MIND THE GAP

Autonomous Collaborative Ground Traversal of Discontinuous Terrain

Critical Design Review Report

TEAM A

Dhanvi Sreenivasan | Hari Shankar Ravisankaran | Sandhya Vidyashankar | Sankalp Chopkar | Sudhansh Yelishetty

Sponsors: Prof. Zeynep Temel | Prof. Katia Sycara | Sha Yi

Date: May 6, 2022

Master of Science
Robotic Systems Development

Carnegie Mellon University
The Robotics Institute

Table of Contents:

1. Project Description **3**
2. Use Case **3**
3. System-Level Requirements **4**
    3.1 Mandatory Requirements 4
        3.1.1 Functional Requirements 4
        3.1.2 Performance Requirements 5
        3.1.3 Non-Functional Requirements 5
    3.2 Desirable Requirements 6
4. Functional Architecture **6**
    4.1 System Architecture 6
    4.2 Server functional architecture 6
    4.3 Agent functional architecture 7
5. Cyber-physical architecture **8**
    5.1 Server cyber-physical architecture 9
    5.2 Agent cyber physical architecture 10
6. Current System Status **11**
    6.1 Spring Semester Targeted Performance Requirements 11
    6.2 Overall system depiction 11
    6.3 Subsystem descriptions 12
        6.3.1 Controllers 12
        6.3.2 Coupling Mechanism 12
        6.3.3 Mapping and Localization 13
        6.3.4 Preprocessing 13
        6.3.5 Task allocation 13
        6.3.6 Multi-robot planning 14
        6.3.7 Fleet management system 14
    6.4 Modeling, Analysis and Testing 14
    6.5 SVD performance evaluation 15
    6.5 Strong and Weak Points 15
7. Project Management **17**
    7.1 Work Breakdown Structure 17
    7.2 Project Management: Schedule Status 18
    7.3 Project Management: Fall Test Plans 19
    7.4 Project management: Parts list and Budget 19
    7.5 Risk Management 24
8. Conclusions **25**
        8.1 Lessons Learnt 25
        8.2 Key Fall Activities 25
9. References **26**

1. Project Description

Robot swarms have been shown to improve the ability of individual robots by inter-robot collaboration. Introducing physical coupling between robots can further extend their ability in an unstructured environment. This is particularly beneficial in inspection scenarios, especially in mines, where the environment is complicated and thus challenging. Instead of having the crew spend precious day hours inspecting several sites in a mine in person, they can deploy a swarm of robots to do the same in a more efficient and collaborative manner.

Rather than sending in a team of people for inspection at each site, it is more efficient to send a swarm of robots to collect more information at each point of interest to the crew. Each agent in the system will inspect a set of allocated points of interest, and once every point is done, they will return all the information to the end users, using which further plans can be made.

Previously, PuzzleBots [1][2] was proposed for the same. It is a robotics swarm system where the robots can dynamically couple with each other to form bridges and decouple to perform individual tasks but on a small scale. This system can either be used to build bridges for the transportation of other materials, or to get the robots from one side of a gap to the other. The aim of this project is to investigate the physical coupling of robot swarms on a larger scale, as well as implement and test existing control and planning algorithms on a larger system.

2. Use Case

Consider a mine with several hotspots, where a hotspot is a location with high concentrations of mineral deposits. These minerals are of critical importance and the mining crew is on a time crunch. That being said, each hotspot in the mine must be inspected in detail before approaching the next steps.

Mines, in general, can be narrow, in caves, and can have cracks or fissures on the ground or other surfaces. Due to rugged terrains in such unstructured environments, traversing these conditions can be potentially risky for humans, e.g., collapse of overhead rock structures. The mining crew needs to inspect each hotspot but will lose out on time and need to find a better solution. An autonomous distributed solution.

Bring in PuzzleBots. The crew inputs a blueprint of the map of the mine. This would typically be an occupancy grid of the mine with hotspots marked in them. All agents in the system start from the base station at the entrance of the mine. Each agent gets allocated hotspots optimally, and collision-free optimal trajectories are planned. The agents branch off from the swarm to inspect individual hotspots. In case of emergencies involving imminent collisions, each agent is equipped with a collision avoidance system. They spend a predetermined fixed amount of time at the hotspots before being allocated new ones.

It is very likely that hotspots are located across fissures in the mine, which will be referred to as gaps. Should there be any hotspots on either side of any gaps, the system determines the feasibility of getting there. If feasible, the system plans for a group of agents to physically couple with each other and cross the gap. After that, they split up, and follow their respective planned trajectories to inspect individual hotspots. After inspecting the entire map, the agents return to the base station.

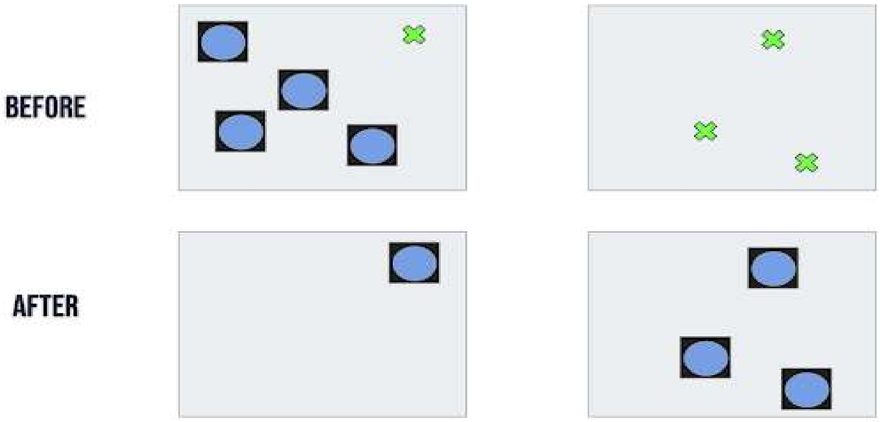


Figure 1. Use Case Illustration

3. System-Level Requirements

Our system-level requirements were elicited from the use case, discussions with our sponsors, and internal team meetings based on the time and budget constraints to the project timeline. All the requirements are shown in Table 1 to Table 3.

3.1 Mandatory Requirements

3.1.1 Functional Requirements

Table 1. Functional Requirements

	Requirement
MF1	The system shall localize agents in a given map.
MF2	The system shall route agents and avoid collisions.
MF3	The system shall determine feasible gaps.
MF4	The system shall determine and achieve coupled configurations.
MF5	The system shall cross gaps.
MF6	The system shall reach given regions of interest.

3.1.2 Performance Requirements

Table 2. Performance Requirements

	Requirement
MP1	The system will cross gaps up to 1.3 agent lengths.
MP2	The system will have 0 unplanned collisions between agents.
MP3	The system will achieve formations with at least 3 robots.
MP4	The system will cross feasible gaps 75% of the time.
MP5	The system will reach all POIs 75% of the time.
MP6	The coupling mechanism will bear the weight of one agent.
MP7	The coupling mechanism will self-align against heading errors of 5 degrees and position error of 10 mm.

- MP1 was modified from before so as to not damage the robots' hardware over time. The robots rely on momentum to cross gaps. With larger gaps, the robots hit the edges of the gaps at a larger impact. We observed that the caster wheels of the robots pop out of their sockets from time to time when trying to test this out. In order to not damage the robots further, we rescoped this requirement.

3.1.3 Non-Functional Requirements

Table 3. Non-Functional Requirements

	Requirement
MN1	The weight of an agent shall be minimal.
MN2	The coupling mechanism shall consume a low amount of energy.
MN3	The system shall be scalable.
MN4	The system shall be easily maintainable.
MN5	The team shall maximize learning and fun throughout the project through flexible work plans across subsystems.

3.2 Desirable Requirements

We have chosen not to go with any desirable requirements because the project is highly complex.

4. Functional Architecture

4.1 System Architecture

The system is divided into two blocks: the server and the agents. A brief overview of the functions of agents and the server is shown below:

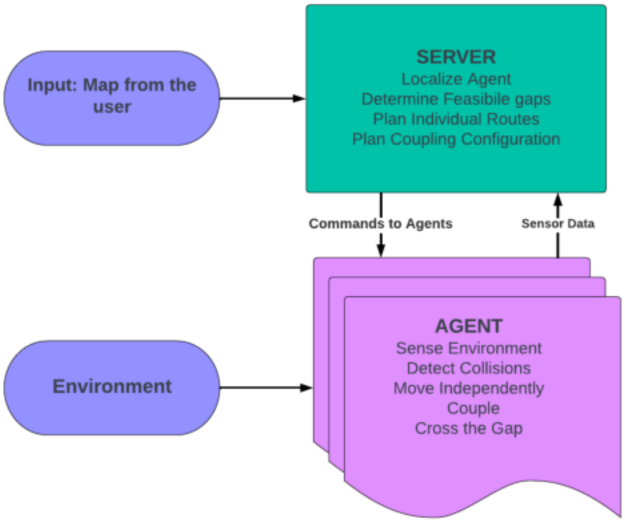


Figure 2. Functional Architecture of the System

4.2 Server functional architecture

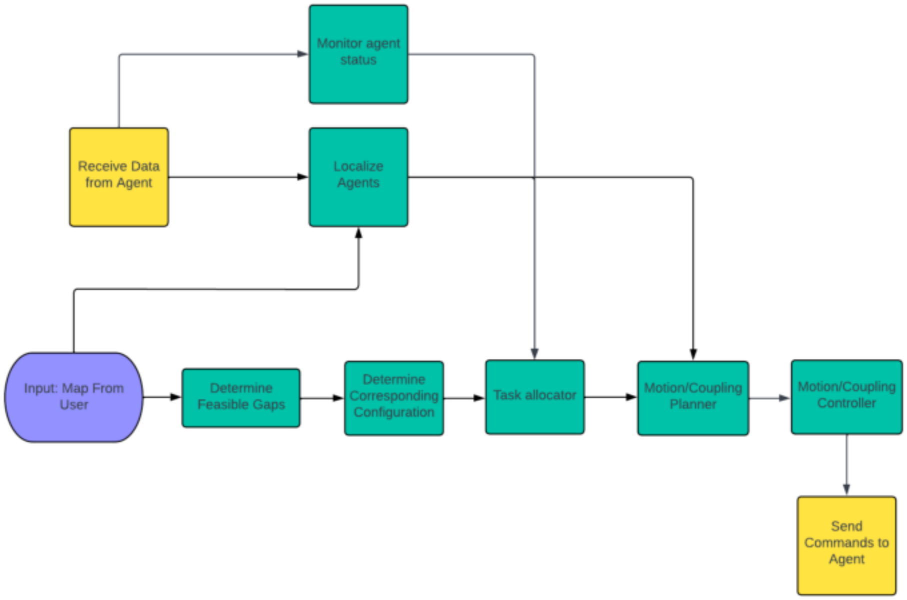


Figure 3. Functional Architecture of the Server

The user provides an initial map of the environment to the server which includes information about the gaps, the points of interest, and the initial agent poses. The server receives sensor data from the agents. The server monitors the agent's status and localizes each agent within the environment. Based on the input map, the server determines the feasible gaps and also the

configuration required to cross the gaps. The task allocator takes in this information and allocates specific tasks to each individual agent.

The tasks assigned are then sent to the planner block which generates either collision-free paths for agents to cover the points of interest or coupling and gap-crossing plans if the task is to cross the gaps. To execute the planned paths, the controller block computes the error between the desired and current agent pose to generate linear and angular velocity commands for movement and coupling commands to enable coupling.

4.3 Agent functional architecture

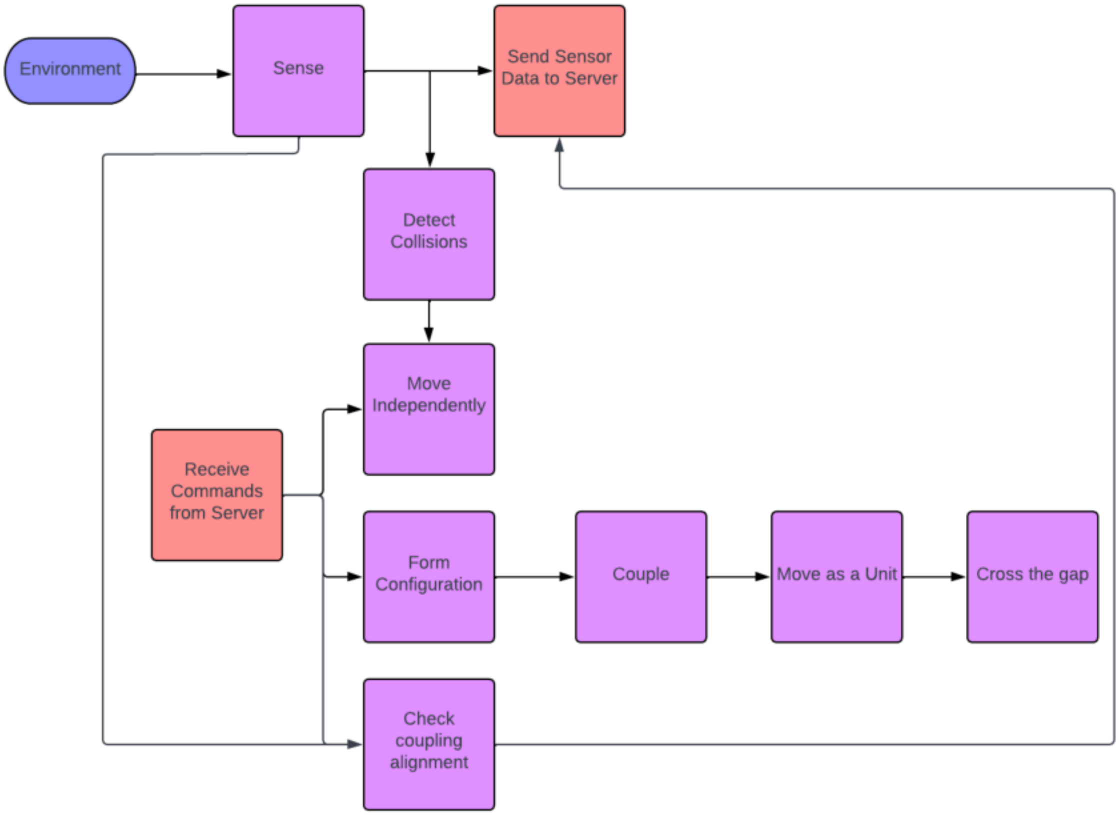


Figure 4. Functional Architecture of the System

The agent receives linear, angular velocities, and coupling commands from the server. The linear and angular velocities are used to generate the motor commands that enable the agent to move independently to cover points of interest, form configuration, and move as a unit depending upon the state of the agent. When the agent moves, the sensor readings are continuously transmitted to the server. The sensor data is also used to detect imminent collisions and avoid them. The coupling commands are used to actuate the coupling motors to enable successful coupling between the agents. To refine localization estimates for coupling, the agent also performs coupling alignment checks using the sensor readings which are then sent to the server.

5. Cyber-physical architecture

5.1 Server cyber-physical architecture

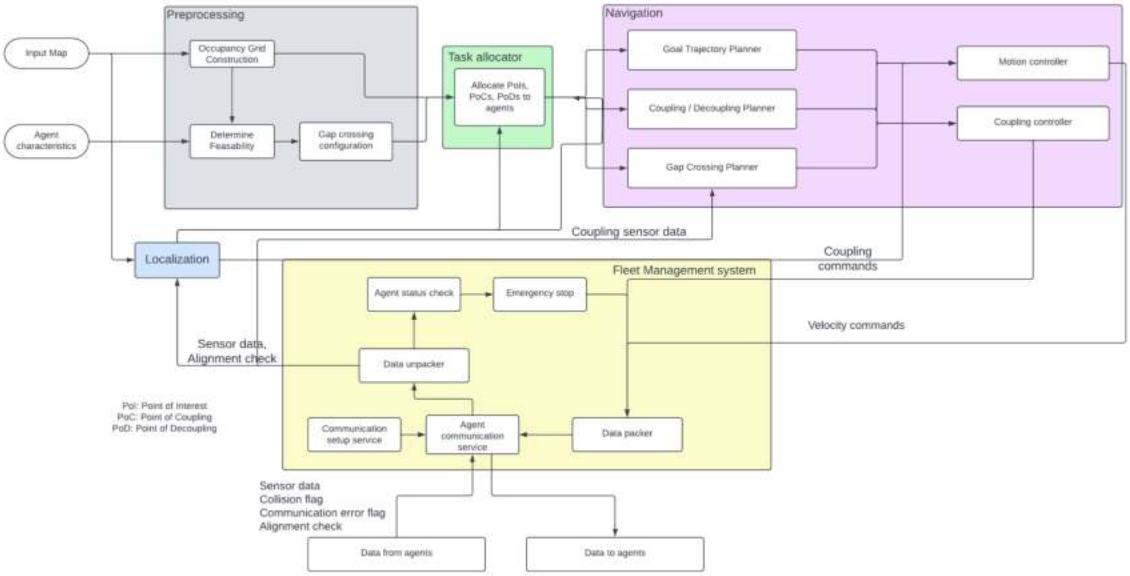


Figure 10. Server Cyber-Physical Architecture

The server acts as the brain of the system and controls the behavior of the individual agents. It takes an initial map of the environment and a config file from the user. The map contains information about the gaps and points of interest. The config file contains information about all the available agents. The FMS block uses the config file to set up communication service between the agents and the server. FMS handles the packing of data that needs to be sent to the agents and the unpacking of data from the agents to feed them into the corresponding subsystems. The map and the sensor information are used by the localization subsystem to determine the poses of each agent. The FMS also monitors the individual agents to check for communication failures.

The map provided by the user is preprocessed into Voronoi diagrams/connected grids to feed into the task allocation block. The preprocessing block also determines the feasible gaps in the map and the corresponding configuration required to cross those gaps. Based on these, the task allocator assigns tasks to each individual agent. The tasks can be covering points of interest or reach points of coupling and decoupling. Allocation is formulated as a modified Multiple Traveling Salesman problem to account for the gaps and coupling behavior.

The assigned tasks are then passed to the navigation subsystem. The navigation subsystem includes two major blocks: the trajectory planner and the controller. Based on the assigned tasks, for each agent, the goal trajectory planner will be used to generate collision-free paths to cover points of interest, the coupling planner will be used to generate coupling paths and commands, the gap crossing planner will be used to generate paths for the coupled robots to cross the gaps. The controller takes in the generated paths and the actual agent pose to compute the desired linear, angular velocities for agent movement and coupling commands for enabling coupling. These commands are then passed to the FMS to send to the agents.

5.2 Agent cyber physical architecture

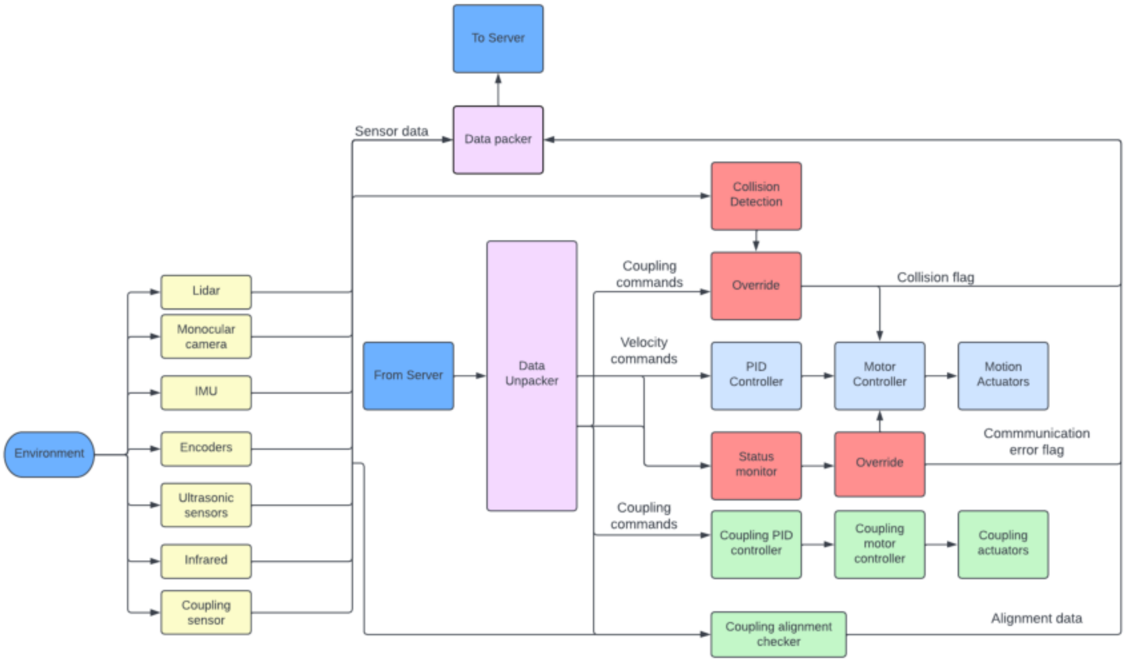


Figure 11. Agent Cyber-Physical Architecture

Based on the trade studies, we are using Khepera 4 robots for the agents. The platform was primarily designed for testing multi-agent systems, so they are suitable for our purpose. The sensors include wheel encoders, a 3-axis gyroscope, a 3-axis accelerometer, infrared sensors, ultrasonic sensors, laser range finders, RGB camera. All of these sensor data are packaged and transmitted to the server. The agent runs a collision detection service on its own to detect imminent collisions and avoid them. It also monitors its own status based on the server commands and sends messages to the server in case of any failure. The velocity and coupling commands from the server are passed to the respective controllers which convert them to actuator velocities. For refinement of localization estimates during coupling, the agent also runs an alignment-checking algorithm based on Aruco markers and sends the processed data to the server.

6. Current System Status

6.1 Spring Semester Targeted Performance Requirements

The following functional requirements were targeted in the spring semester:

Table 4. Spring semester Functional Performance Requirements

	Requirement
MP1	The system will cross gaps up to 1.3 agent lengths.
MP2	The system will have 0 unplanned collisions between agents.
MP3	The system will achieve formations with at least 3 robots.
MP4	The system will cross feasible gaps 75% of the time.
MP5	The system will reach all POIs 75% of the time.
MP6	The coupling mechanism will bear the weight of one agent.

The functional performance requirements covered all subsystems including task allocation, localization, planning, control and mechanical subsystems. The performance requirements related to autonomous coupling with perception and integration of the two isolated parts are left for the fall semester. We were successfully able to demonstrate all the above requirements during SVD and SVD encore.

6.2 Overall system depiction

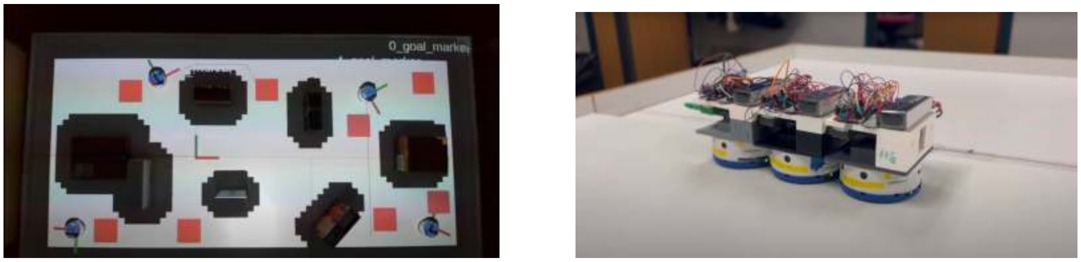


Figure 12. Overall System depiction

Figure 12 shows the overall system depiction. The left image shows the first part of our targeted requirements where the user can choose the points of interest in a given map. Based on the user input, the tasks are allocated to the agents optimally, multi-agent collision free paths are planned by the planning subsystem and executed reliably by the controller to cover all the points of interest. The right image shows the second part of our requirements to demonstrate the coupling behavior. The agents are manually aligned, then based on the user input, they couple and cross the desired gap length and decouple on the other side of the gap.

6.3 Subsystem descriptions

6.3.1 Controllers

The goal of the controller subsystem is to execute the collision free paths given by the planner reliably and ensure time synchronization. It also needs to communicate with the planning subsystem to reinitiate the planning process whenever one of the agents reach their goal location. We tried two different approaches using PID and pure pursuit controller. The pure pursuit method did not give good results due to the fact that the use of lookahead distance does not guarantee collision avoidance. The current implementation for SVD used a dual PID control to control the linear and angular velocity independently. The switching between the two controllers takes place based on an error threshold for the linear and angular component. Additionally, time synchronization was implemented to ensure that the agents avoid collisions between themselves.

6.3.2 Coupling Mechanism

This subsystem is a hardware addition to the robot mobility platform Khepera IV. It is responsible for facilitating the physical coupling between robot agents, allowing them to collaborate and traverse discontinuous terrain. The coupling mechanism uses a pin-and-hole method of mechanical coupling and is developed from electromechanical components like a Linear actuator to actuate the pin, a Solenoid Lock to latch on to the pin after complete actuation, Roller support to facilitate the sliding of the pin.

The electromechanical system consists of ESP8266- Node MCU (Microcontroller with WIFI module), L298N (motor driver for linear actuator), 3v Single Channel Relay (for solenoid lock), power distribution board with 11.1V power input from the battery and 3 outputs (24V, 5V and 6V), Linear actuator and Solenoid lock.

With the present coupling mechanism, 3 agent 1 dimensional coupling configuration can be achieved post pre-alignment and can successfully cross 19 cm gaps.

6.3.3 Mapping and Localization

The pose of every agent on the map needs to be determined precisely to enable coupling between the robots and cover the points of interest efficiently. The user provides a map and the initial positions of the agents. The sensor readings from the agent will be used for getting the pose estimate. The wheel encoders provide an initial first estimate of the robot's pose. Then the data from the laser range finder is used to refine the estimate of the change in the pose by matching scans of consecutive frames. The wheel encoder provides the odometry and the laser range finder (LRF) and the odometry together help in building the map, which is fed downstream to the localization, task allocation, and planning subsystems. We also explored OpenCV-based map building methods to enable quick variability of maps.

The laser scans are also matched with the provided map to provide localization information, coupled with the wheel odometry. This process is done separately for each agent using the map provided by the user. As of now, since we require very accurate localization for multi-agent motion planning and coupling, we use the VICON tracker system by placing trackable patterns as markers on the robots. During the coupling process, the localization precision should be high. Therefore, when the robots are sufficiently close to each other during coupling, additional redundant methods like Aruco marker-based pose estimation will also be used along with the previously mentioned method.

6.3.4 Preprocessing

The preprocessing block involves the processing of the map into Voronoi diagrams/connected grids or any other format required by the planning block. Additionally, the map also contains information about the gaps in the system. The preprocessing block determines all the feasible gaps in the map based on the number of agents available and also determines the configuration required by the agents to cross those feasible gaps. The infeasible gaps will be considered obstacles in the map. The outputs from the system include the processed map, feasible gaps, and their corresponding configuration.

6.3.5 Task allocation

The task allocator assigns uncompleted tasks to each agent based on the inputs from the preprocessing block while minimizing the overall time taken. Specifically, it will allocate points of interest to cover each agent and also allocate the points of coupling and decoupling in case the agent has to cross a gap. Greedy heuristics or some sort of optimization technique will be used for this purpose. The problem will be formulated as a modification of Multiple Traveling Salesman to include the gaps and coupling behaviors. It outputs the task queues for each agent and sends it to the planning subsystem.

6.3.6 Multi-robot planning

The planning subsystem takes as input the map and the task queues determined by the task allocator for each agent and formulates collision-free paths for each agent. This is done in a sequential manner, making the entire process piecewise-optimal. The planner is based on Conflict Based Search, a high level decision tree to resolve conflicts between agents and re-route them if required. Individual agent plans are found using Theta*, an extension of A* search that generates smoother, shorter paths at the cost of increased computation. This is achieved using line of sight checks with static and dynamic obstacles.

6.3.7 Fleet management system

The fleet management system serves as the communication bridge between all the agents and the server. The user provides a config file with information about all the agents in the fleet to set up communication channels for transmission and reception for all the agents. It also handles packing, unpacking, and logging of data and monitors the agents during operation. It relays information back to all the other subsystems in case of any changes in the fleet. The commands to the individual agents from each subsystem also go through the FMS. Since everything is integrated through ROS, the FMS becomes the direct interface with the robots. The mechanism has also been integrated with the FMS. This enables communication between the host computer and the NodeMCU microcontrollers in the robots' mechanisms.

6.4 Modeling, Analysis and Testing

Test No.	Objective	Procedure	Requirement
1	Build a map of the test arena and localize agents within the map	Run SLAM Toolbox which takes in the robot's wheel odometry and LRF scan messages to build a map, localize agents within the map using VICON.	MF1

2	Plan collision-free trajectories for agents as per their assigned goal locations	Given the map of the environment and localization using the VICON tracker, run the planner to assign collision-free paths for each robot in the swarm.	MF2, MF6
3	Allocate robots respective PoIs optimally and allocate robots for the gap-crossing task whenever required.	Given the map and localization information, run the task allocator to allocate goal points to respective robots.	MF2, MF6
4	Couple three robots and make them cross gaps	Place three robots with the mechanism close to each other. Manually align them to allow for smooth coupling. Cue the coupling routine and gap-crossing routine.	MF5
5	Ensure lossless communication between server and agents	Turn on the robots and connect them to the server through the FMS, observe packet information and rostopic rates.	

6.5 SVD performance evaluation

The system performed very well and met all the requirements set for the spring semester at both SVD and SVD encore. The following table summarizes our test results for different requirements

Success Criteria	Requirements	Unit tests	SVD	Encore
Crossing gaps of length 19cm	MP1, MP3, MP6	10/15	2/2	1/1
80% reliability for coupling mechanism	MP3, MP4	14/15	2/2	1/1
100% POIs reached	MP2, MP5	24/30	3/3	3/4
100% collisions avoided	MP2	29/30	2/3	3/4
POI exploration less than 5 minutes per run	MP2	22/30	3/3	3/4

6.5 Strong and Weak Points

Strengths

- POI coverage is sufficiently reliable
- The system successfully crosses the gaps of desired length
- There are no unplanned collisions between agents and/or obstacles

Weaknesses

- Mechanism is prone to wear and tear, increasing chances of failure
- Localization still relies on the Vicon system
- Planner can take a long time to compute incase of multiple collisions between agents.

7. Project Management

7.1 Work Breakdown Structure

We chose to split the project into 5 key verticals - Hardware, Fleet Management System, Planning, Localization and Management. This is derived from the major blocks in our cyber physical architectures explained above (section ref.).

For Hardware, We have completed the design and development of the coupling mechanism and enclosure with few iterations. The coupling mechanism has been tested thoroughly but has scope of improvement in terms of robustness. The electromechanical system has been integrated and tested with the mechanism. Upcoming semester plans include design improvement and integration with the entire system

For FMS, all the mentioned tasks have been completed, but will need revisiting in the upcoming semester once we add feedback and perception to the communication pipeline

For Planning, task allocator and goal trajectory planner have been developed and tested. configuration determination, coupling/decoupling planner and gap crossing planner are the tasks for the upcoming semester.

For Localization, all the tasks are completed. We will be revisiting all the tasks again, since we will be using different methods of localization for upcoming semester

For all the verticals some tasks will be revisited, in order to address the issues incurred due to entire system integration.

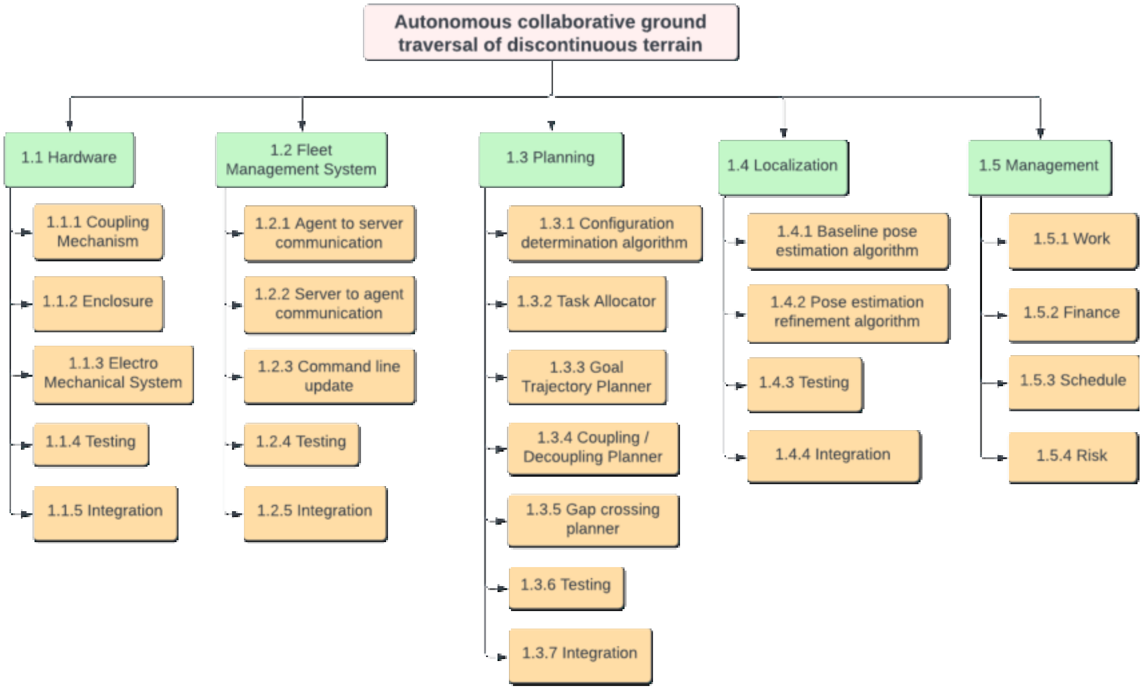


Fig 12: WBS Schematic

7.2 Project Management: Schedule Status

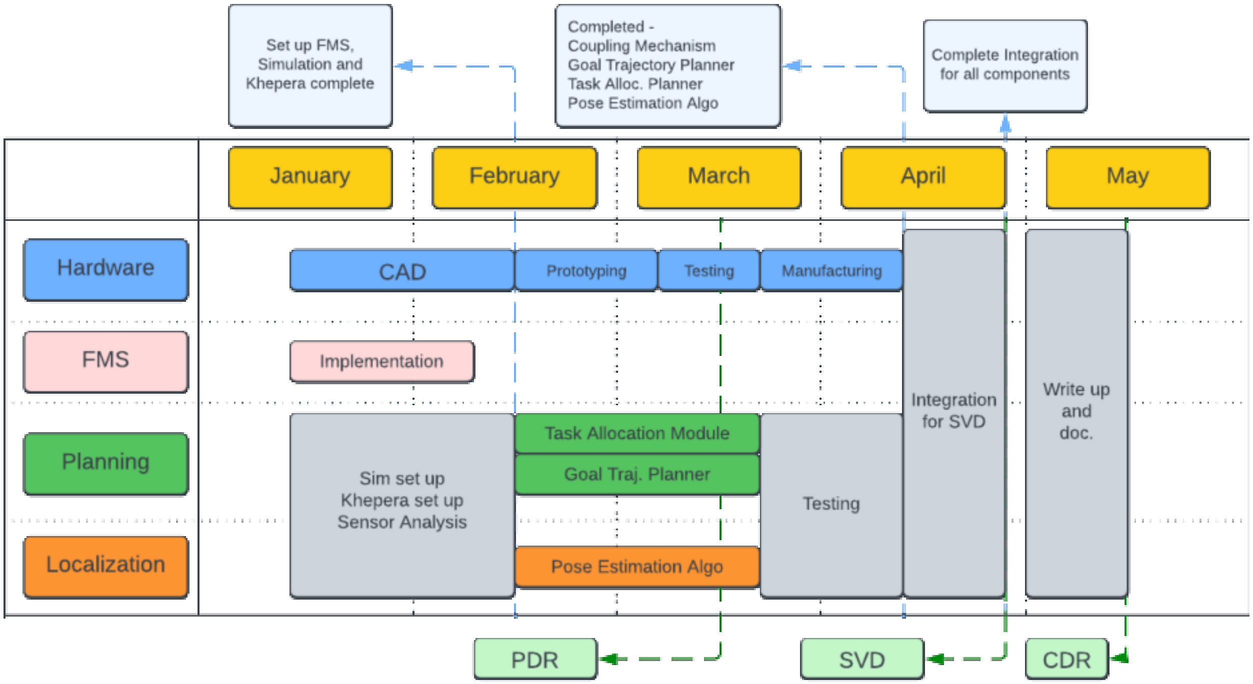
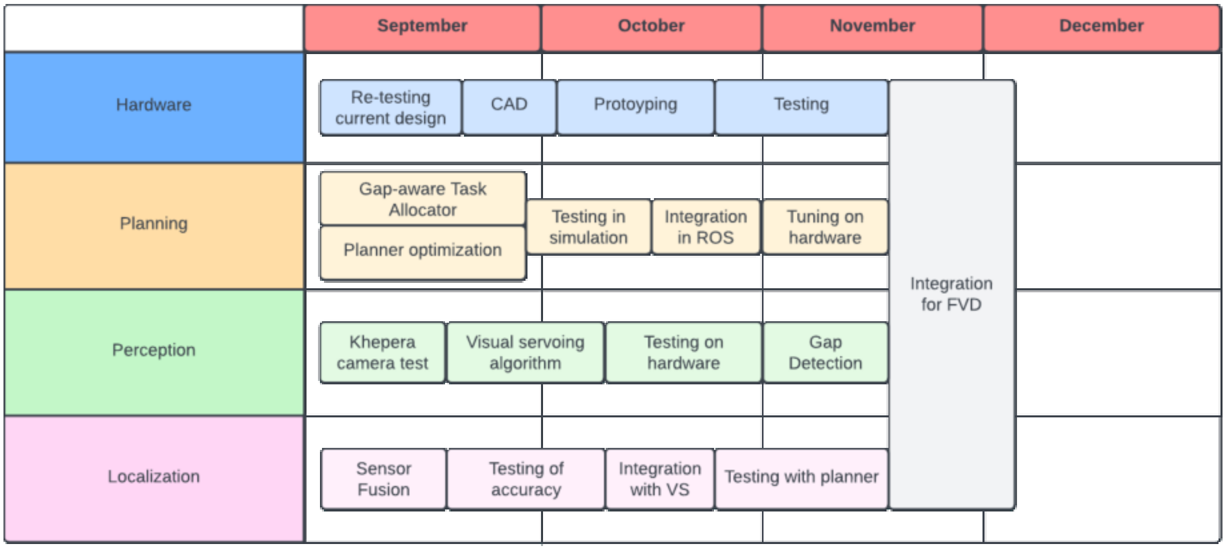


Fig 13: Spring Schedule



	September	October	November	December
Hardware	Re-testing current design CAD	Protoyping	Testing	Integration for FVD
Planning	Gap-aware Task Allocator Planner optimization	Testing in simulation Integration in ROS	Tuning on hardware	
Perception	Khepera camera test Visual servoing algorithm	Testing on hardware	Gap Detection	
Localization	Sensor Fusion Testing of accuracy	Integration with VS	Testing with planner	

Fig 14: Fall Schedule

7.3 Project Management: Fall Test Plans

Date	PR	Capability Milestones	Subsystem
13th Sept	6	● Robust Coupling Mechanism with improved reliability ● MPC controller testing ● Khepera Camera Testing with Aruco Tags ● LRF + encoders vs Vicon	All Subsystems
27th Sept	7	● First prototype of redesigned mechanism ● Visual Servoing first iteration ● Gap-aware task allocation in simulation ● Planner integrated with new localization	All Subsystems
11th Oct	8	● Visual servoing final testing ● Integration of task allocator, planner and controller in simulation ● Manufacturing of mechanisms	All Subsystems
1st Nov	9	● Integrating perception pipeline with the navigation stack on physical robots ● Finalize demonstration concepts	All Subsystems
15th Nov	10	● Full system testing on various configurations ● Edge case documentation ● Variable gap-width test	All Subsystems

7.4 Project management: Parts list and Budget

Part No.	Quantity	Part Name	Unit Price	Total Price
T0601-50	3	Linear Rail 50mm / 100mm / 150mm/ 200mm Linear Stage Actuator with Square Linear Rails Mini Slide Table + NEMA 11 Stepper Motor for DIY CNC Router Milling Machine (50mm)	$62	$186
TS-04-09	2000mm	drylin® T, miniature profile rail	164.56	$164.56
TW-04-09-LLY	8	drylin® T, miniature slide carriage	15.7	$125.60

a19042500ux0042	3	uxcell DC 5V 1A 30g 3mm Mini Electromagnetic Solenoid Lock Pull Type for Electric Lock Cabinet Door Lock	11.99	35.97
SRRM0142	4	Pcchips One Channel Relay 5V Module low Level Trigger - Blue | SRRM0142	5.75	23
GLLATHM-340	2	Glarks 370Pcs M2 M3 M4 M5 Female Thread Knurled Brass Threaded Insert Embedment Nut Assortment Kit for 3D Printing	17.92	35.84
590Metric bearing ball	1	Breezliy 590Pcs 1-8mm Metric Precision 304 Stainless Steel Assorted Loose Bicycle Bearing Steel Ball Assortment Kit（12 Sizes）	11.88	11.88
MR104-2RS	4 set (40 pcs, 10 pcs each set)	Donepart Small Bearings 4mm x10mm x4mm MR104 2RS Minature Ball Bearings Double Metal Shielded for Mini Motors, Fidget Spinners, Industrial Machinery, etc (10 Pack)	10.98	43.92
M4-0.7	1	binifiMux 100pcs Pan Phillips Head M4 Machine Screws Hex Nuts 304 Stainless Steel Assortment Kit, M4x60mm/ 65mm/ 70mm/ 75mm/ 80mm	12.99	12.99
M5-3050H-12.9	1	iexcell 50 Pcs M4 x 30/35/40/45/50 Alloy Steel 12.9 Grade Hex Socket Head Cap Screws Bolts Assortment Kit, Black Oxide Finish	8.98	8.98
A15071700ux0425	1	uxcell M2x30mm Hex Socket Head Knurled Cap Screws Bolts Nuts Set 20Pcs	8.49	8.49

ESP8266MOD	3 (Pack of 3)	HiLetgo 3pcs ESP8266 NodeMCU CP2102 ESP-12E Development Board Open Source Serial Module Works Great for Arduino IDE/Micropython (Large)	16.39	32.78
	1	Neoprene Rubber Sheet - 1/16 Inch Thick x 12 Inch Wide x 4 Feet Long Neoprene Rubber Strips Rolls for DIY Gaskets, Pads, Seals, Crafts, Flooring, Cushioning of Anti-Vibration, Anti-Slip	20.59	20.59
LM2596	3 (9 pcs in total)	Valefod 3 Pack LM2596 DC to DC Voltage Regulator 4-40V to 1.5-35V Buck Converter with LED Display	13.99	41.97
39122300	1	3v Relay Board Raspberry Pi Arduino Relay Module 1 Channel Opto-Isolate High Level Trigger for IOT ESP8266 Microcontrollers Development Board	14.99	14.99
754-1162-2-ND	5	APTL3216CGCK	0.41	2.05
WK6286BK-ND	4	Littelfuse Inc. 40013150440	1.52	6.08
541-RCC120620R0FKEATR-ND	2	Vishay Dale RCC120620R0FKEA	0.39	0.78
541-RCC1206200RFKEATR-ND	2	Vishay Dale RCC1206200RFKEA	0.39	0.78
541-RCC120622R0FKEATR-ND	2	Vishay Dale RCC120622R0FKEA	0.39	0.78
538-39-28-1023	5	Molex 39-28-1023	0.65	3.25
MIC29300-12WU-TR-ND	2	Microchip Technology MIC29300-12WU-TR	4.72	9.44
576-1122-ND	2	Microchip Technology MIC29300-5.0WU	4.96	9.92
P4SMA16CATR-N	3	Littelfuse Inc. P4SMA16CA	0.43	1.29
497-2999-2-ND	3	STMicroelectronics SMAJ5.0A-TR	0.44	1.32

SMAJ26A-E3/61 GITR-ND	3	Vishay General Semiconductor - Diodes Division SMAJ26A-E3/61	0.45	1.35
311-1993-2-ND	3	YAGEO CC1206MKX5R5BB476	0.51	1.53
PMK432C6477 MM-T	6	Taiyo Yuden PMK432C6477MM-T	9.29	55.74
1738-1144-ND	8	DFRobot DFR0123	7.5	60
11.1V 120C 1500mAh	1	Zeee 11.1V 120C 1500mAh 3S RC Lipo Battery Graphene Battery with XT60 Plug for FPV Racing Drone Quadcopter Helicopter Airplane RC Boat RC Car RC Models(2 Pack)	33.99	33.99
6000mAh 120C 6S Lipo Battery	1	6S Lipos 22.2V 120C 6000mAh Lipo Battery Pack with EC5 Connector RC Battery for RC Car/Truck/RC Plane/Helicopter/Quadcopter/ Boat/Arrma Mojave/Krajave EXB/Felony/ARRMA Typhon 6S V5/Avanti Jet(2PCS)	152	152
EC5-001	1	NeeKare 10 Pairs EC5 Battery Connector Plugs Goldby EC5 Male Female 5.0mm Banana Plug Connectors Gold Bullet Connector for RC ESC LIPO Battery Device Electric Motor	12.99	12.99
105.082.20	2	LAGKAPTENTabletop, white, 63x31 1/2 "	89	178
C150	1	LiPo Charger Lipo Battery Balance Charger RC Charger RC Car Battery Charger Discharger 150W 10A 1-6S AC/DC for Li-ion/Life/NiCd/NiMH/LiHV /PB/Smart Battery(Battery Charger Adapter)	56.99	56.99

BG001US	2	Zeee Lipo Safe Bag Fireproof Explosionproof Bag Large Capacity Lipo Battery Storage Guard Safe Pouch for Charge & Storage(8.46 x 6.5 x 5.71 in)	13.49	26.98
T0601-50	3	Linear Rail 50mm / 100mm / 150mm/ 200mm Linear Stage Actuator with Square Linear Rails Mini Slide Table + NEMA 11 Stepper Motor for DIY CNC Router Milling Machine (50mm)	$62	$186
a19042500ux0042	6	uxcell DC 5V 1A 30g 3mm Mini Electromagnetic Solenoid Lock Pull Type for Electric Lock Cabinet Door Lock	11.99	71.94
Zeee 3S	4	Zeee 3S Lipo Battery 5200mAh 50C 11.1V RC Batteries with XT60 Connector Soft Case for RC Airplane Helicopter Plane Quadcopter RC Car Truck Boat	31.49	125.96
3M VHB	2	Double Sided Tape Heavy Duty (2 Pack), Waterproof Strong Mounting Adhesive Tape for Walls, Car, Home Decor, Office Decor, Made of 3M VHB Tape (16.4FT x 0.94IN)	33.99	67.98
MR104-2RS	4 set (40 pcs, 10 pcs each set)	Donepart Small Bearings 4mm x10mm x4mm MR104 2RS Minature Ball Bearings Double Metal Shielded for Mini Motors, Fidget Spinners, Industrial Machinery, etc (10 Pack)	10.98	43.92
			Total	**$1,805**

Table 13: BOM Budget Requirement

7.5 Risk Management

Table 14: Key Risks and Mitigation Plans

Sr No	Risk Description	Likelihood	Consequence	Mitigation Plan
1.	ESP + NodeMCU is unreliable in communications and controlling linear actuators	4	5	● Test with alternatives like Arduino Nano with current shield ● Created detailed SOP to avoid random resets and resolve latency issues
2.	Enclosure and pins break easily during testing	4	5	● Implement design improvements in the pin and lock mechanism to address common points of failure ● Fabricate pins with aluminum to improve rigidity
3.	Kheperas get stuck on small bumps and edges during the demo	3	3	● Level out and inspect the test surfaces thoroughly before finalizing setup ● Update the map to reflect the regions as obstacles
4.	Unavailability of testing area	2	4	● Identify well in advance areas that are suitable for our full demo ● Book the venues well ahead of time and conduct multiple dry runs to ensure smooth demo experience
5.	Team Member(s) fall sick	1	2	● Keep documentation of work up-to-date ● Distribute work such that two members are involved in any critical task ● Rest days to prevent burnout and stress

8. Conclusions

8.1 Lessons Learnt

Through the course of the semester working on our system, we had quite a few challenges that we needed to overcome and some valuable learnings, both in the technical aspects as well as in managing project progress, stakeholders and conducting a smooth demonstration.

1. Test early, fail early - our integration went quite smoothly because we had tested each subsystem early in development and documented the known issues well.
2. Clear communication with stakeholders - there were multiple times where we were facing issues in hitting our performance requirements due to unforeseen technical issues in the platforms. Keeping a clear line of communication with our sponsors made sure that all decision were transparent
3. Too much testing can backfire - since we had a significant electromechanical component in our project, the wear and tear due to repeated testing meant that our system lost reliability closer to the demo, necessitating last minute problems in integration

8.2 Key Fall Activities

For the Fall semester, we plan on building on the work done throughout Spring and integrate all the subsystems to enable full end-to-end autonomous coverage of discontinuous maps. To achieve this, we have the following key activities lined up

1. Perception-based relative alignment of agents for coupling - this is necessary since our mechanism has a low tolerance to misalignment. We plan on using fiducial markers and visual servoing for agents to form configurations autonomously.
2. Task Allocation Improvements - our current task allocation is a relatively unmodified Vehicle Routing Problem. Once gaps are introduced into the mix, the system will need to account for the gap crossing task and assign multiple agents to it in an optimal manner
3. Controller and Planner improvements - to improve the tracking of waypoints and smoothen out the motion, we plan on defining primitives and implementing our planner on a state lattice of unit-time motions rather than cartesian space. In accordance to this, the underlying controller will be shifted to MPC for smoother motions and better time synchronization across multiple agents
4. LRF-based localization - on moving away from Vicon tracking, we anticipate issues in LRF and encoder based localization for all downstream processes since we rely on accurate position information for planning, controls and task allocation.

9. References

1. Sha Yi, Zeynep Temel, and Katia Sycara. “PuzzleBots: Physical Coupling of Robot Swarm.” IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2021.
2. Sha Yi, Zeynep Temel, and Katia Sycara. “Configuration Control for Physical Coupling of Heterogeneous Robot Swarms.” IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2022.
3. Saab, W., Racioppo, P., & Ben-Tzvi, P. (2019). A review of coupling mechanism designs for modular reconfigurable robots. doi:10.1017/S0263574718001066
4. Khepera IV manual and datasheet https://www.k-team.com/khepera-iv
5. Full project schedule for spring: https://app.teamgantt.com/projects/gantt?ids=3327849