing of these artifacts shall occur within the visual sensor's operational field of view.

Back

Backup Slide: Technical Design Requirements

	TDR2.1.2	The sensing apparatus shall be able to sense and detect carbon dioxide (CO_2) at 2000 parts per million concentration.
FR3.1	TDR3.1.1	RESCUE shall be able to determine its position relative to the ClearPath Husky, within 1 meter accuracy of its ground truth location at all times.
	TDR3.1.2	RESCUE shall be able to determine its orientation to the ClearPath Husky within 5 accuracy of its ground truth orientation at all times.
FR4.1	TDR4.1.1	When in its standby configuration, RESCUE shall not exceed a volume of 38 centimeters wide by 45 centimeters long by 22-30 centimeters tall.
	TDR4.1.2	RESCUE shall not exceed a total mass of 10 kilograms.
	TDR4.1.3	If RESCUE is directly connected to the Husky, power drawn from the Husky robot shall be less than or equal to 24-30 Volts at 25 Amps.

Back

Backup Slide: Technical Design Requirements

FR4.2	TDR4.2.1	When RESCUE is deploying, in its active state, or in its operational state, the sensing apparatus shall not apply a force or moment that can unintentionally alter the position of or damage the MARBLE Clearpath Husky.
FR4.3	TDR4.3.1	Upon receiving an activation command from the MARBLE team when in standby configuration, RESCUE shall reach an active state in 30 seconds or less.
	TDR4.3.2	Upon receiving a firing command from the MARBLE team when in its active configuration, RESCUE shall respond in an operational state as soon as (< 1 second) the command is received.
	TDR4.3.3	Upon receiving an deactivation command from the MARBLE team while, RESCUE shall return from its operational/active configuration to its standby configuration within 120 seconds.

Back

Backup Slide: Technical Design Requirements

FR5.1	TDR5.1.1	RESCUE's mechanical and electrical components shall meet at least IP43 water exposure tolerances.
	TDR5.1.2	RESCUE's mechanical and electrical components shall meet at least IP43 dust exposure tolerances.
	TDR5.1.3	RESCUE shall accomplish all other design requirements in an nominal thermal environment of 50-65 F.
FR5.2	TDR5.2.1	RESCUE shall have enough electrical power to maintain a standby state for at least 135 minutes.
	TDR5.2.2	RESCUE shall have enough electrical power to maintain an operational state for at least 30 minutes.

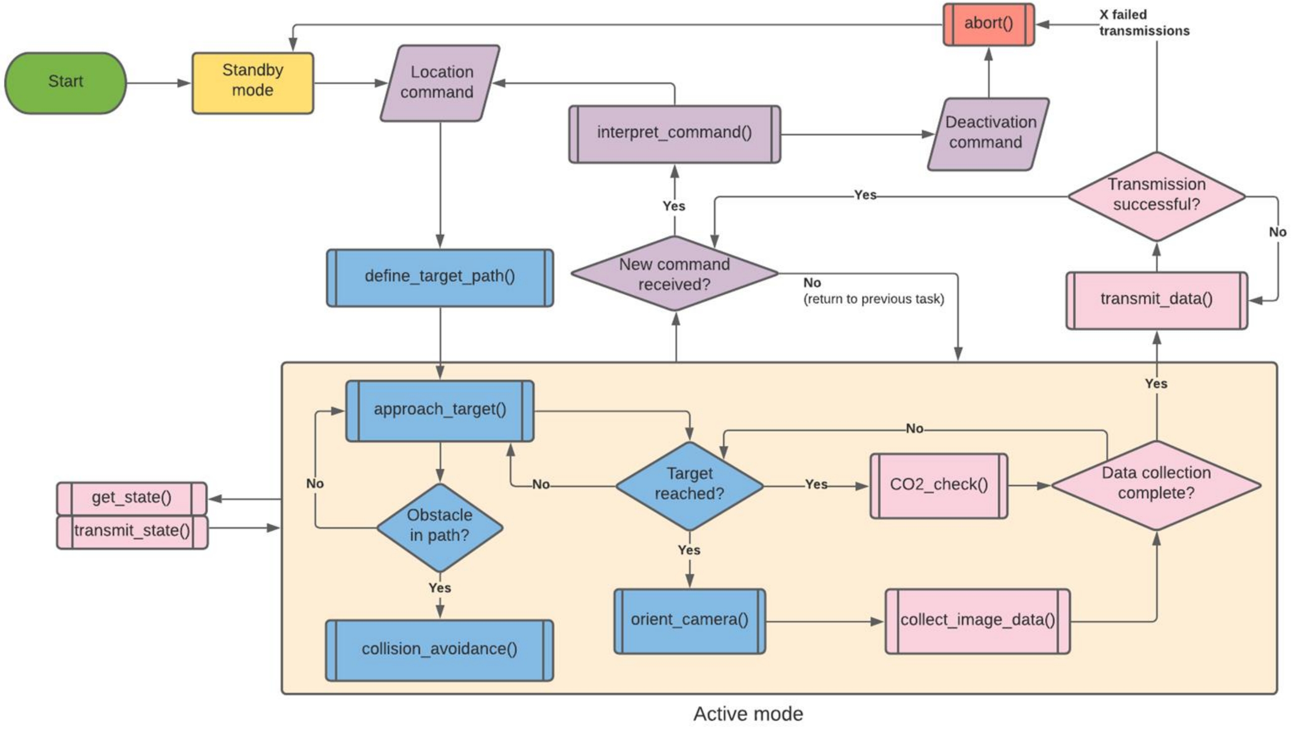
Back

Backup Slide: Technical Design Requirements

FR5.3	TDR5.3.1	
FR6.1	TDR6.1.1	The sensing system shall be capable of receiving firing commands from the ROS nodes in the existing MARBLE architecture.
	TDR6.1.2	After deployment and retraction, RESCUE shall communicate sensing data with the MARBLE robot before its next deployment, or within approximately 60 seconds.
	TDR6.1.4	RESCUE shall deliver frequent status reports to the MARBLE robot regarding deployment status and data collection.

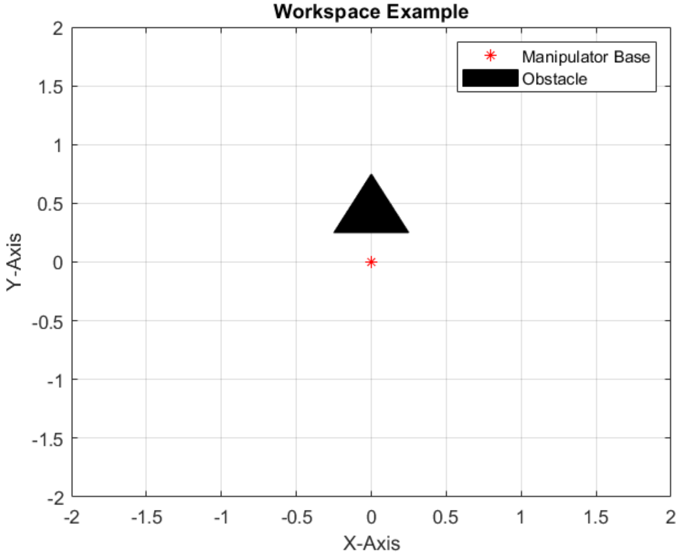
Back

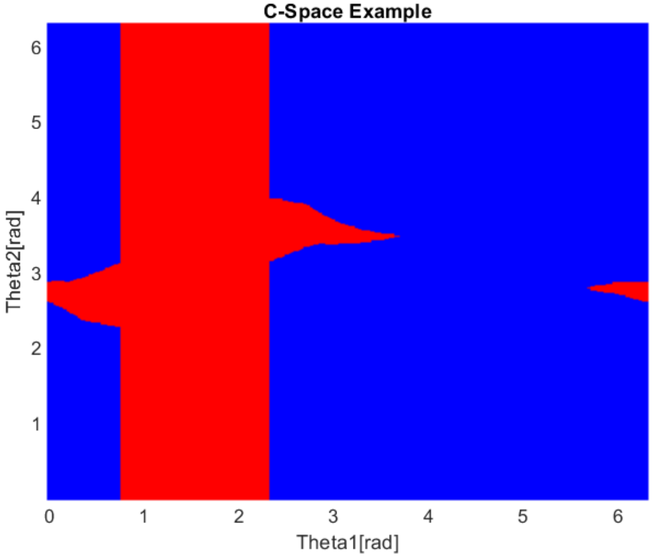
Backup Slide: Detailed Software Flowchart



Back

Backup Slide: Workspace Vs. C-Space





In the C-Space the two link manipulator becomes a point robot. This makes it much easier to apply planning algorithms.

Back to Example 2 Back

Backup Slide: CO_2 Sensor

FR2.1: The sensing system shall be able to sense DARPA subterranean challenge competition artifacts.

Options Considered: Sensirion SCD30 and SparkFun CCS811 Air Quality Breakout

Sensirion SCD30:
35x23x7 mm
Sensing: >440 PPM
Accuracy: ±(30 PPM + 3%)
Power Required: 400 mW
Cost: $60.00

Decision Drivers:

1. Tested accuracy
2. No startup time
3. Proven integration with Raspberry Pi

Back

Backup Slide: RGB-D Output Filtering

ROS Filtering Packages:
- depth_image_proc
- pcl_ros

Intel Realsense Filtering Process:
1. Decimation
2. Edge-preserving filter
3. Spatial filter
4. Temporal filter
5. Hole filling

Back

Backup Slide: Currently Worst Case Lighting

- 2X Streamlight MicroStream USB
- 1.50 hr life on high
- Batteries contained
- 250 Lumens Max
- 34g mass
- 68m beam

Backup Slide: Camera Lighting

Bright white and IR camera light:

- Built specifically to interface with the Raspberry Pi via I2C inputs
- Cree white light LED
- Liteon IR LEDs
- Cost: **$23**

Back

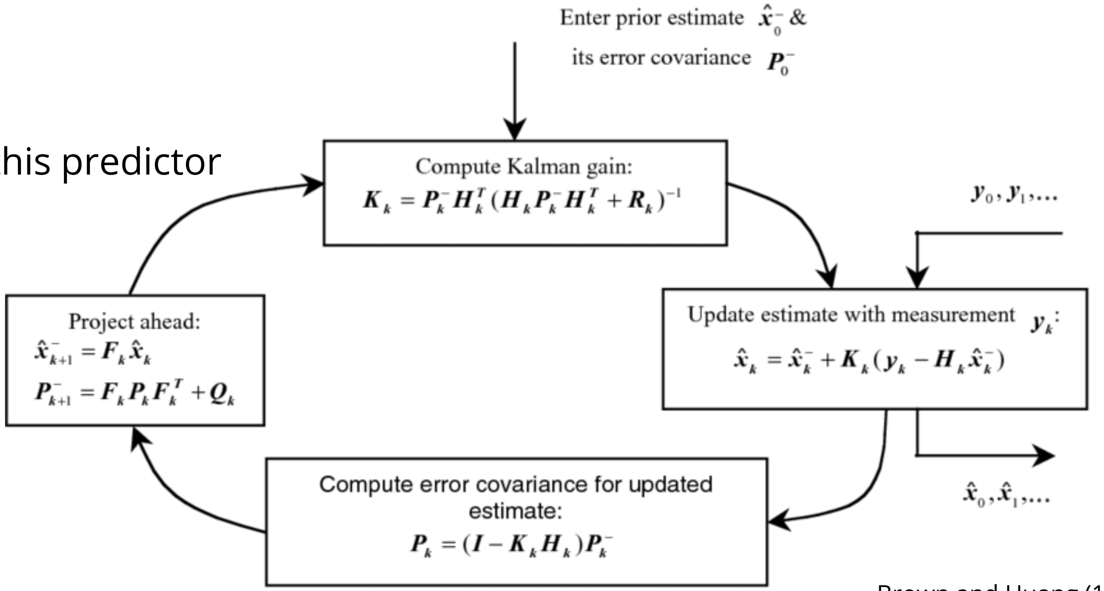
Backup Slide: Power Budget Tracking

Part Name	Devices per Board	Supply Current per Device (A)	Supply Current per Board (A)	Supply Voltage	Power Subtotal (W)
Raspberri Pi 4 Model B	1	3	3	5	15
Sensirion SCD30	1	0.019	0.019	5.5	0.1045
Intel RealSense Depth Camera D435	1	1.6	1.6	3.3	5.28
JX Servo C70 HV 12V 72KG	1	6.1	6.1	12	73.2
100kg-cm torque Ultra Heavy Duty Giant Scale Digital HV Brushless Servo	2	6.6	13.2	12	158.4
CYS S0650 Large 55KG HV High Torque Metal Gear Digital Servo	1	6.9	6.9	7.4	51.06
				Power Total (W)	303.0445
				Est. Efficiency	50%
*Take max values				Input Power Needed (W)	606.089
*From MARBLE: raw power is 24-30 V at 25A				Power Burn (W)	

Backup Slide: Kalman Filter Overview

The point of a Kalman Filter is to obtain an estimate of the current state based on noisy measurements.

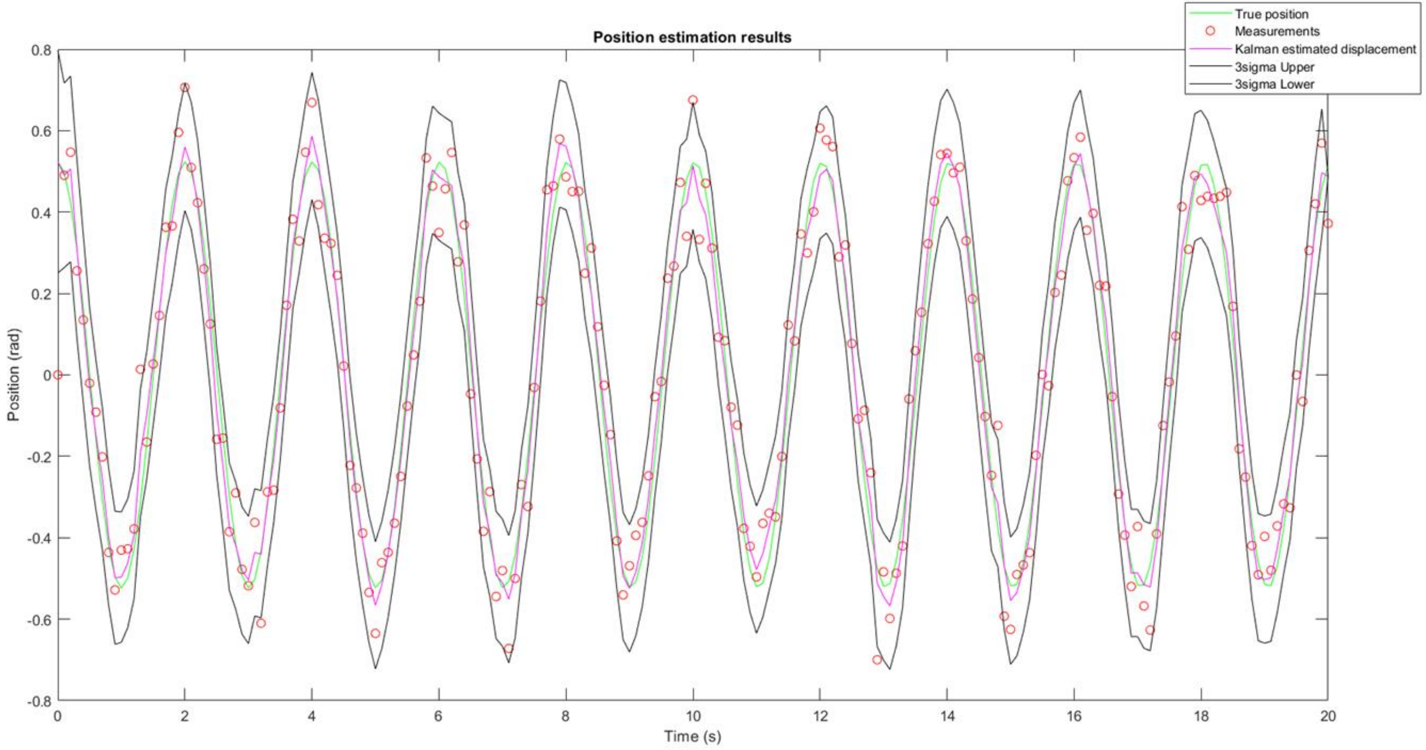
The process follows this predictor corrector loop:



Brown and Huang (1997)

Back

Backup Slide: Kalman Filter Overview (Example)



Back

Backup Slide: Numerical Inverse Kinematics

With a 3 link arm inverse kinematics cannot be solved because the system of equations is under defined. So we'll use the Pseudo-Inverse Jacobian method:

We have a known end effector position: **X** and initial guess **Y**

1. Compute the Jacobian
2. Compute the Pseudo-Inverse of the Jacobian
    a. $J^{+} = (J^{T}J)^{-1}J^{T}$
3. Compute the change in the end effector position Δe
    a. $\Delta e = (X-Y)$
4. Compute the change in the joint configuration Δθ
    a. $\Delta\theta = J^{+}\Delta e$
5. Compute the next joint configuration
    a. $\theta_{k+1} = \theta_k + \Delta\theta$
6. Repeat 1-5 until Δe is sufficiently small

Back

Backup Slide: Budget Tracking

Project Estimation Form

Project Name: RESCUE
Project Manager: Seth Krein

	Project Task	Material Cost ($)	Manufacturing Cost ($)	Other Cost ($)	Total per Task
1	Project Initiation				
1.1	PDD	$0	$0	$0	$0
1.5	CDD	$0	$0	$0	$0
	PDR	$0	$0	$0	$0
1.8	CDR	$0	$0	$0	$0
	Subtotal	$0	$0	$0	$0
2	Project Planning				
2.1	Gantt Chart and PM Software	$400	$0	$0	$400
	Subtotal	$400	$0	$0	$400
3	Project Delivery				
3.1	Base Mount Construction	$200	$0	$0	$200
3.2	Arm Sections Construction	$300	$0	$0	$300
3.3	Joint Construction	$400	$0	$0	$400
3.4	Servo Acquisition & Installation	$500	$0	$0	$500
3.5	Sensor and Lighting Mount Construction	$50	$0	$0	$50
3.6	Sensor Acqusition & Installation	$500	$0	$0	$500
3.7	Lighting Acqusition & Installation	$50	$0	$0	$50
3.8	Microcontroller Acquisition & Integration	$85	$0	$0	$85
3.9	Software development	$0	$0	$0	$0
	Subtotal	$2,085	$0	$0	$2,085
4	Project Closeout				
4.1	Assessment of overall project quality	$0	$0	$0	$0
4.2	Survey/Feedback of MARBLE	$0	$0	$0	$0
4.3	Final Presentation to Project Stakeholders	$0	$0	$0	$0
	Subtotal	$0	$0	$0	$0
10 - Other	General and Administrative Costs	$0	$0	$0	$0
11 - Other	Other cost	$0	$0	$0	$0
	Sub-Totals:	$2,485	$0	$0	$2,485
	Risk (Contingency):	$625	$250	$250	$1,125
	TOTAL (scheduled):	$3,110	$250	$250	$3,610

Back

Backup Slide: DARPA Subterranean Challenge

https://www.subtchallenge.com/#about

Backup Slide: MARBLE

- Multi-agent Autonomy with Radar-Based Localization for Exploration (MARBLE)
- CU Boulder’s team participating in the systems track of the DARPA Subterranean challenge
- $4.5 million dollar project led by the College of Engineering and Applied Science

Back

Backup Slide: Attitude Determination specifications

3-Space™ Watertight USB/RS232

General

Part number	TSS-USB-WT-S (Watertight Screw-down Sensor Unit)
Dimensions	36.4mm x 94.1mm x 58.6mm (1.43 x 3.71 x 2.31 in.)
Weight	150 grams (5.29 oz)
Supply voltage	+5v USB, +3.3v~+6.0v for RS232
Communication interfaces	USB 2.0, RS232 Asynchronous Serial
Serial baud rates	1,200~921,600 selectable, default: 115,200
Filter update rate	up to 250Hz with Kalman AHRS(higher with oversampling) up to 850Hz with QCOMP AHRS(higher with oversampling) up to 1350Hz in IMU mode
Orientation output	absolute & relative quaternion, Euler angles, axis angle, rotation matrix, two vector
Other output	raw sensor data, normalized sensor data, calibrated sensor data, temperature
Shock survivability	5000g
Temperature range	-40C ~ 85C (-40F ~ 185F)

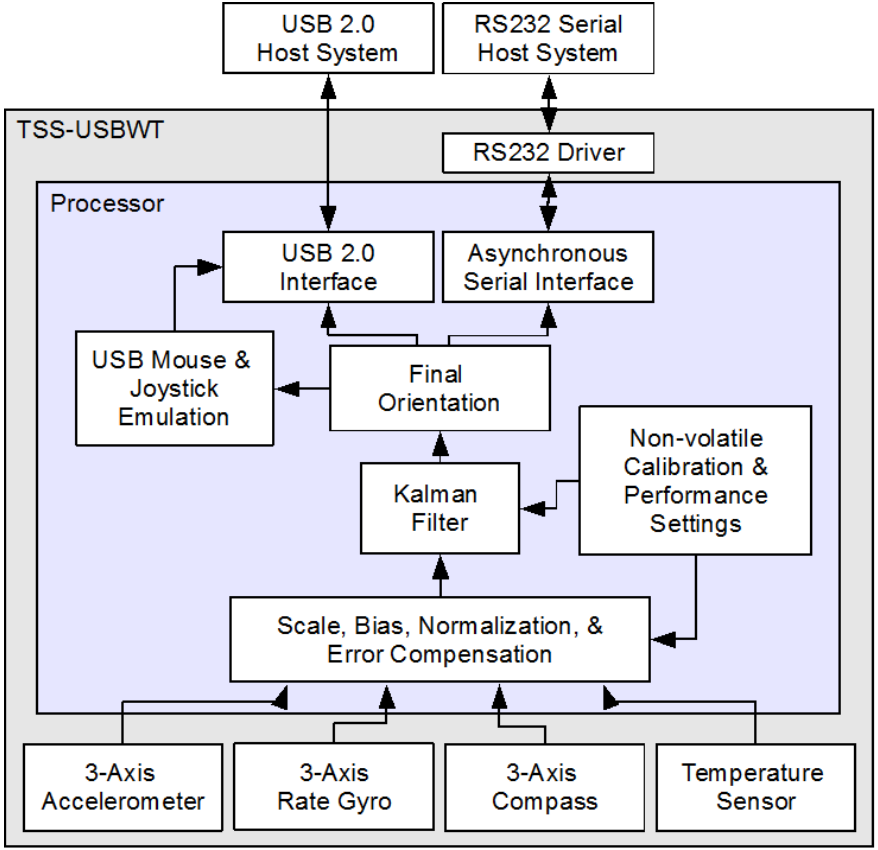
Sensor

Orientation range	360º about all axes
Orientation accuracy	±1º for dynamic conditions & all orientations
Orientation resolution	<0.08º
Orientation repeatability	0.085º for all orientations
Accelerometer scale	±2g / ±4g / ±8g selectable for standard models ±6g / ±12g / ±24g selectable for HH models(coming soon) ±100g / ±200g / ±400g selectable for H3 models(coming soon)
Accelerometer resolution	14 bit, 12 bit(HH), 12 bit(H3)
Accelerometer noise density	99µg/√Hz, 650µg/√Hz(HH), 15mg/√Hz(H3)
Accelerometer sensitivity	0.00024g/digit-0.00096g/digit 0.003g/digit-0.012/digit(HH) 0.049g/digit-0.195g/digit(H3)
Accelerometer temperature sensitivity	±0.008%/°C, ±0.01%/°C(HH, H3)
Gyro scale	±250/±500/±1000/±2000 °/sec selectable
Gyro resolution	16 bit
Gyro noise density	0.009°/sec/√Hz
Gyro bias stability @ 25°C	2.5°/hr average for all axes
Gyro sensitivity	0.00833°/sec/digit for ±250°/sec 0.06667°/sec/digit for ±2000°/sec
Gyro non-linearity	0.2% full-scale
Gyro temperature sensitivity	±0.03%/°C
Compass scale	±0.88 Ga to ±8.1 Ga selectable (±1.3 Ga default)
Compass resolution	12 bit
Compass sensitivity	0.73 mGa/digit
Compass non-linearity	0.1% full-scale

Back

Backup Slide: Attitude Determination FBD

3-Space™ Watertight USB/RS232



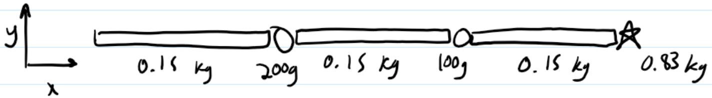
Back

Mech Backup Slides

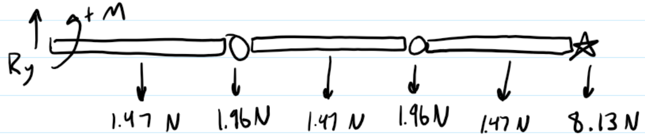
Backup Slide: Arm Calculations

- Hand calculations originally performed, switched to MATLAB script
- Use of statics, sum of forces and moments for the whole system = 0
- Parallel axis theorem used for dynamic moments

fully extended arm:

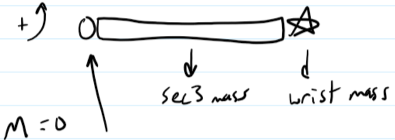


in terms of force:



Sectional Torques (worst case scenario – fully extended, horizontal

starting from the right:



$\Sigma M = 0$

servo 2 torque = $\left(\text{sec 3 mass} \times 9.81\right) \times \frac{\text{sec 3 length}}{2} + \left(\text{wristmass} \times \left(\text{sec 3 length} + \text{wristlength}/2\right)\right)$

(min. required)

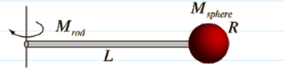
Dynamic Torque

$\Sigma M = I\alpha$

// axis thrm: $I = I_{cm} + mr^2$

rod about end: $I = \frac{1}{3} ML^2$

sphere about center: $I = \frac{2}{5} MR^2$



$$I = \frac{1}{3} M_{rod} L^2 + \frac{2}{5} M_{sphere} R^2 + M_{sphere} (L+R)^2$$

$I = I$ rod about end + I sphere about center + Parallel axis contribution

starting from the end:

joint 2: $I = \frac{1}{3} M_{sec3} L_{sec3}^2 + \frac{2}{5} M_{wrist} R_{wrist}^2 + M_{wrist} (L_{sec3} + R_{wrist})^2$

$I = \frac{1}{3}(0.0053 \text{ kg})(0.2 \text{ m})^2 + \frac{2}{5}(0.830 \text{ kg})(0.1 \text{ m})^2 + (830 \text{ kg})(0.2 + 0.1 \text{ m})^2$

$I = 0.0781$

[Back](#)

Backup Slide: MATLAB Script for Arm Calculations

RESCUE Calculations for Arm Mass, Size, Strucutal Analysis

Evan Welch

```
clear all; clc;
% all units are in standard metric, for obvious reasons :)
```

Define Variables

```
mat_density = 1661; % density of tube in kg/m^3
OD = 0.047625; %outer tube diameter
ID = 0.04445; %inner tube diameter

% arm section lengths in meters:
sec1length = 0.4;
sec2length = 0.4;
sec3length = 0.2;

% joint and wrist masses in kg:
joint1mass = 0.200;
joint2mass = 0.1;
wristmass = 0.83;

% joint and wrist lengths in meters:
joint1length = 0.05;
joint2length =0.05;
wristlength = 0.207;
```

Arm mass and size

Lengths:

```
mat_length = sec1length + sec2length + sec3length; % length of tubing
total_length = mat_length + joint1length + joint2length + wristlength; % total arm length

% Mass:
cross_section = OD*pi*(OD/2)^2 - ID*pi*(ID/2)^2; %cross sectional area
mat_volume = cross_section*mat_length; % volume of tubing
arm_sections_mass = mat_density*mat_volume; % mass of tubing
total_mass = arm_sections_mass + joint1mass + joint2mass + wristmass; %total arm mass
```

```
% Masses of the tubes:
sec1mass = sec1length*cross_section*mat_density;
sec2mass = sec2length*cross_section*mat_density;
sec3mass = sec3length*cross_section*mat_density;
```

Reaction forces and moments (worst case scenario: fully extended horizontal arm)

all forces are vertical in this case

```
Ry_worstcase = total_mass * 9.81; % vertical reaction force in newtons
Rm_worstcase = 9.81*(sec1mass*(sec1length/2) + joint1mass*(sec1length+(joint1length/2)) + sec
2mass*(sec1length+joint1length+(sec2length/2)) +...
    joint2mass*(sec1length+joint1length+sec2length+(joint2length/2)) + sec3mass*(sec1length+j
oint1length+sec2length+joint2length+(sec3length/2))+...
    wristmass*(total_length-(wristlength/2))); % reaction moment at the base in Nm
```

Reaction forces and moments (stowed)

```
Ry_stowed = Ry_worstcase; %same total vertical force regardless of configuration
Rm_stowed = 9.81*(sec1mass*(sec1length/2) + joint1mass*(sec1length+(joint1length/2)) + sec2ma
ss*(sec2length/2) + joint2mass*(sec2length+(joint2length/2)) + ...
    sec3mass*(sec3length/2) + wristmass*(sec3length+(wristlength/2))); % reaction moment at t
he base in Nm
```

Sectional torques (servo torque needed for worst case scenario)

starting from the wrist:

```
servo2torque = 9.81*(sec3mass*(sec3length/2) + wristmass*(sec3length+wristlength/2)); % torqu
e necessary for servo in joint 2
servo1torque = 9.81*(sec2mass*(sec2length/2) + joint2mass*(sec2length+(joint2length/2)) + sec
3mass*(sec2length+joint2length+(sec3length/2)) +...
    wristmass*(sec2length+joint2length+sec3length+(wristlength/2))); %torque necessary for se
rvo in joint 1
basetorque = Rm_worstcase; %torque necessary for the base
```

Back

Backup Slide: Arm Structural Calculations

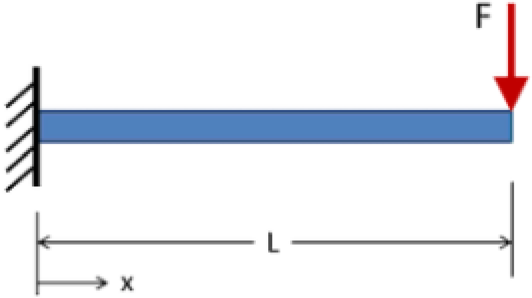
- Arm modeled as a cantilever beam with a fixed end and a point load
- Force calculated based on maximum moment on base

Assumptions:
- Fixed end
- Uniform beam

$$\sigma_{b.max} = \frac{Mc}{I_c}$$

$$\tau_{max} = \frac{VQ_{max}}{I_c b} = \frac{4V}{3A}\left(\frac{r_o^2 + r_o r_i + r_i^2}{r_o^2 + r_i^2}\right)$$

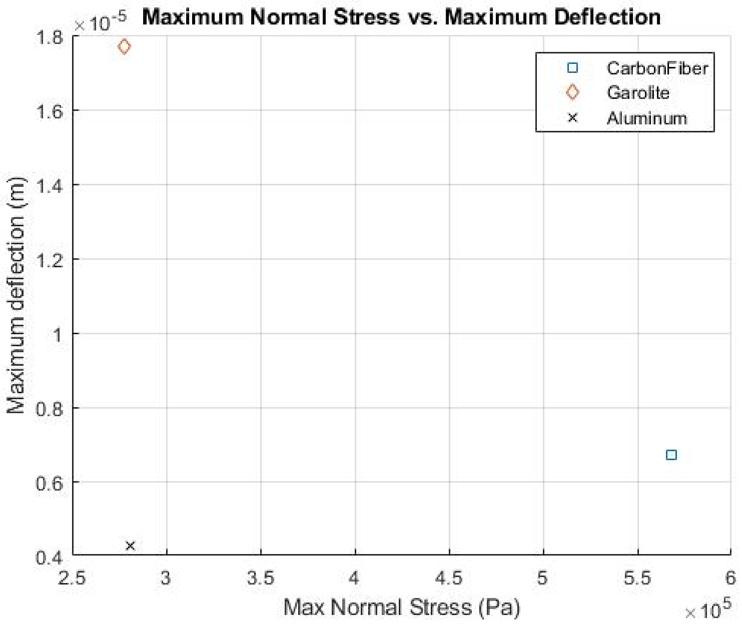
$$\delta_{max} = \frac{FL^3}{3EI} \qquad @\, x = L$$

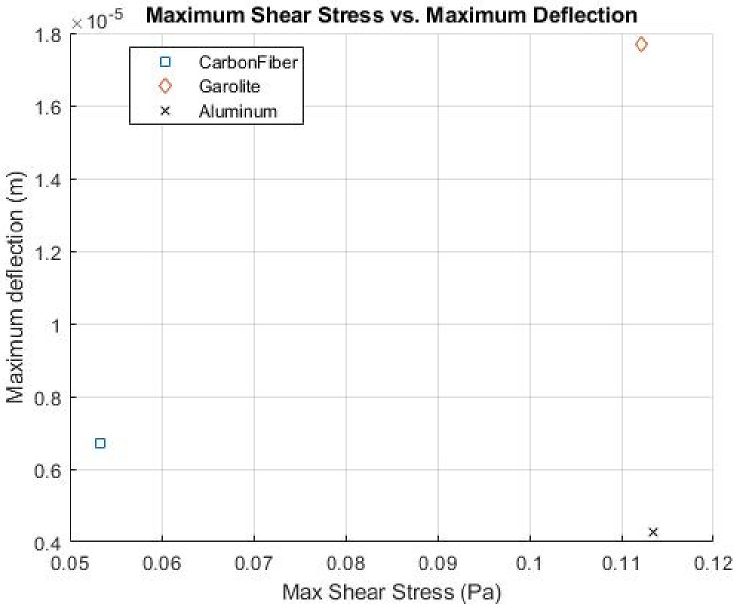


Material	Max Normal Stress	Max Shear Stress	Max Deflection
Aluminum	0.281 MPa	0.1135 Pa	$4.28 * 10^{-6}$ m
Carbon Fiber	0.568 MPa	0.0534 Pa	$6.73 * 10^{-6}$ m
Garolite	0.277 MPa	0.1121 Pa	$1.76 * 10^{-5}$ m

Backup Slide: Arm Structural Calculations

- Maximum normal and shear stress vs maximum deflection for carbon-fiber, garolite G10, and aluminum 6061

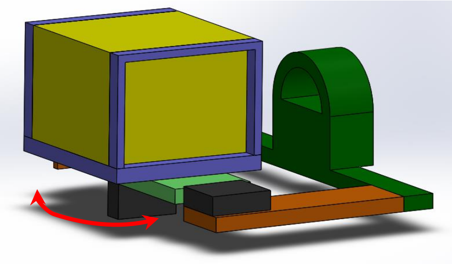


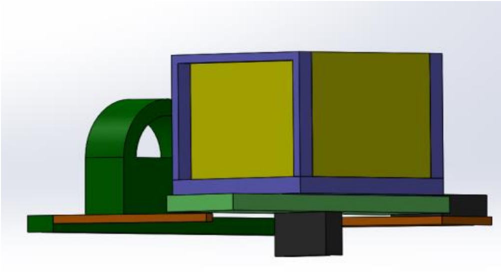


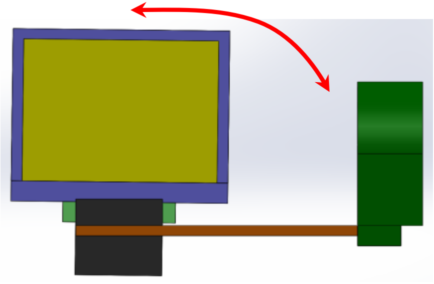
Back

Backup Slide: Wrist Mass Evaluation

1) Sensor package
2) Sensor mounting plate
3) Support Fork
4) Servos
5) Rotation Plate
6) Connection to arm







Wrist Section Masses

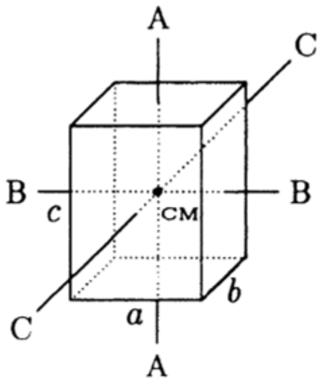
Component	Mass (g)
1	305
2	150
3	40
4a, 4b	22, 22
5	105
6	183
Total	**827**

Key Assumptions

- Sensor package includes D435i, SCD30, Yost Labs 3-Space IMU, and 2 Streamlight MicroStream lights (worst case)
  - SCD30 may be on arm
- Solid PLA (1.24 g/cm^3) mounting components
- 6) Evaluated only half cut out for arm
- 360° Rotation capability, both axes
- 1 cm thick bases for 2) and 5)
- 5mm thick fork sections
- 1x2 cm cross section for 6) horizontal portion

Back

Backup Slide: Wrist Torque Analysis



Rectangular Prism

$$I_{AA} = \frac{1}{12} m\left(a^2 + b^2\right)$$
$$I_{BB} = \frac{1}{12} m\left(b^2 + c^2\right)$$
$$I_{CC} = \frac{1}{12} m\left(c^2 + a^2\right)$$
$$V = abc$$

Parallel Axis Theorem:

Mass Moment of Inertia

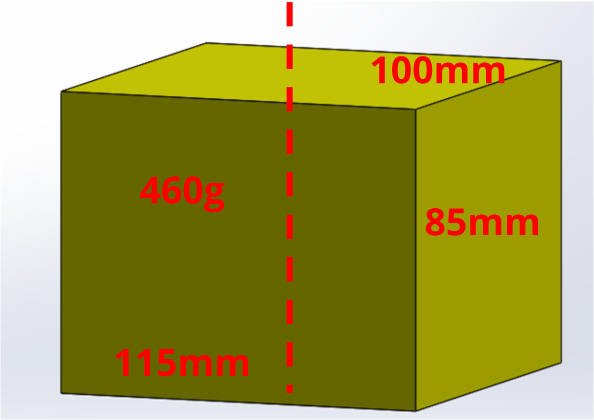
$$I_O = I_C + \underset{\substack{\uparrow \\ \text{Mass of Body}}}{m}\, d^2$$

<u>Assumptions</u>

- 90◦ in 1s from rest
- Uniform Density Blocks

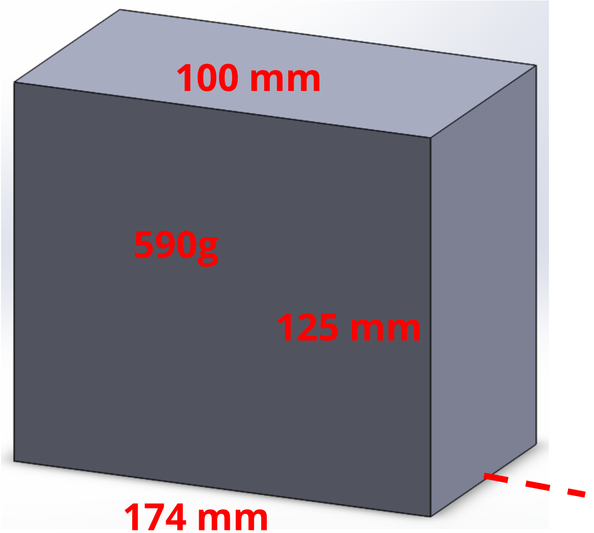
<u>Servo A:</u>

- Sensor Package (1) + Sensor Mounting Plate (2)
- Servo torque centered



<u>Servo B:</u>

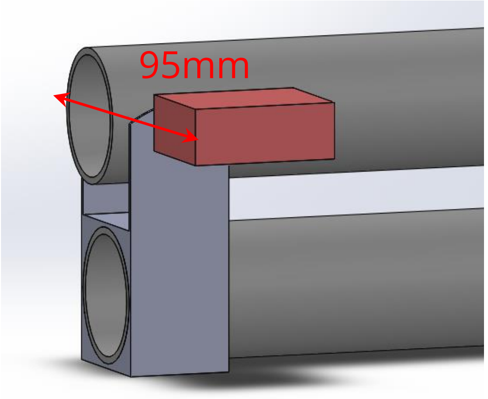
- Sensor Package (1), Sensor Mounting Plate (2), Servo A (4), Rotation Plate (5)
- Servo torque applied at bottom edge (worst case)



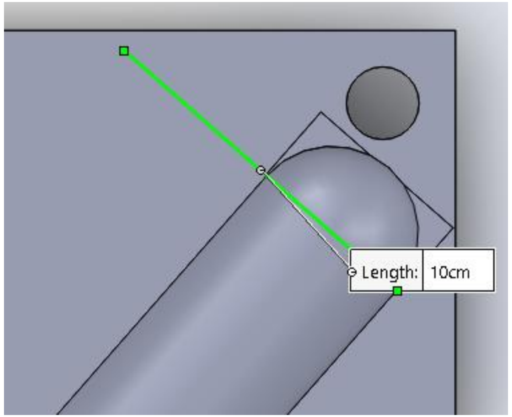
Back

Backup Slide: Mechanical Joints

- Largest Possible 70 Kg Cm Servo (Joint 1)
- Additional Volume Fitting Assessment Required



Worst Case Fit Scenario



Bevel Gearing

Large Example Servo

- 54x40x20mm

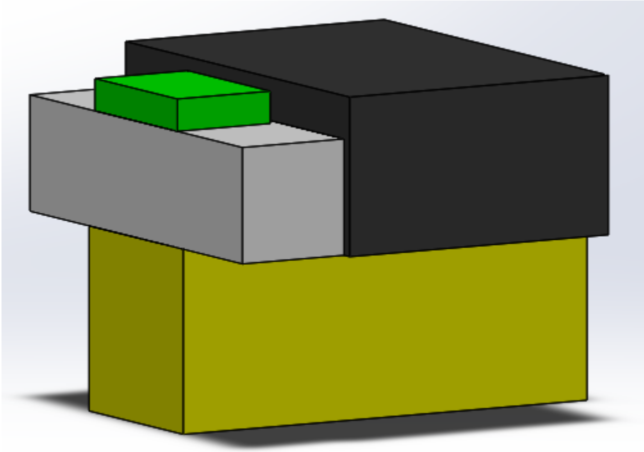
[Back](#)

Backup Slide: Cable Routing

Universal Robotics:
BN UNI-Kit Flex

Abb Industrial: 6 DOF Arm

Backup Slide: Sensor Package Fit



Sensor Package Components

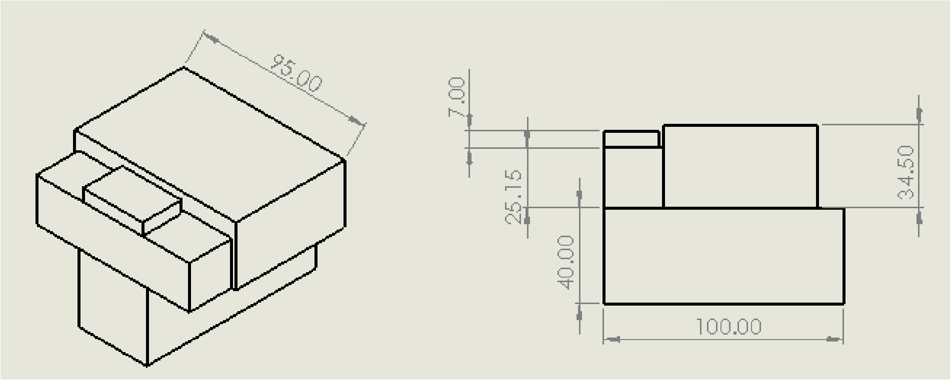
IMU

CO_2 Sensor

Camera

Lights

Total Dimensions (mm)



[Back](#)

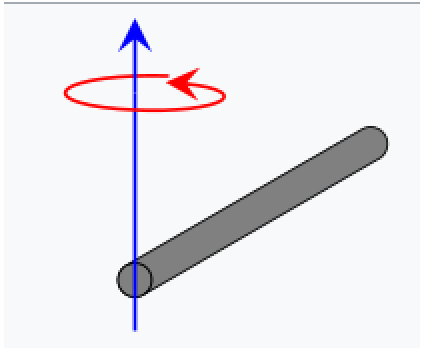
Backup Slide: Base/Waist Design

Assumptions:

- Solid rod of uniform density
- Infinitely thin rod
- α = 10 rad/s

I_{end} = ⅓ m L^2
τ = I * α
m = 1.156 kg
L = 1.307 m

$\tau_{required}$ = 6.5825 Nm



Possible base servo setups:

Patented

Back

Backup Slide: Tipping Analysis

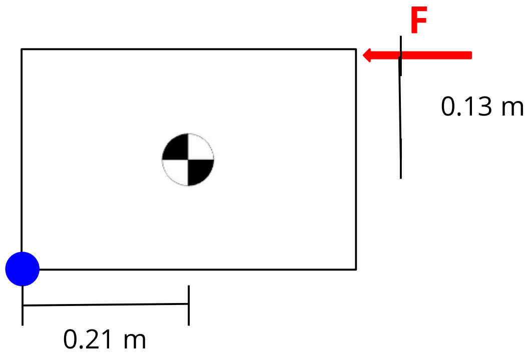
Assumptions:

- Husky Rover is a simple box with uniform density
- Force is induced from the moment of the robotic arm at full extension

Counterclockwise Moment = Clockwise Moment

$\mathbf{F{*}d_{ccw} = F{*}d_{cw}}$

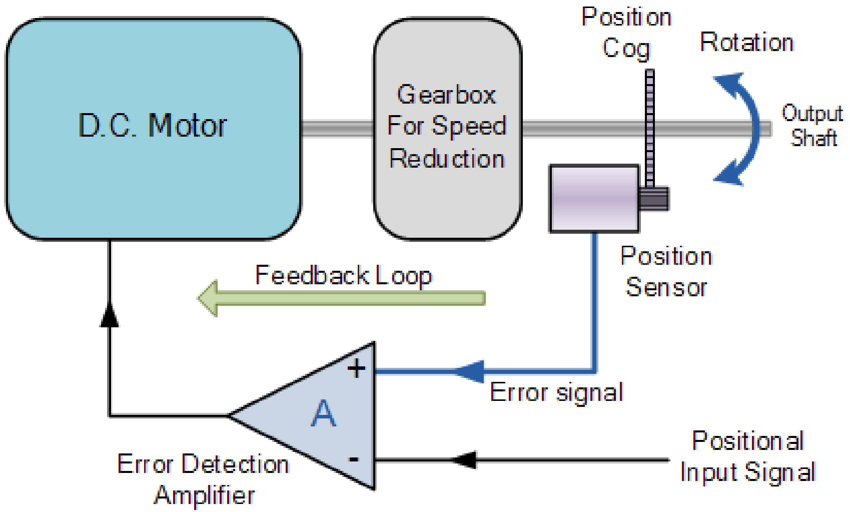
$\mathbf{F_{needed\ to\ tip} = 792.346\ N}$
$\mathbf{F_{arm\ at\ full\ extension} = 89.23\ N}$



Back

Backup Slide: Servo DC Motor

100 kg-cm (left) and 70 kg-cm servos



Example closed loop for a servo

Back to Main Servo Slide Back

Backup Slide: Stepper Motors

- Possible Stepper solutions up to 50 Nm
- Open-loop
- Issues with weight, size, and power
- Hard implementation of controls

https://www.jvl.dk/326/high-torque-stepper-motors

[Back to Main Servo Slide] [Back]

Backup Slide: Water/Dust Proofing

- Most servos under consideration are already waterproof
- Sealant shall be used to fill gaps
- Wire wrapping solutions (see cable routing slide)
- Worst case: covering

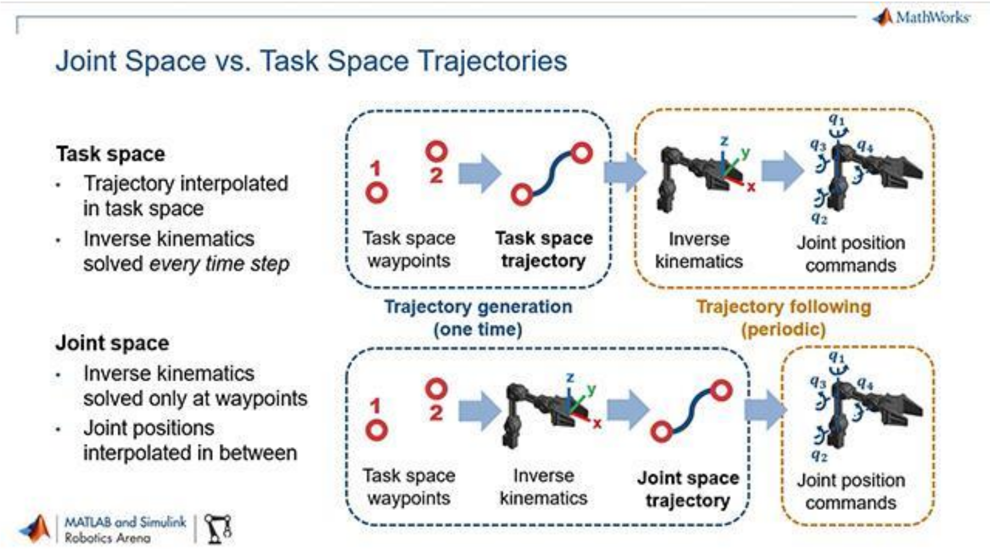
Universal Robotics UR+ Lightweight Cover

Generic Camera Splash Cover

Back to Main Servo Slide Back

Backup Slide: Task Space vs Joint Space

	Task Space	Joint Space
Pros	• Motion is predictable (interpolation in task space) • Better handling of obstacles and collisions	• Faster execution (solve IK at waypoints only) • Actuator motion is smooth and easier to validate
Cons	• Slower execution (solve IK every time step) • Actuator motion not necessarily smooth and harder to validate	• Intermediate points not guaranteed to respect joint limits or collisions



Backup Slide: PD Control

$$u = K_P\tilde{q} - K_D\dot{q} \qquad (8.7)$$

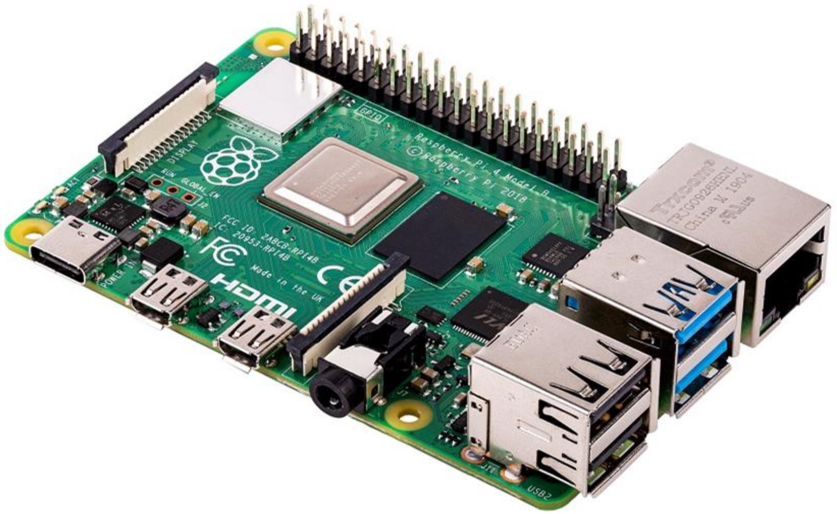
where $\tilde{q} = q^d - q$ is the error between the desired joint displacements q^d and the actual joint displacements q

$$\begin{pmatrix} u1 \\ u2 \\ u3 \\ u4 \\ u5 \\ u6 \end{pmatrix} = \begin{pmatrix} K11 & K12 & K13 & K14 & K15 & K16 \\ K21 & K22 & K23 & K24 & K25 & K26 \\ K31 & K32 & K33 & K34 & K35 & K36 \\ K41 & K42 & K43 & K44 & K45 & K46 \\ K51 & K52 & K53 & K54 & K55 & K56 \\ K61 & K62 & K63 & K64 & K65 & K66 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ \dot{X} \\ \dot{Y} \\ \dot{Z} \end{pmatrix} - \begin{pmatrix} K11 & K12 & K13 & K14 & K15 & K16 \\ K21 & K22 & K23 & K24 & K25 & K26 \\ K31 & K32 & K33 & K34 & K35 & K36 \\ K41 & K42 & K43 & K44 & K45 & K46 \\ K51 & K52 & K53 & K54 & K55 & K56 \\ K61 & K62 & K63 & K64 & K65 & K66 \end{pmatrix} \begin{pmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \\ \ddot{X} \\ \ddot{Y} \\ \ddot{Z} \end{pmatrix}$$

Spong, Mark W., Seth Hutchinson, and Mathukumalli Vidyasagar. *Robot Modeling and Control*. Hoboken, NJ: Wiley, 2006.

Back

Backup slides: Microcontroller



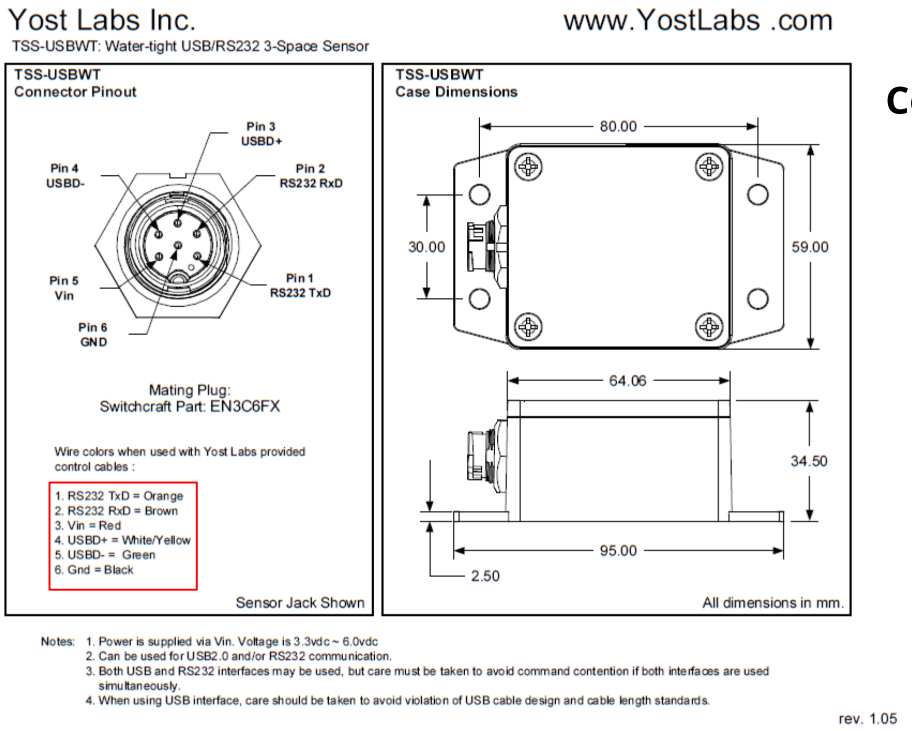
FR3.1	RESCUE shall determine and report its location and orientation relative to the ground robot.	System Position and Orientation
FR6.1	RESCUE shall communicate its sensed data with MARBLE and this process shall not interfere with MARBLE's communication systems. RESCUE shall be able to receive firing commands, nested firing commands,and deactivation commands from MARBLE's team.	Communications

Raspberry Pi 4 Model B:

- **Processor:** Broadcom BCM2711, quad-core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz
- **Memory:** 2/4/8 GB LPDDR4 (depending on model)
- **Connectivity:**
  - 2x USB 3.0 ports
  - 2x USB 2.0 ports
  - Gigabit Ethernet port
  - 2.4 GHz and 5.0 GHz IEEE 802.11b/g/n/ac wireless LAN, Bluetooth 5.0, BLE
- **GPIO:** Standard 40-pin GPIO header
- **SD Card Support:** Micro SD card slot for loading operating system and data storage
- **Power:** 15W required

Back

Backup slides: AHRS sensor feasibility



Courtesy of Yost Labs

Compatible with the microcontroller

Backup slides: Communication feasibility

Wiring Connections

- Gigabit Ethernet from MARBLE to Raspberry Pi (RasPi)
- USB-A 3.0 to USB-C 3.0 from RasPi to RealSense camera
  - Continuous-flex-and-twist cable

Backup slides: Interfacing with MARBLE

- Graphic User Interface
- Command Line interface

Back

Backup slides: Subsystems

Backup: Subsystems Breakdown

- RESCUE
  - Software
    - Controls
      - Mapping and Motion planning
      - Execution
    - Communications
  - Mechanical
    - Joints
      - Arms connections
      - Base
      - Wrist
    - Arms

Backup Slide: Embedded System