Preliminary Design Review

DRAGON

Deployed RF Antennas for GPS-denied Optimization and Environmental Navigation

Dawson Beatty, Christian Carmack, Jeremy Fie, Chris Greer, Ross Kloetzel, Jack Maydan, Kyle Nieukirk, Virginia Nystrom, Amanda Siirola, Ryan Stewart, Luke Tafur, Ivan Yurkin

Aerospace Engineering

COHRINT

Project Overview

Project Overview

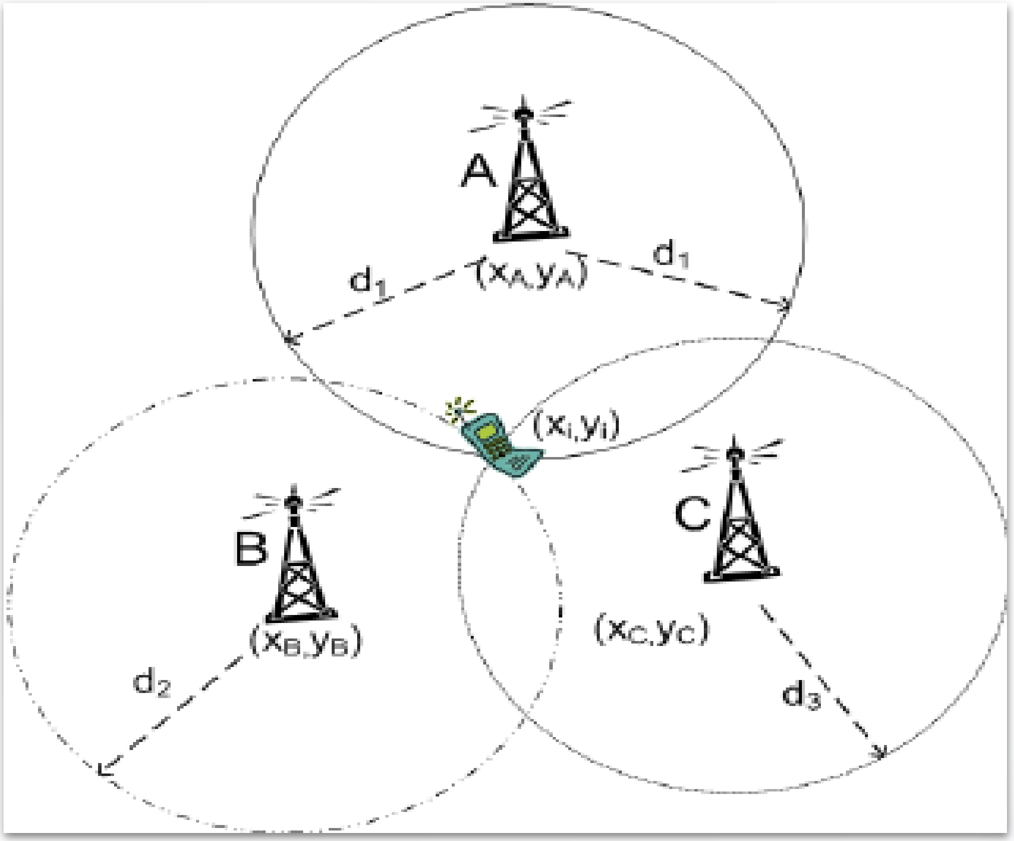
Baseline Design

Feasibility Studies

Feasibility Summary

Project Motivation

- The field of autonomous navigation is prevalent in a variety of environments, from Martian deserts to battlefield urban canyons; these systems typically rely on GPS. However, in these environments the availability of GPS is never guaranteed.
- Presently, GPS denied autonomous navigation is limited to small scale, high error, inertial dead-reckoning measurements. The Dragon team is pursuing a solution using RF-Localization to trilaterate location for inertial error correction.



Problem Statement

The DRAGON team will provide a fully autonomous method to **<u>improve unmanned rover navigation in GPS-denied environments.</u>**

This will be done by:

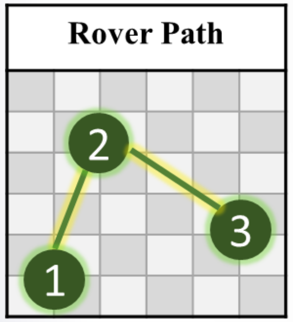
- Creating pods which contain RF-Localization beacons.
- The pods will be accurately deployed to software-determined locations via a ranged-deployment mechanism.
- The pods (and the beacons within them) will act like an in-situ GPS network which the rover can determine its relative position from to correct navigational error.

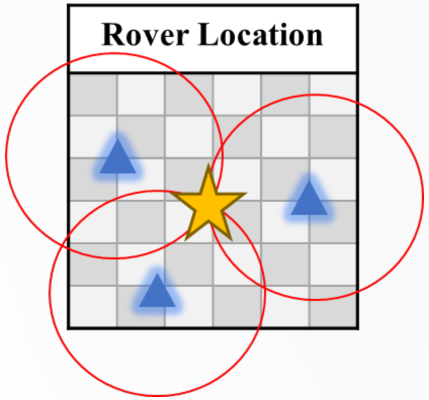
As the pods will remain in the environment permanently, and can access areas the rover cannot, they will also have the *demonstrational ability* to collect and transmit 'environmental' data.

Critical Project Elements

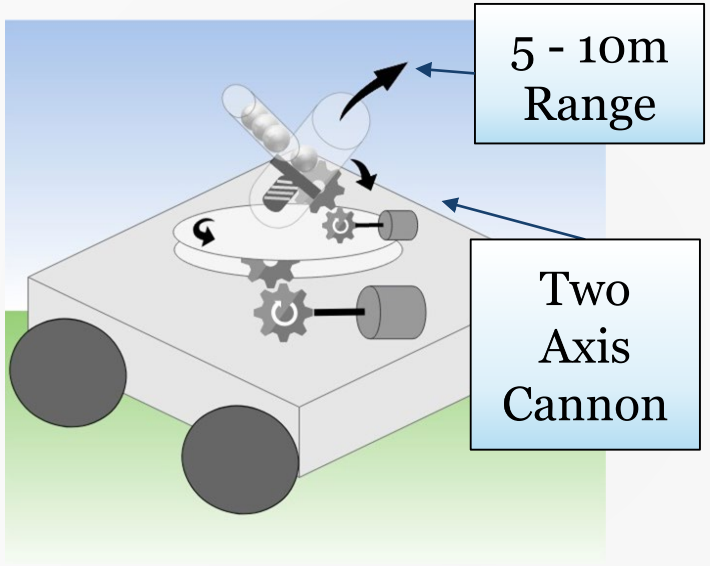
Software:

ROS
Robot Operating System

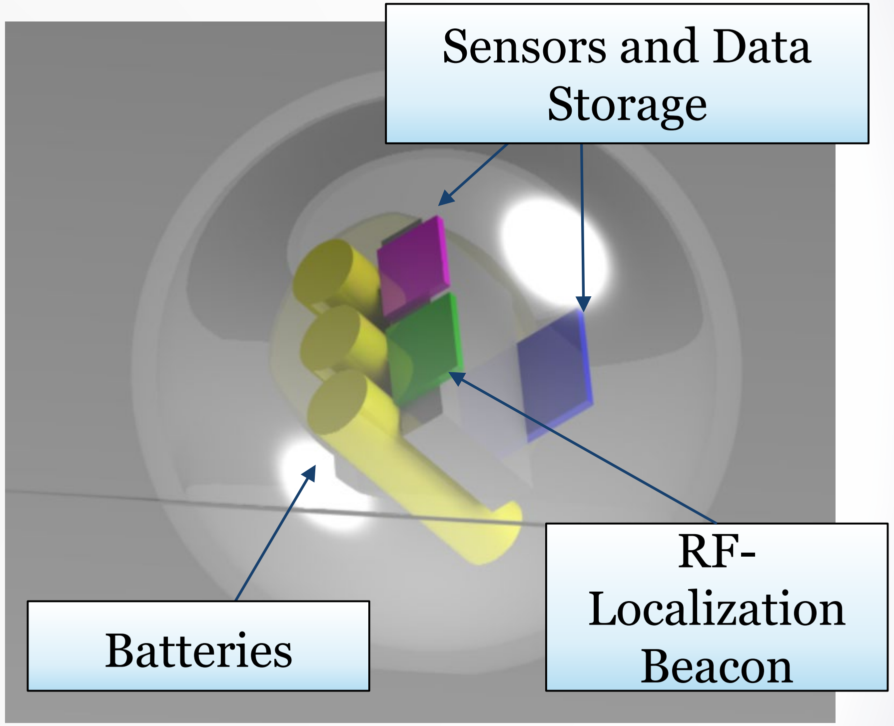




Deployment:

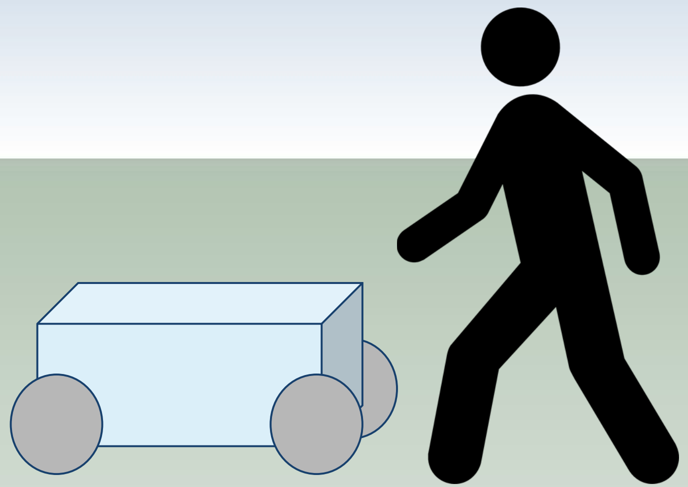


Pods:



- Generate path between waypoints
- Estimate rover position using ranging data
- Deploy pods from rover
- Get ranging data from RF beacons
- Power pods for 2 Hours

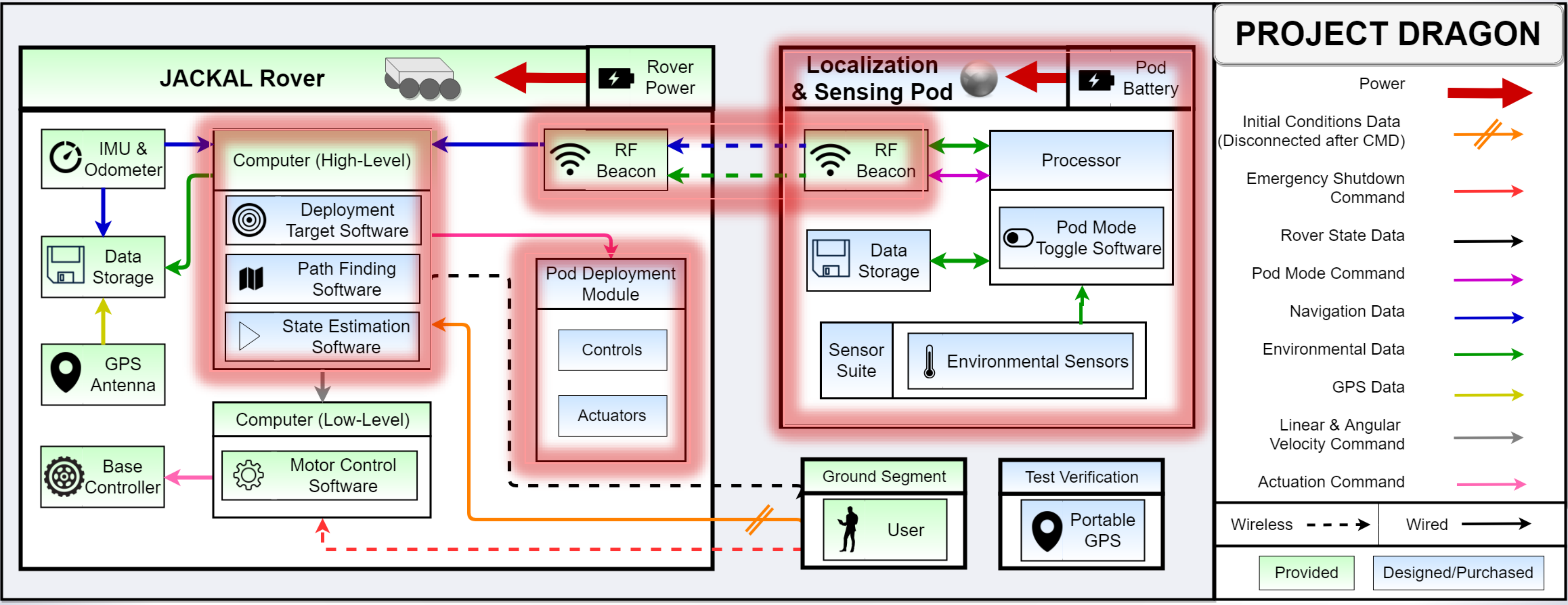
Concept of Operations



Functional Requirements

ID	Description
M1	**Rover shall autonomously navigate along software generated path within 1m accuracy using RF-Localization Beacon correction to inertial navigation**
M2	**The rover shall estimate its absolute position**
M3	**The deployment mech shall have capability to deploy pods to software defined locations**
M4	*The rover and ground inputs shall prevent damage to all hardware systems*
M5	*The pods shall function as RF navigation beacons and as environmental data monitors, to the rover*
M6	*The pods shall be able to function as a long-term deployable environmental data monitor*
M7	*The team shall verify absolute navigation ability*
M8	*The team shall use the customer-provided hardware*

FBD


JACKAL Rover
Rover Power
Localization & Sensing Pod
Pod Battery
IMU & Odometer
Computer (High-Level)
RF Beacon
RF Beacon
Processor
Data Storage
Deployment Target Software
Path Finding Software
State Estimation Software
Pod Deployment Module
Controls
Actuators
Pod Mode Toggle Software
Data Storage
GPS Antenna
Sensor Suite
Environmental Sensors
Computer (Low-Level)
Base Controller
Motor Control Software
Ground Segment
User
Test Verification
Portable GPS
PROJECT DRAGON
Power
Initial Conditions Data (Disconnected after CMD)
Emergency Shutdown Command
Rover State Data
Pod Mode Command
Navigation Data
Environmental Data
GPS Data
Linear & Angular Velocity Command
Actuation Command
Wireless
Wired
Provided
Designed/Purchased

Baseline Design

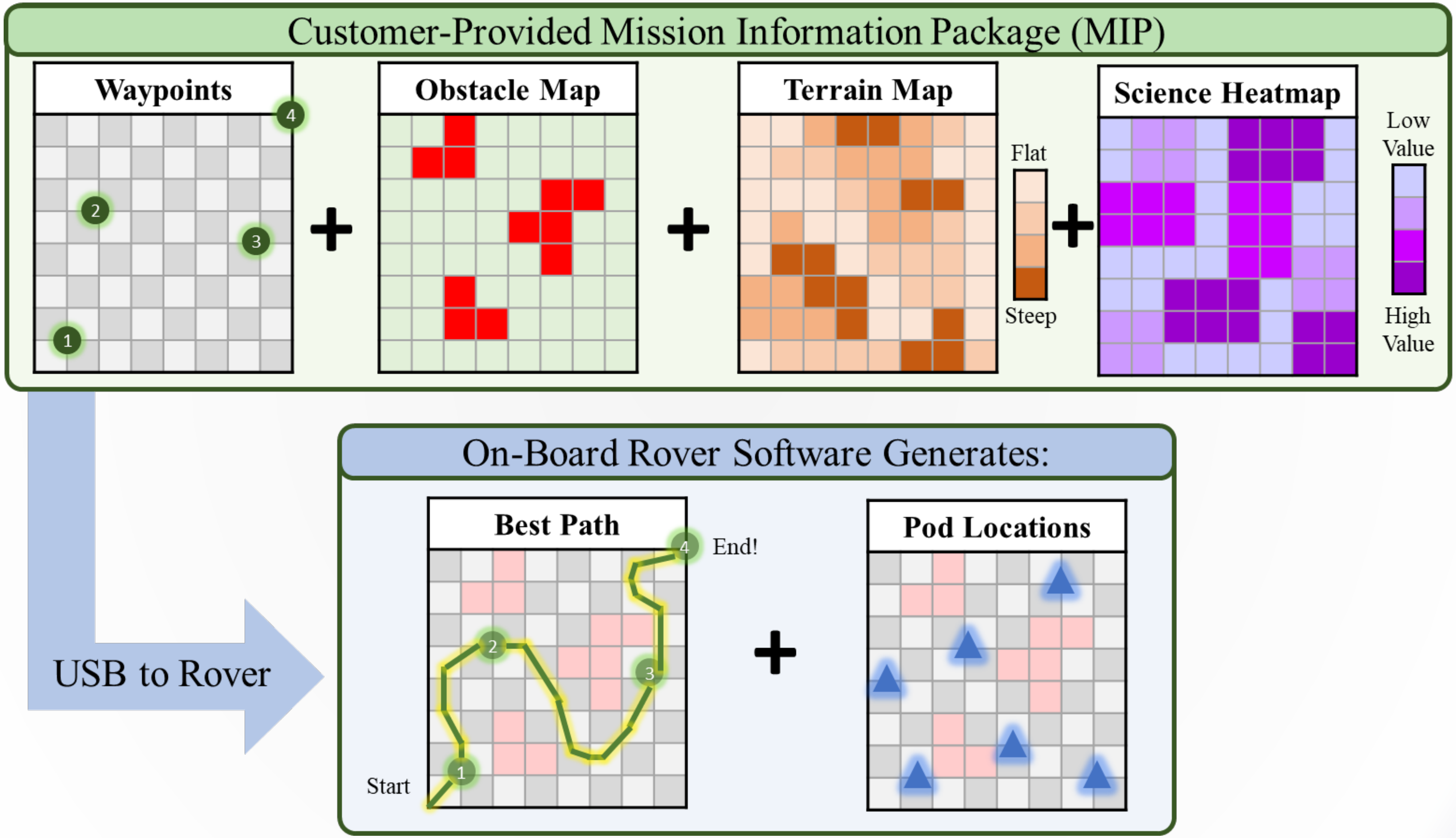
Project Overview

Baseline Design

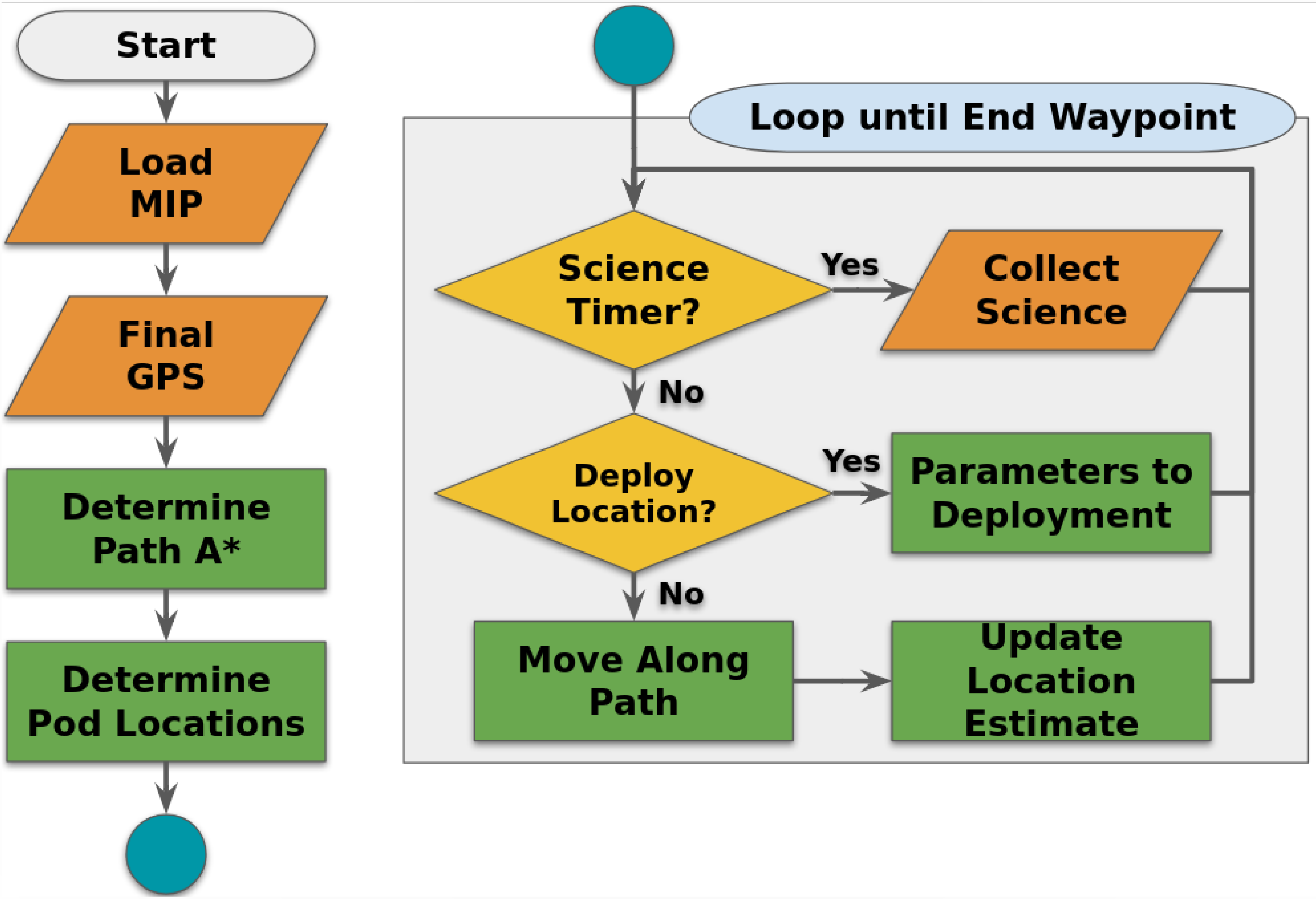
Feasibility Studies

Feasibility Summary

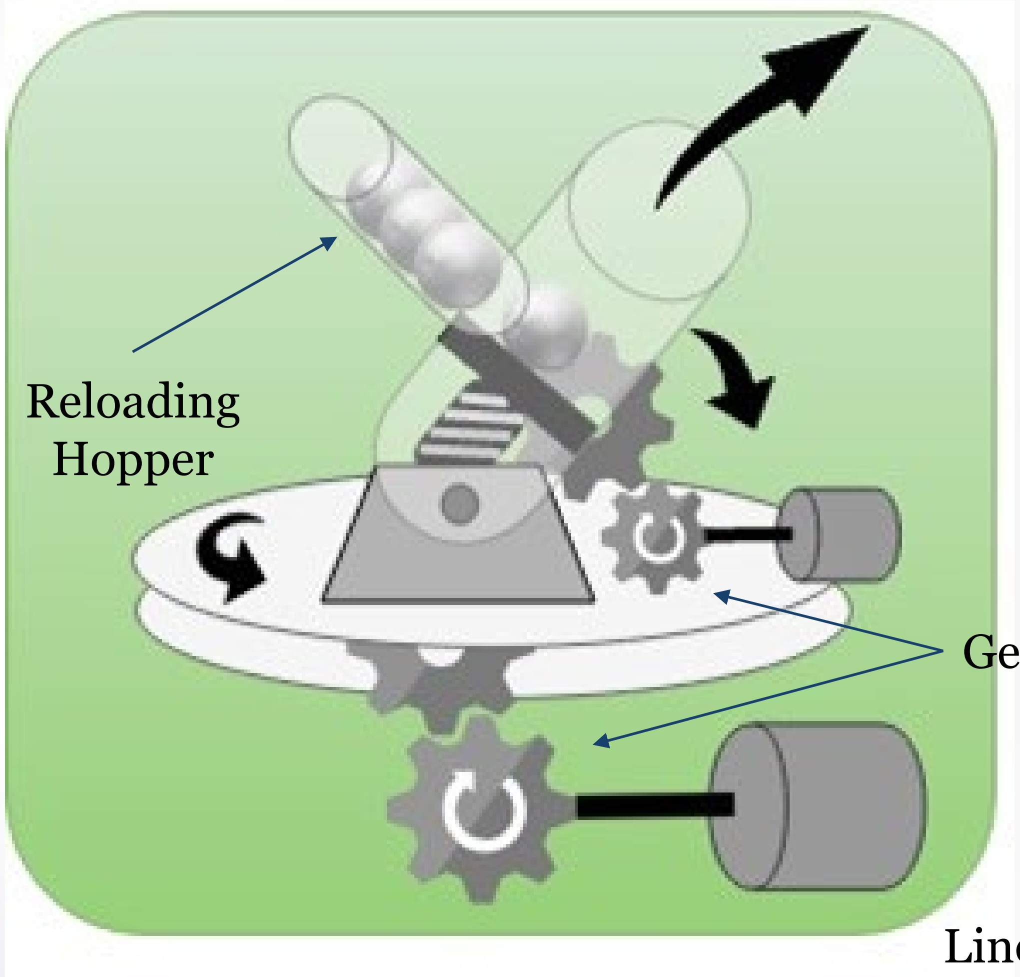
Software Inputs/Outputs

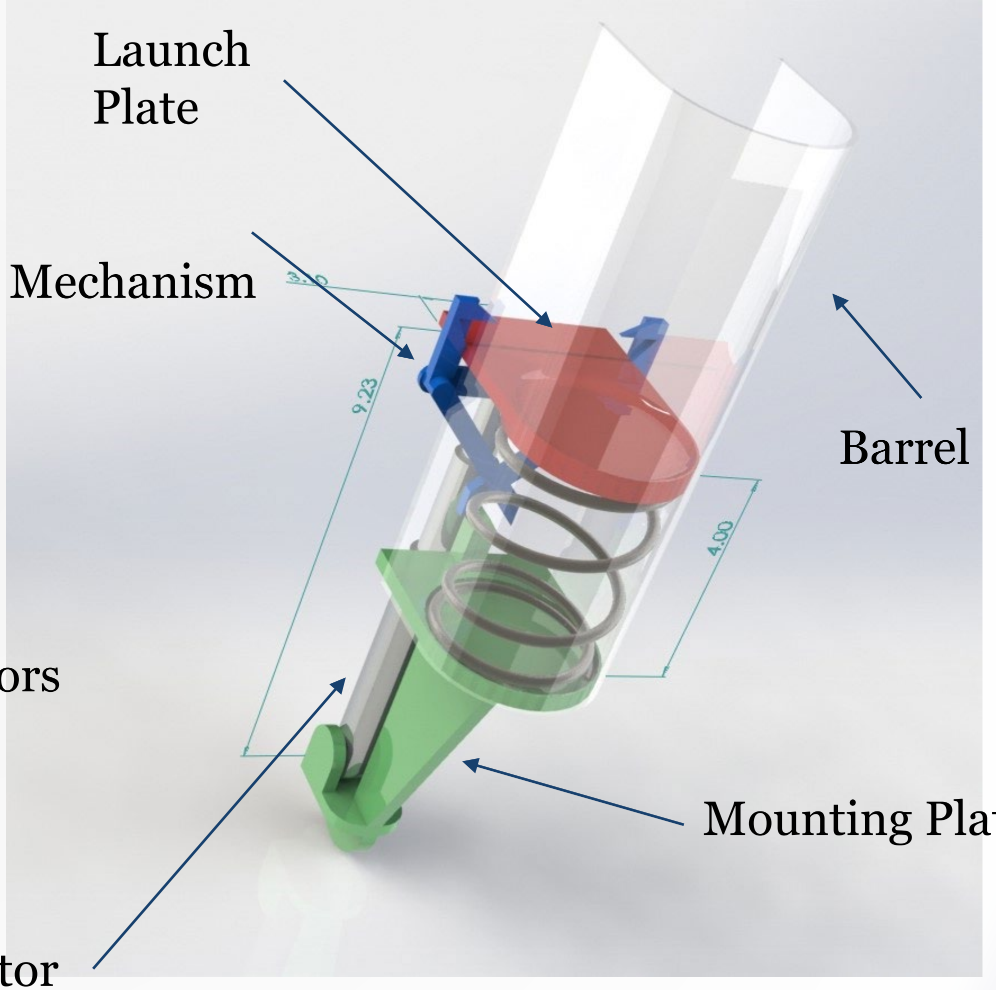


Navigation Software

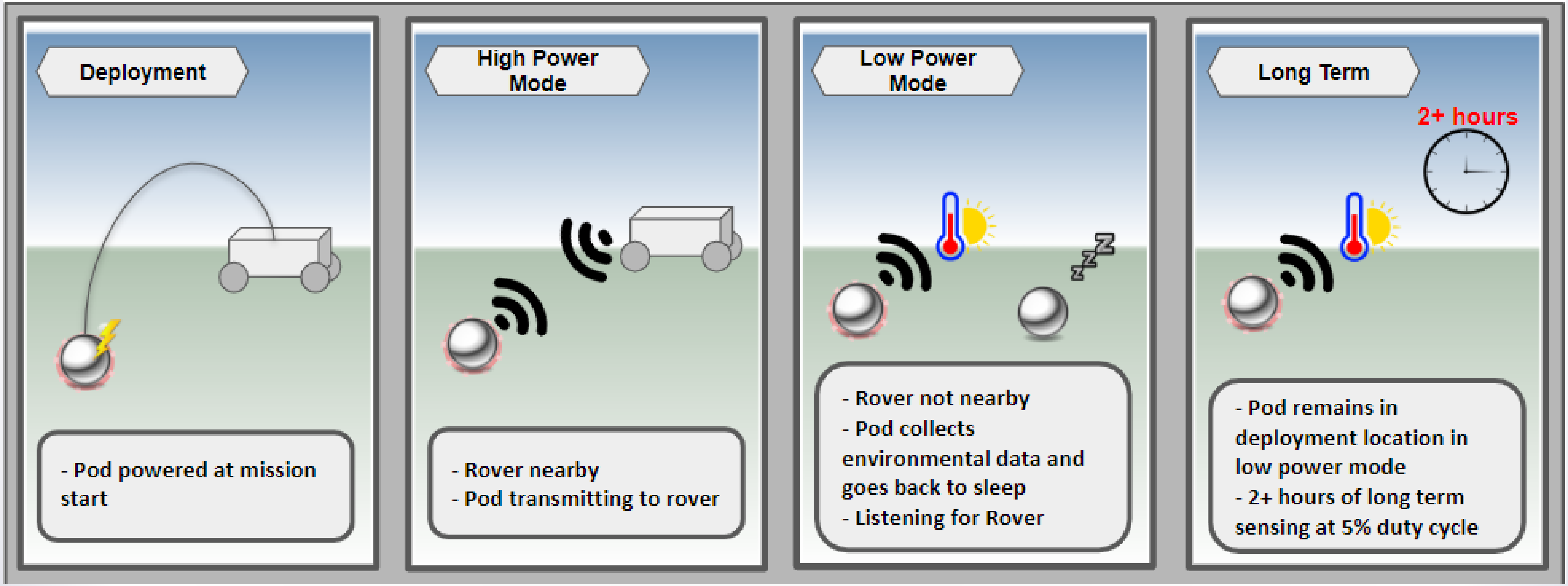


Deployment Structure

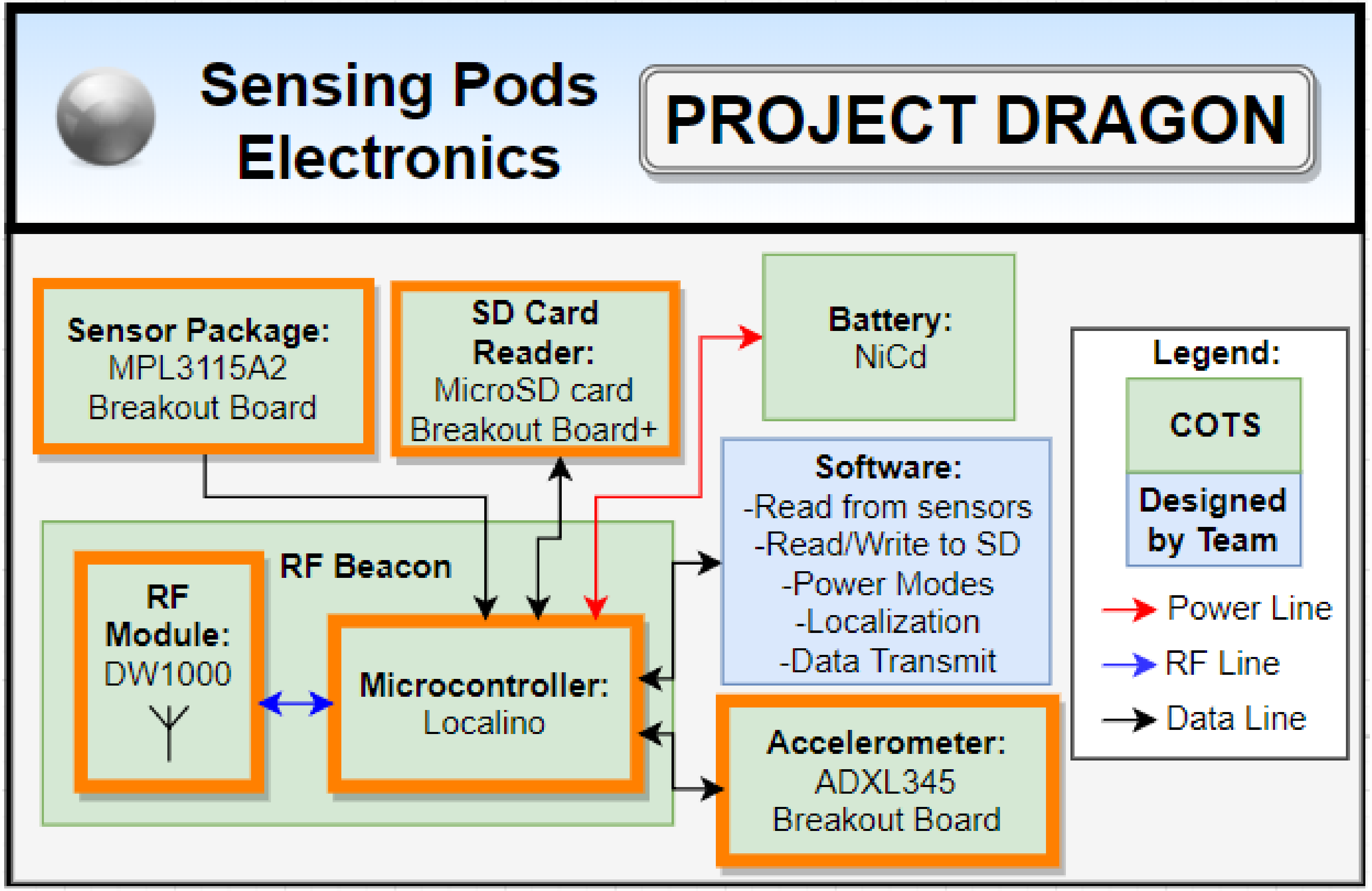




Pod Software



Pod Electronics

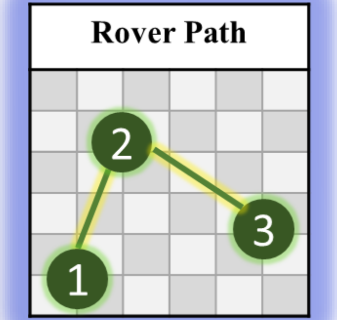


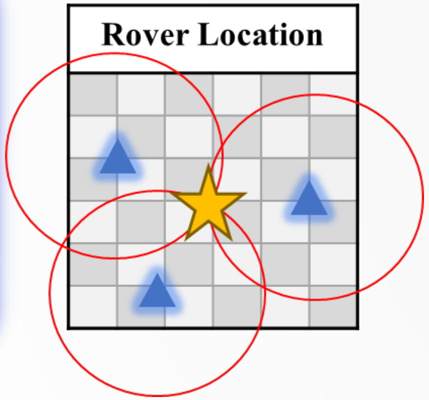
Feasibility Studies

Critical Project Elements

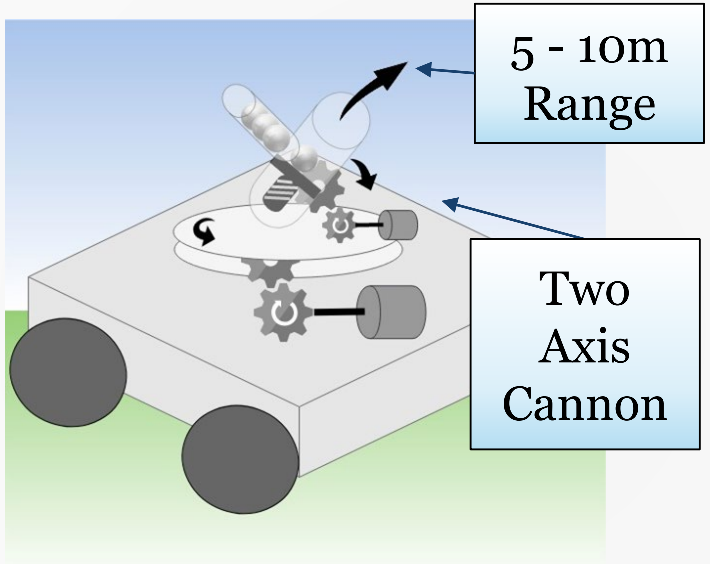
Software:

ROS
Robot Operating System

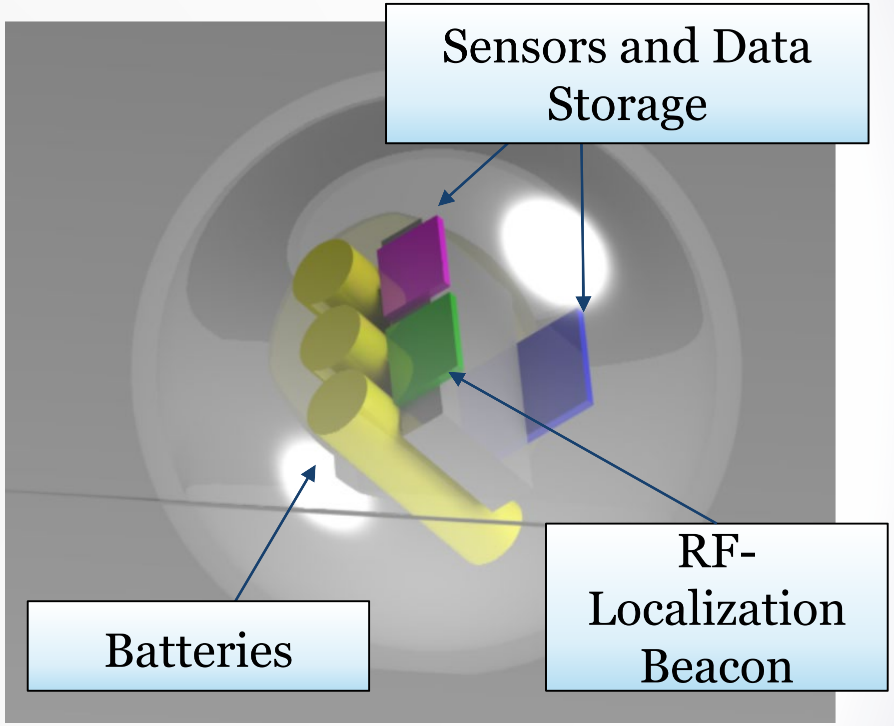




Deployment:



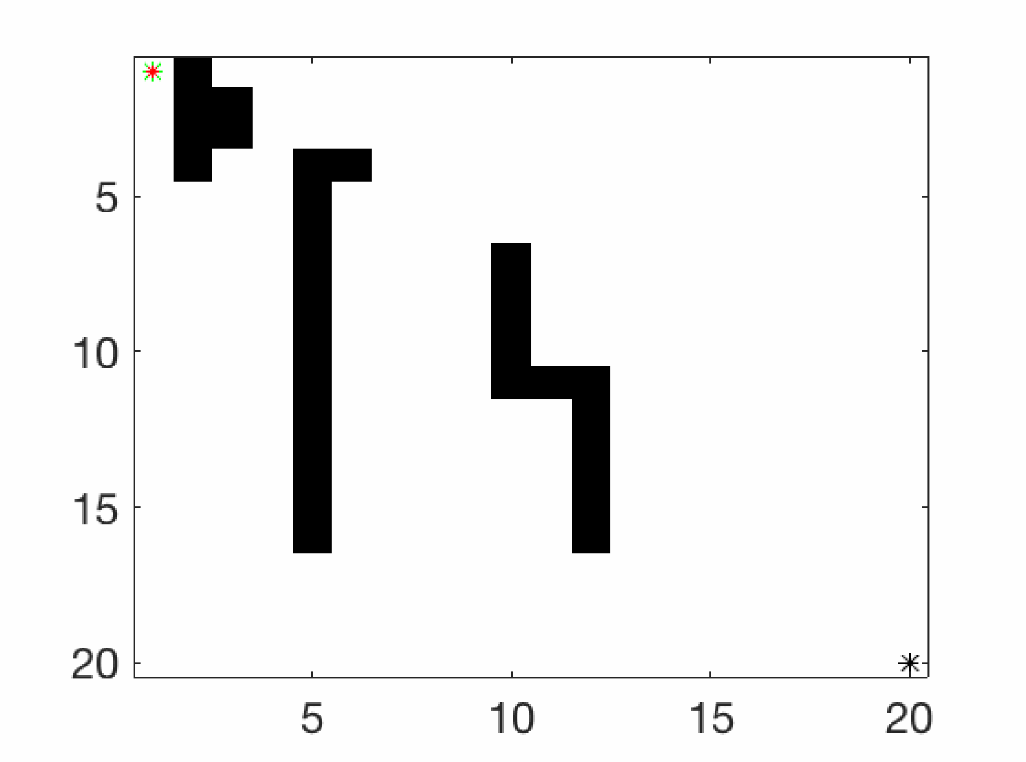
Pods:



- Generate path between waypoints
- Estimate rover position using ranging data
- Deploy pods from rover
- Get ranging data from RF beacons
- Power pods for 2 Hours

Path Generation - Introduction

- Mission requirement involves generating path between all 10 waypoints
- A* finds shortest path between two nodes and allows for obstacle avoidance
- MIP will be received as set of discretized maps and A* will generate path between graph nodes
- A* will be retrofitted to allow for multiple way points instead of start to end only



20x20

Path Generation – Feasibility Approach

Requirements:

- S1.3 - Software shall generate a path through the terrain to reach up to 10 waypoints

Design Approach:

- Use high level MATLAB simulation just to show capability to generate shortest path from start to end while avoiding objects

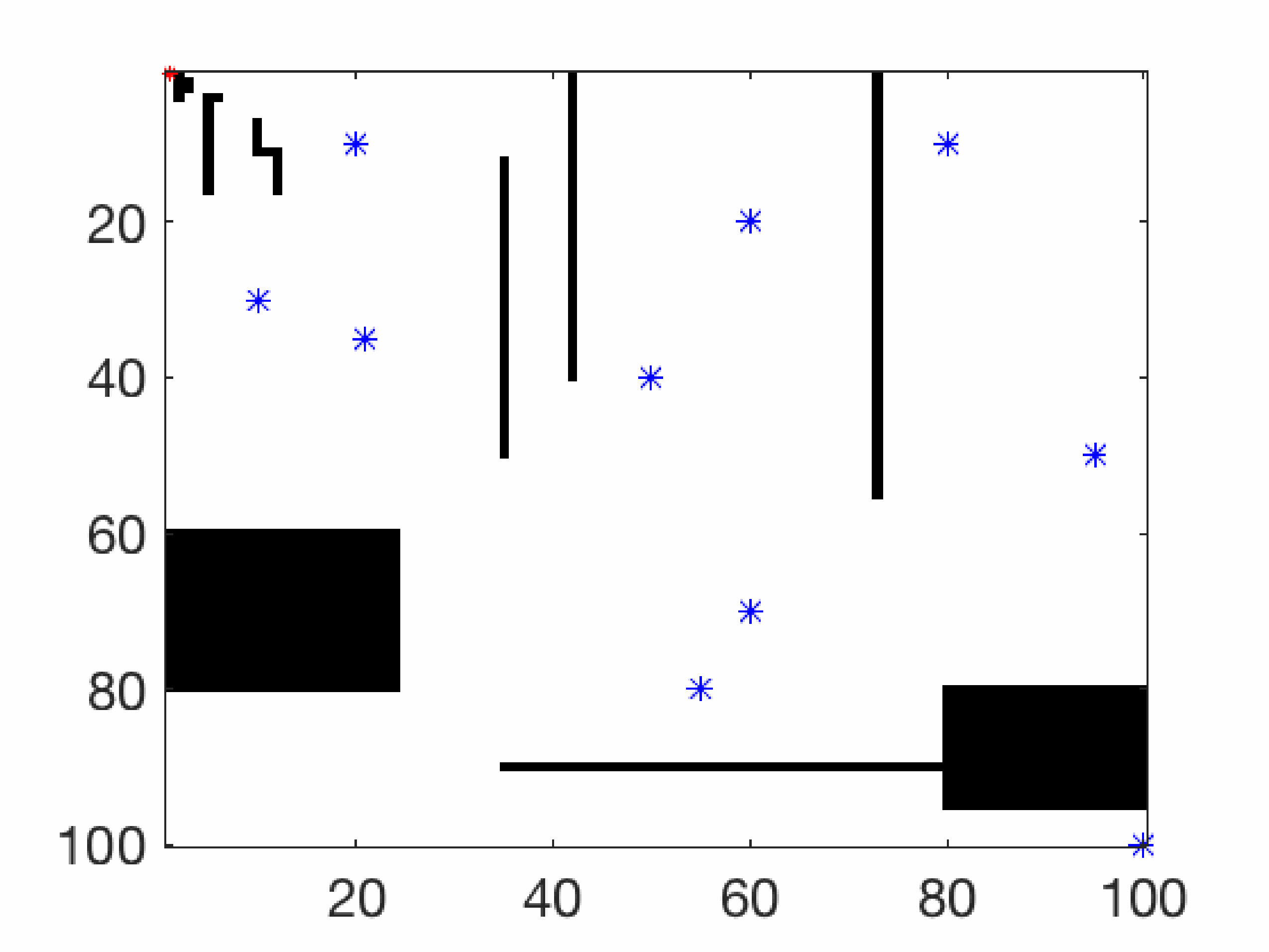
Rationale:

- It is important to show that the rover is capable of generating the shortest safe path from start to the final point of the mission. This study is only used to show team can develop a path to follow from waypoint to waypoint (no quantitative requirement)

Path Generation – Conclusion

S1.3 - Software shall generate a path through the terrain to reach up to 10 waypoints ✓

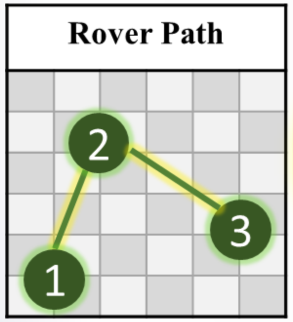
- Proof we are capable of generating shortest path from arbitrary start and end location
- Obstacle avoidance will be set at 1 m to be consistent with path uncertainty
- Algorithm heritage exists and has been previously implemented on Jackal

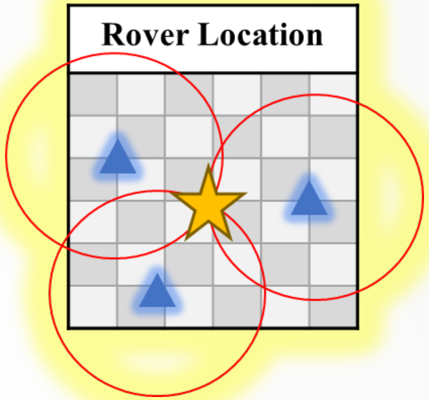


Critical Project Elements

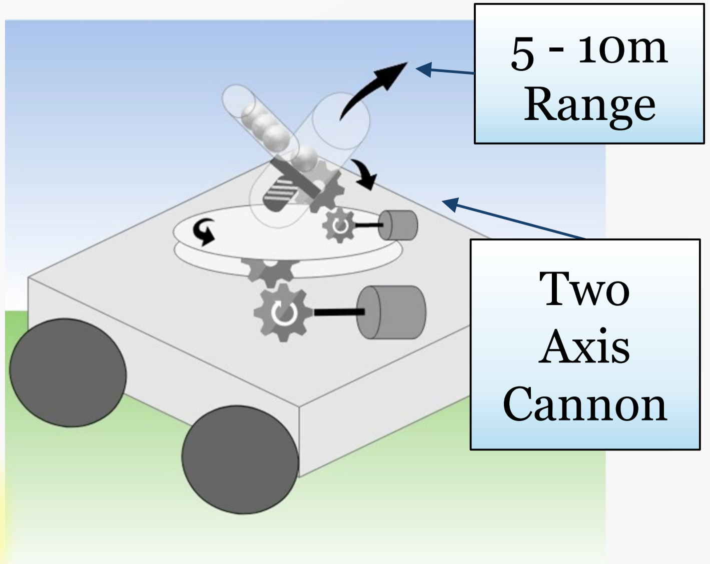
Software:

Robot Operating System

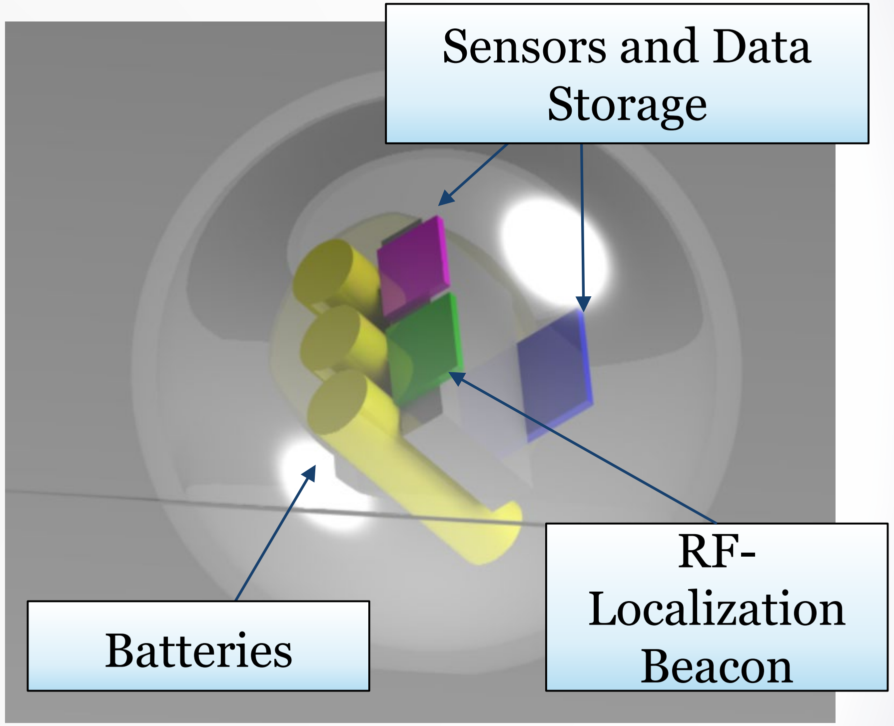




Deployment:



Pods:



✓ Generate path between waypoints

Estimate rover position using ranging data

Deploy pods from rover

Get ranging data from RF beacons

Power pods for 2 Hours

Path Generation – Feasibility Approach

Requirements:

- M1 - Rover shall autonomously navigate along software generated path within 1m accuracy using RF-Localization Beacon correction to inertial navigation

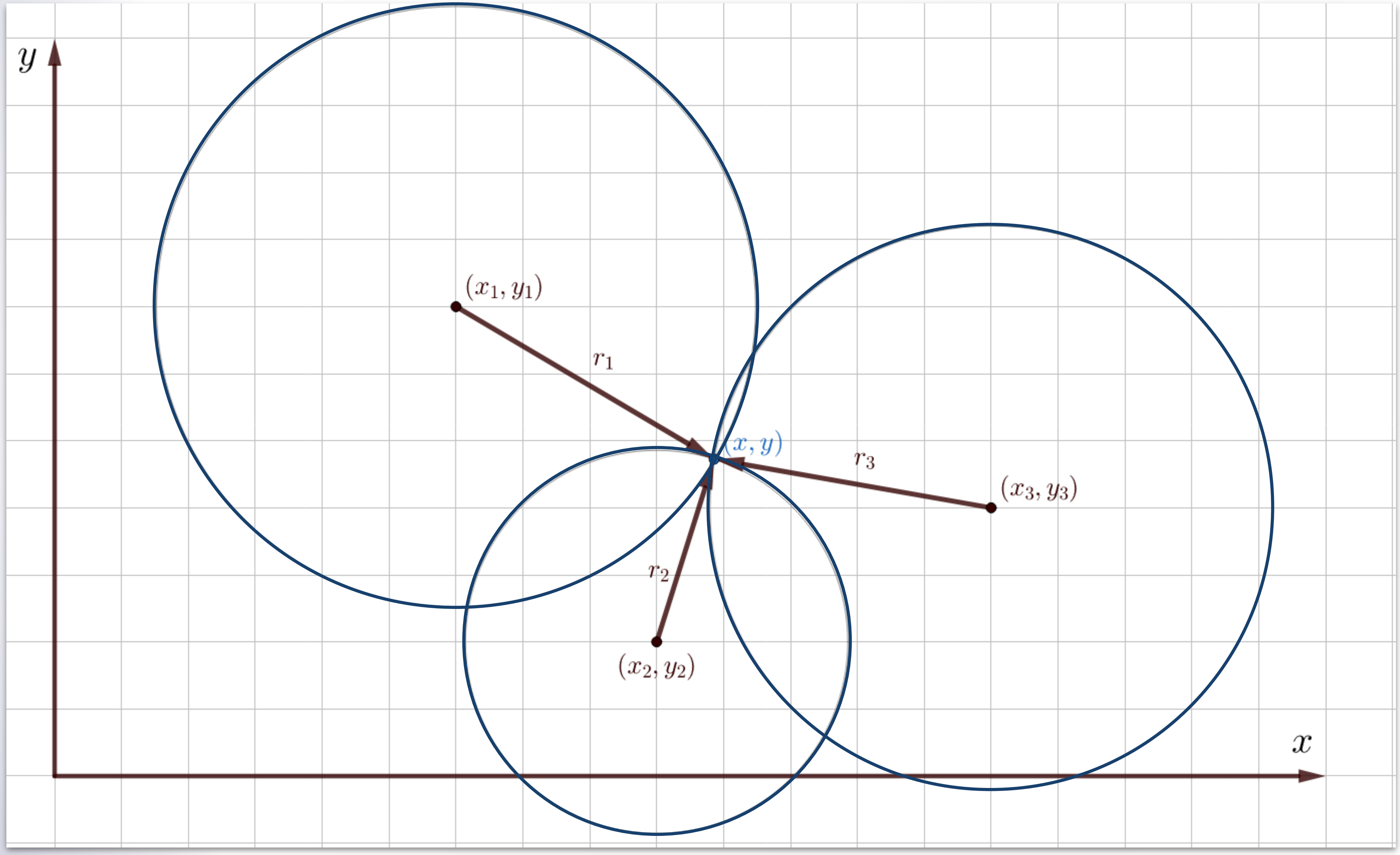
Design Approach:

- Uncertainty can be bounded by trilaterating location using deployed pods if pod locations are selected to not incur large dilution of precision

Rationale:

- Without external correction, onboard IMU error grows without bounds; very inaccurate over long distances

Trilateration Mathematics



$$(x - x_1)^2 + (y - y_1)^2 = r_1^2$$
$$(x - x_2)^2 + (y - y_2)^2 = r_2^2$$
$$(x - x_3)^2 + (y - y_3)^2 = r_3^2$$

Solve system of equations to determine position *(x, y)*

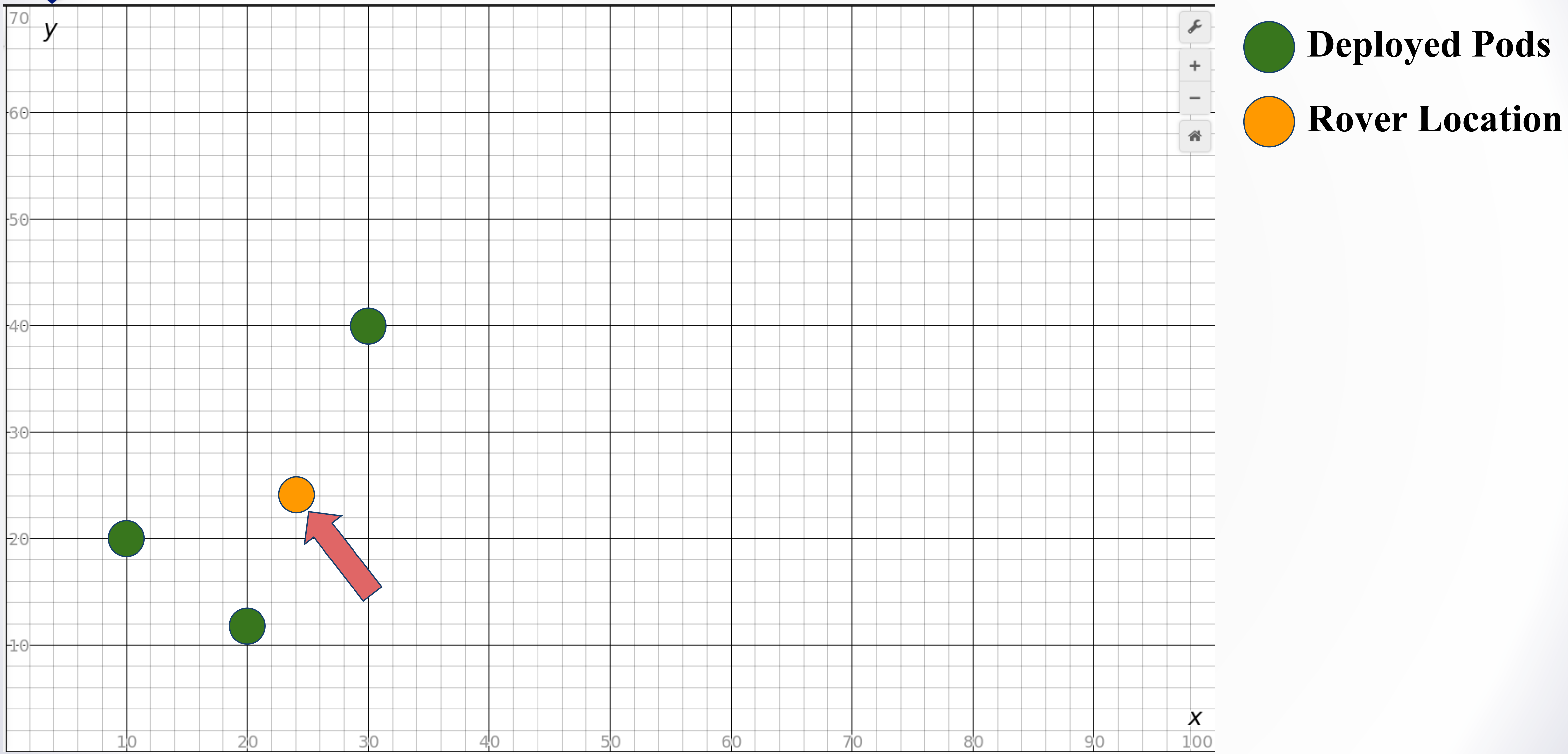
Trilateration Error

$$(\sigma_x, \sigma_y) = f(\underbrace{\{x_i, y_i\}, \quad \{r_i\}}_{\text{Determines Dilution of Precision (DOP) error}}, \quad \{\sigma_{xi}, \sigma_{yi}\}, \quad \{\sigma_{ri}\})$$

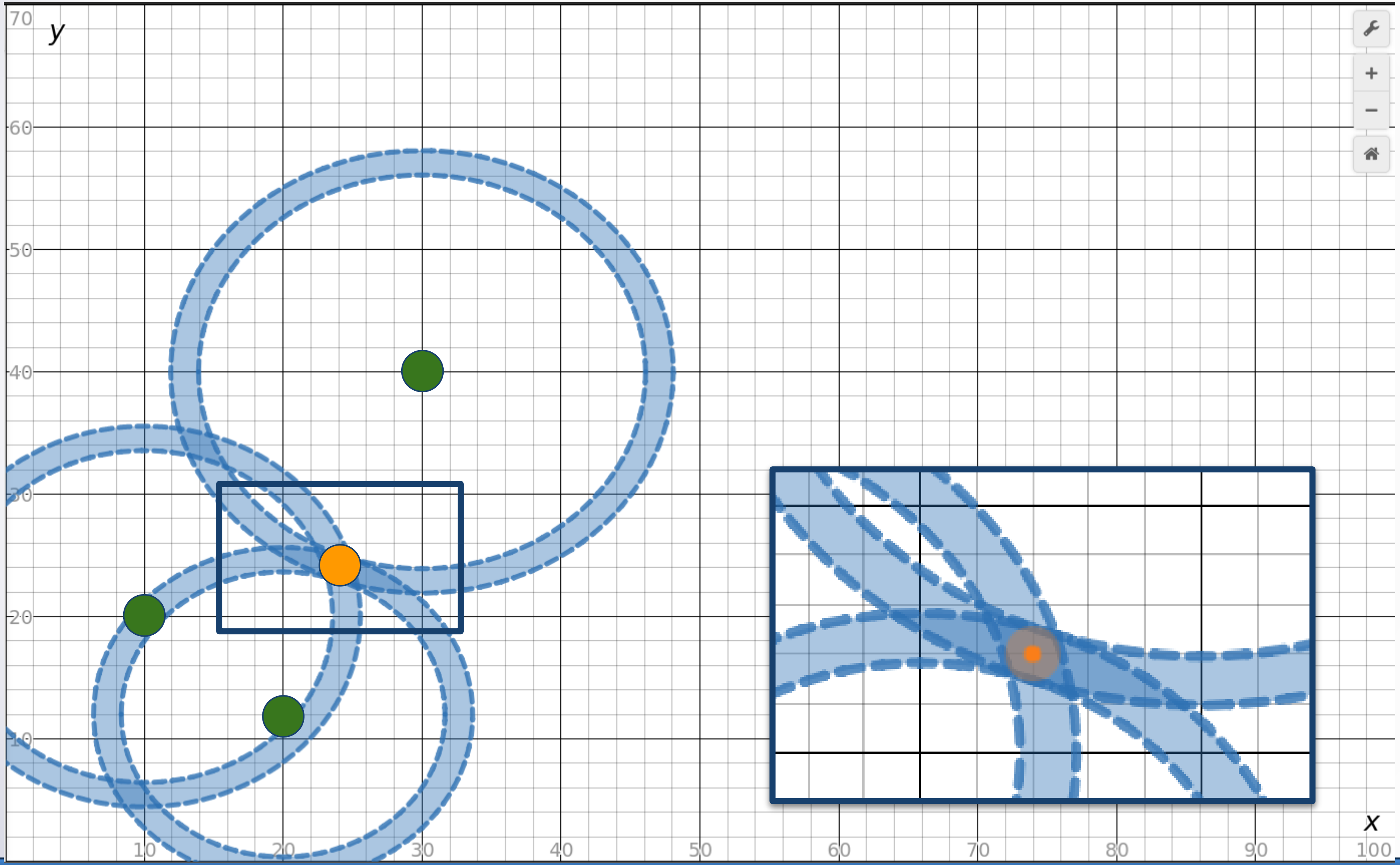
Determines Dilution of Precision (DOP) error

(σ_x, σ_y)	Rover position uncertainty (require $\leq$ 1m)
$\{x_i, y_i\}_{i=1}^3$	Positions of deployed pods
$\{r_i\}_{i=1}^3$	Distance to deployed pods
$\{\sigma_{xi}, \sigma_{yi}\}_{i=1}^3$	Uncertainty in pod location
$\{\sigma_{ri}\}_{i=1}^3$	Uncertainty in range measurement

Trilateration: Situation 1



Trilateration: Situation 1



Deployed Pods

Rover Location

Range with Uncertainty

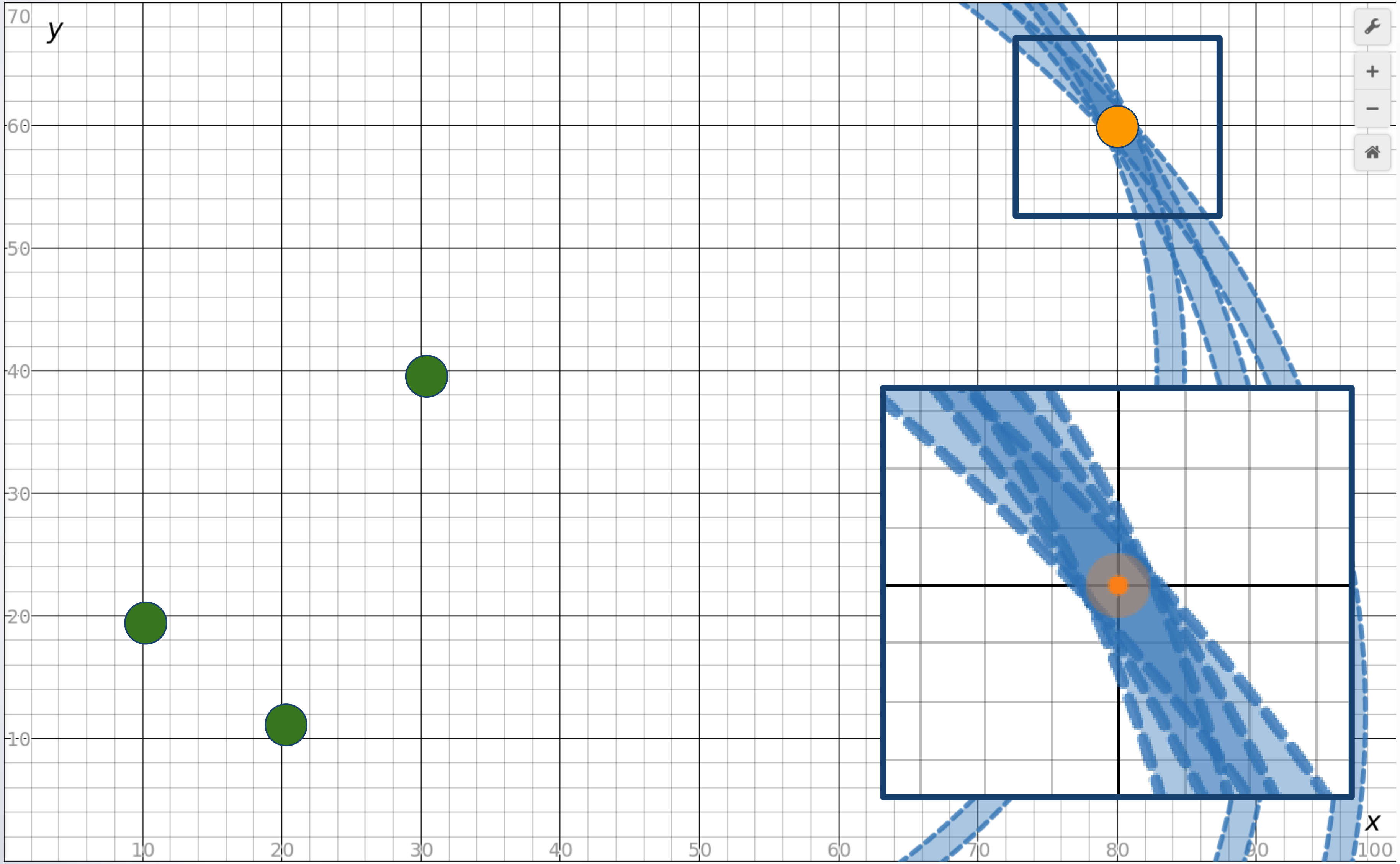
- **Area of overlap is uncertainty in trilateration**
- **Small if rover within triangle created by pods**
- **1m range uncertainty results in .01m^2 region**

Trilateration: Situation 2

Deployed Pods

Rover Location

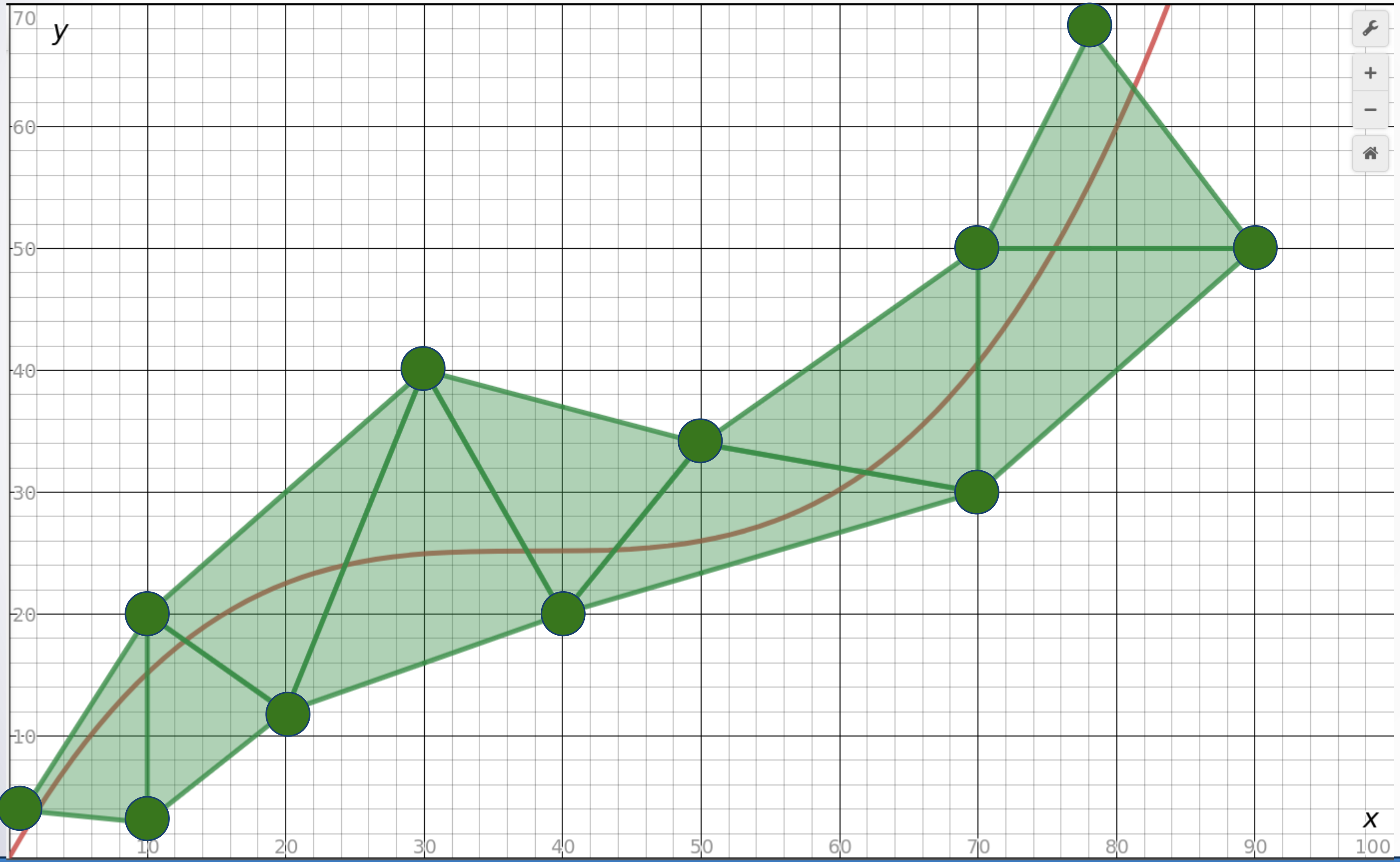
Trilateration: Situation 2



- Deployed Pods
- Rover Location
- Range with Uncertainty

- **Area of overlap is uncertainty in trilateration**
- **Large if rover outside triangle created by pods**
- **1m range uncertainty results in .70m^2 region**

Trilateration: Pod Placement



- **Low Dilution of Precision within triangle created by pods, high outside triangle (.01m² vs. .70m² in previous)**
- **By deploying pods to ensure the rover is always within these triangles, will be able to accurately trilaterate**

Trilateration Error

$$(\sigma_x, \sigma_y) = f(\{x_i, y_i\}, \quad \{r_i\}, \quad \{\sigma_{xi}, \sigma_{yi}\}, \quad \{\sigma_{ri}\})$$

- $\{x_i, y_i\}$, $\{r_i\}$: **Low DOP error with appropriate pod coverage**
- $\{\sigma_{xi}, \sigma_{yi}\}$: **Deployment Mechanism Constraint**
- $\{\sigma_{ri}\}$: **~1% of r_i (Discussed Later)**

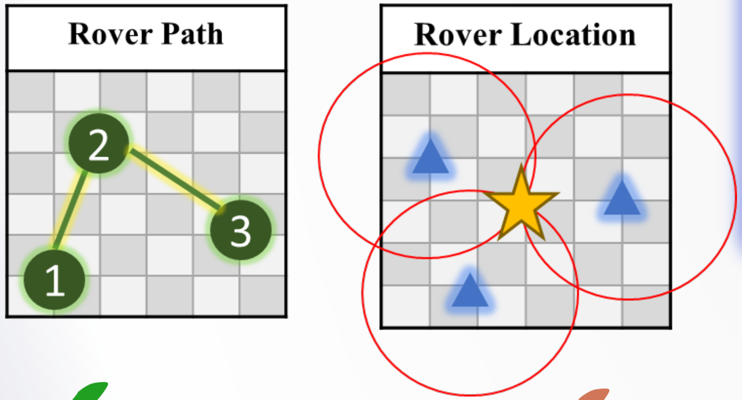
(σ_x, σ_y)	Rover position uncertainty (require ≤ 1m)
$\{x_i, y_i\}_{i=1}^3$	Positions of deployed pods
$\{r_i\}_{i=1}^3$	Distance to deployed pods
$\{\sigma_{xi}, \sigma_{yi}\}_{i=1}^3$	Uncertainty in pod location
$\{\sigma_{ri}\}_{i=1}^3$	Uncertainty in range measurement

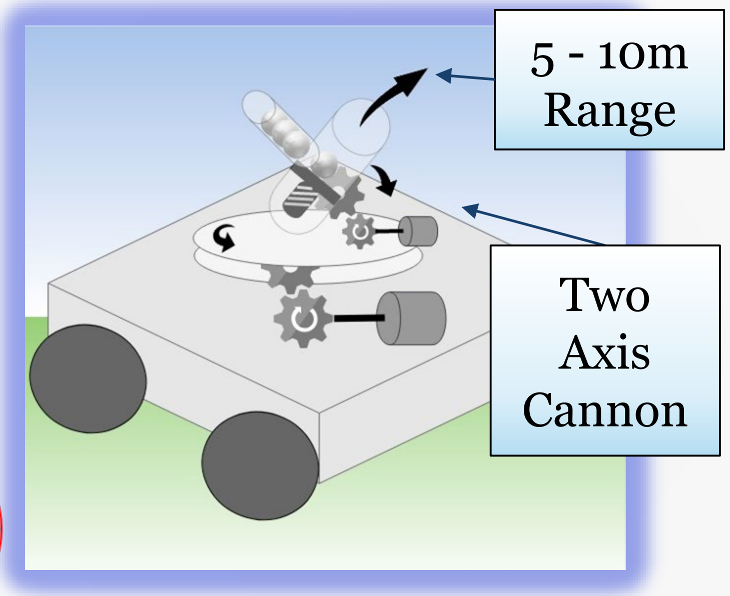
Critical Project Elements

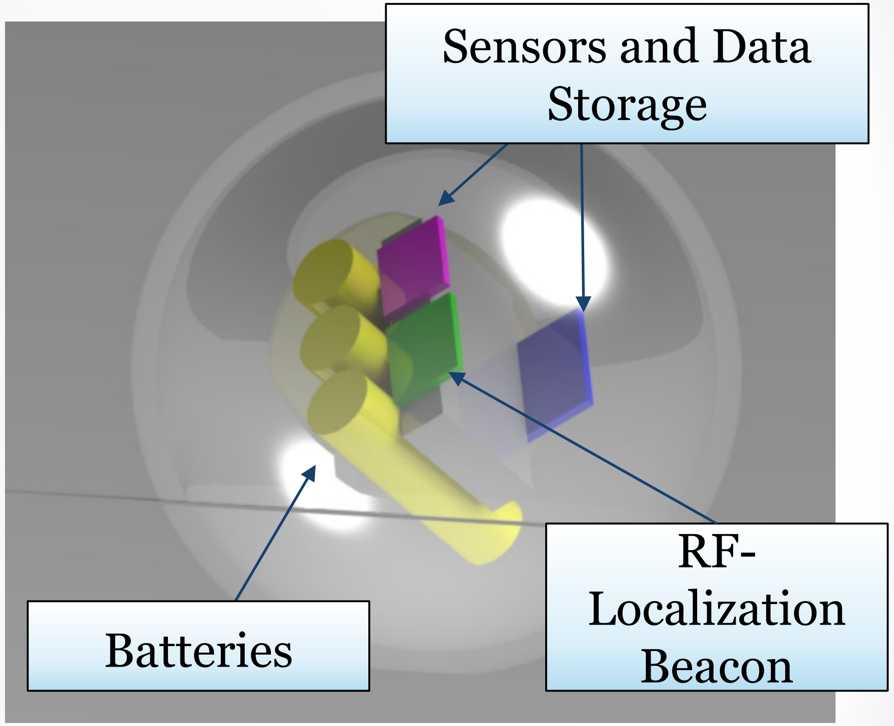
Software:

Deployment:

Pods:

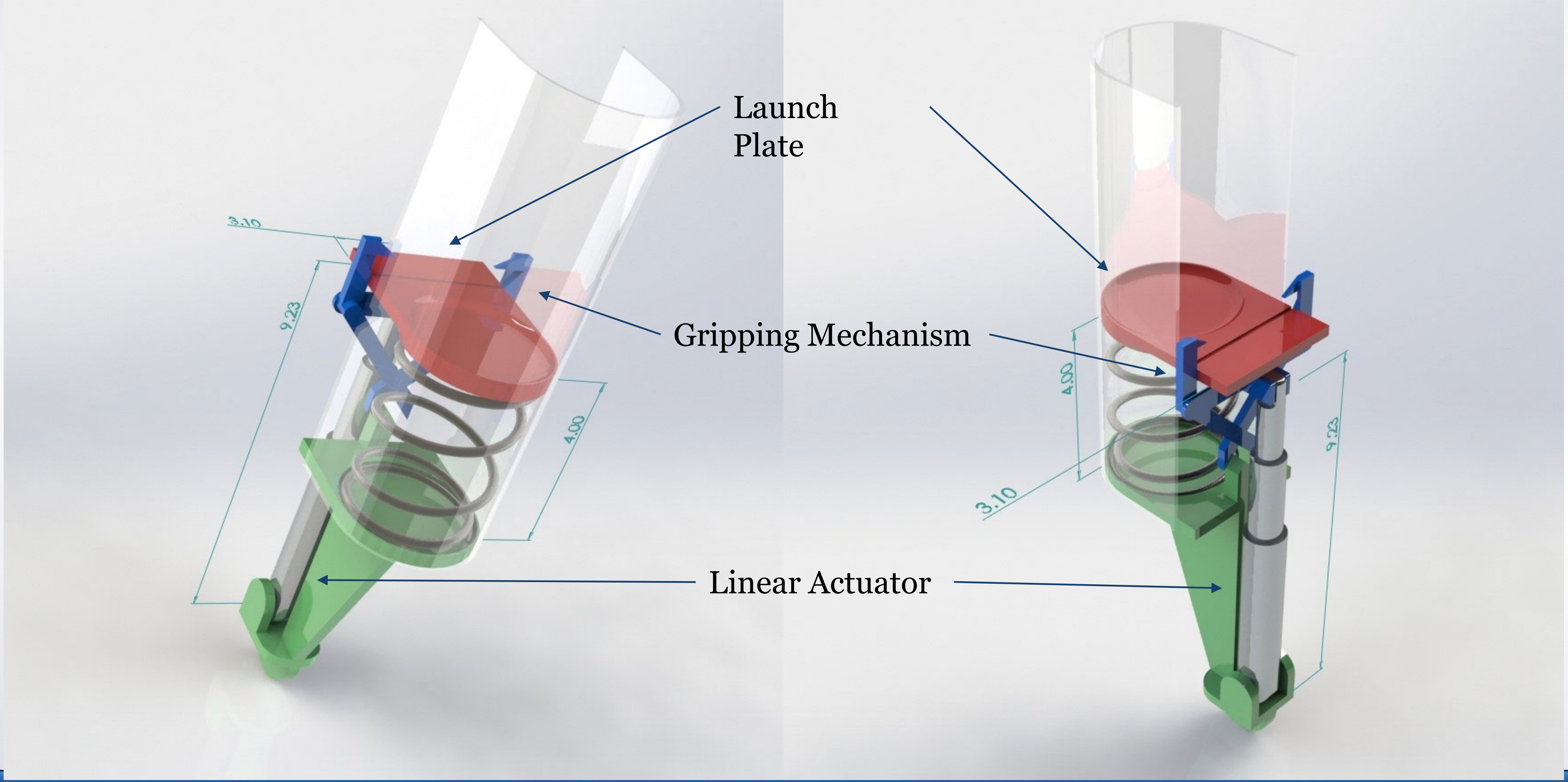






- ✓ Generate path between waypoints
- ✓ Estimate rover position using ranging data
- Deploy pods from rover
- Get ranging data from RF beacons
- Power pods for 2 Hours

Deployment Mechanism



Feasibility - Pod Weight and Size

Requirements

P5.1.1 Pods shall have under .75kg mass

P5.1.2 Pods shall have diametric dimensions less than 8.9cm (3.5in), and total length that shall not impair DM reload capability.

Feasibility Test Rationale

Due to mass budget and rover size, pods must be within required design tolerance

Conclusion

As seen in table to right, both mass and size requirements have been met.

Component	Mass [g]	Volume [cm^3]
Localino	10	15.283
Sensor Suite	1.2	0.64
Accelerometer	2	0.54
Batteries	70	33.75
SD card Reader	2	1.25
Wires/Headers	5	1
Electronics Total	96.5	52.4604
Polycarbonate shell	300	
Stopping Mechanism	60	
Mounting Hardware	40	
Fins	20	
Margin to 0.75kg	239.8	
Total Pod Mass	**516.5** ✓	**~50 x 53 x 40mm** ✓

Feasibility - Deployment Distance

Requirement

D3.2 DM shall have a variable range between 5 and 10 meters

Feasibility Test Rationale

Verify that there are COTS linear actuators and springs that can deploy a pod between 5 and 10 meters.

Method

1. Calculate required launch velocity
2. Use launch velocity to calculate required spring force
3. Check for COTS components that meet specification

Feasibility - Deployment Distance

Velocity Assumptions

- Conservation of energy
- No friction forces
- No wind
- No viscous effects
- Constant gravity (9.81 m/s²)

Velocity Governing Equations

$$x = V_0 cos(\theta)$$
$$y = V_0 sin(\theta) - \frac{1}{2}gt^2$$

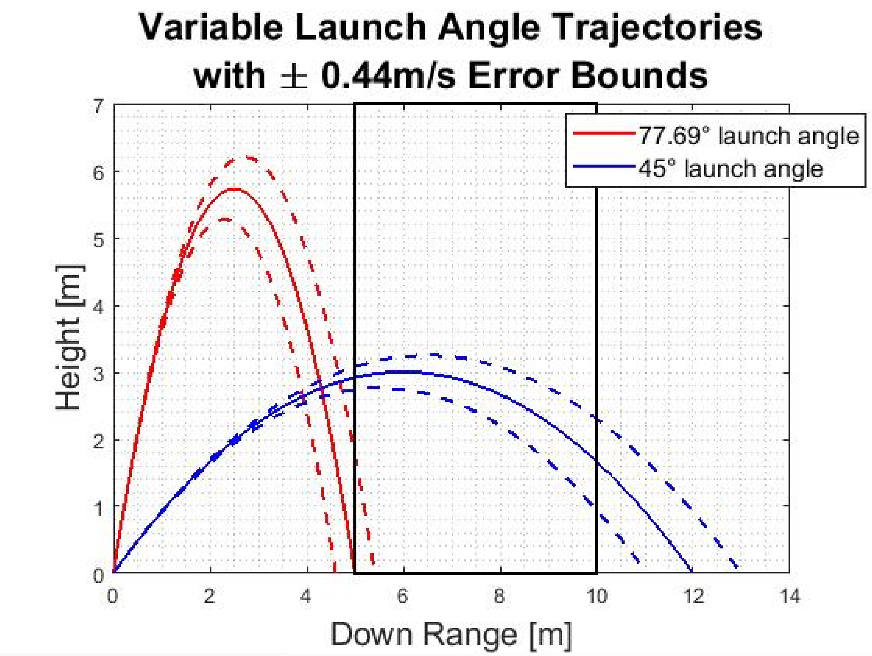
$$\Rightarrow x = \frac{V_0^2 sin(2\theta)}{g}$$

$$\Rightarrow V_0 = \sqrt{\frac{xg}{sin(2\theta)}}$$

Velocity Analysis Results

V_0 = 10.85 ± 0.44 m/s

Required Launch Angles



Feasibility - Deployment Distance

Force Assumptions

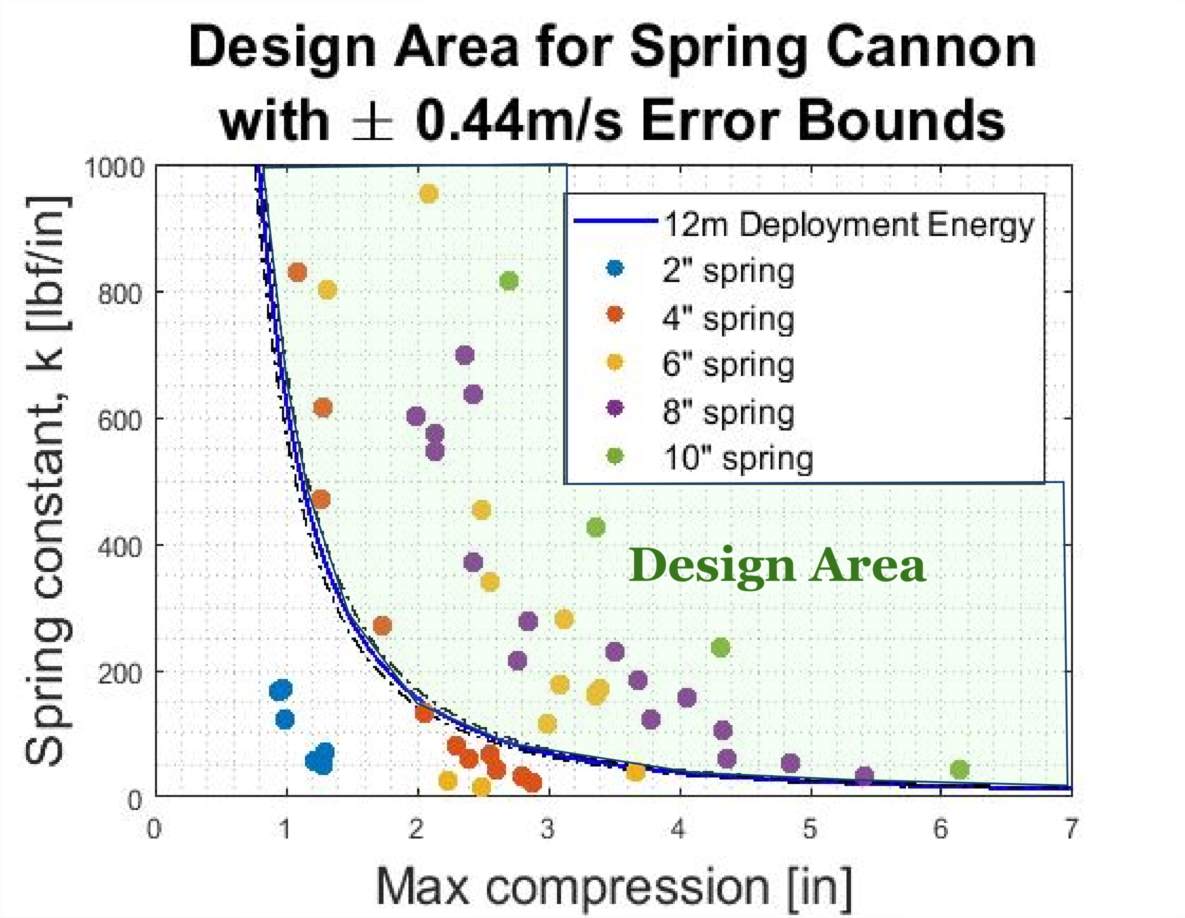
- Conservation of Energy
- Hooke's Law
- Vo = 10.85 m/s = 35.6 ft/s
- m = 600g = 0.0411 slugs
- E = 35.32 J = 312.6 in lbf

Force Governing Equations

$$PE_i + KE_i = PE_f + KE_f$$

$$\frac{1}{2}KX^2 = \frac{1}{2}mV^2$$

$$F = Kx$$



Each dot represents specifications for a COTS spring ✓

Feasibility - Deployment Distance

Linear Actuator Verification

Spring Length	Force Range in Design Space [lbs]
4”	470-906
6”	351-1996
8”	175-1652
10”	258-2200

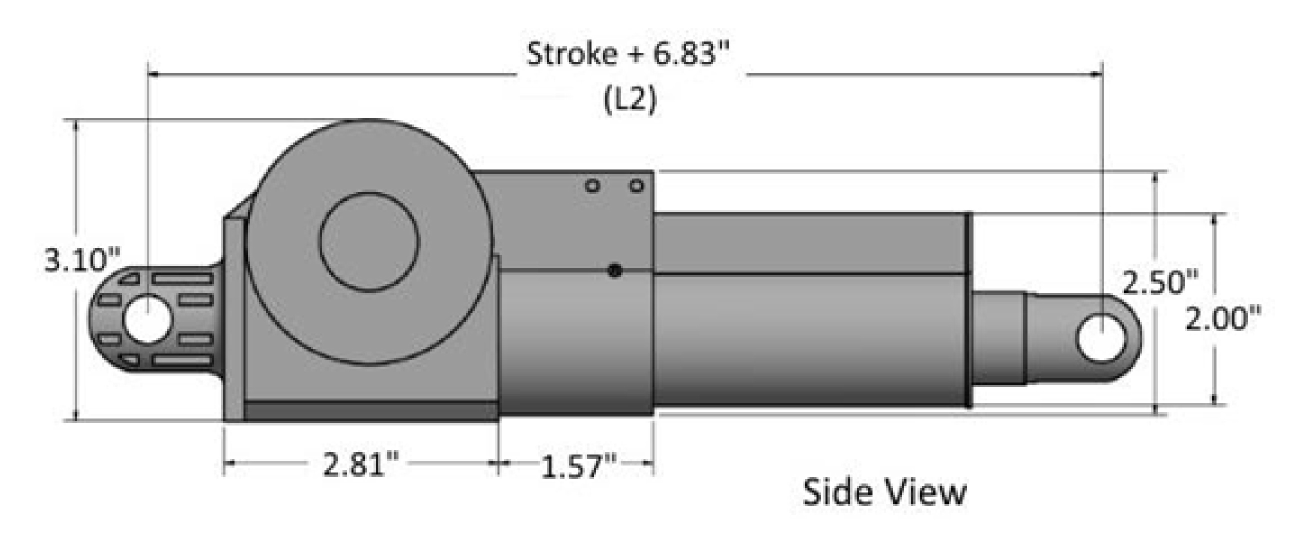
Standard COTS heavy duty linear actuators range between 850-1000 lbs

Minimum Force = 175 lbs ✓

Conclusion

D3.2 DM shall have a variable range between 5 and 10 meters

Based on spring force analysis, there are COTS springs and linear actuators that can deploy a pod to meet requirement D3.2



Feasibility - Deployment Location

Requirement

D3.3 DM shall have deployed pods land within 1m radius of software commanded location

Feasibility Test Rationale

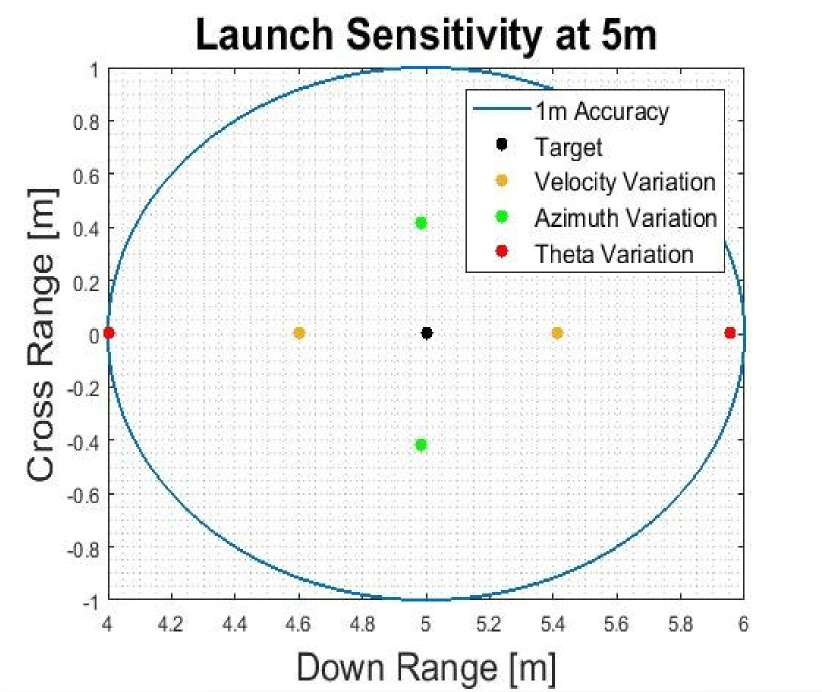
Verify that there are motors/actuators that can keep the deployment angles within tolerance to ensure a 1m radius accuracy

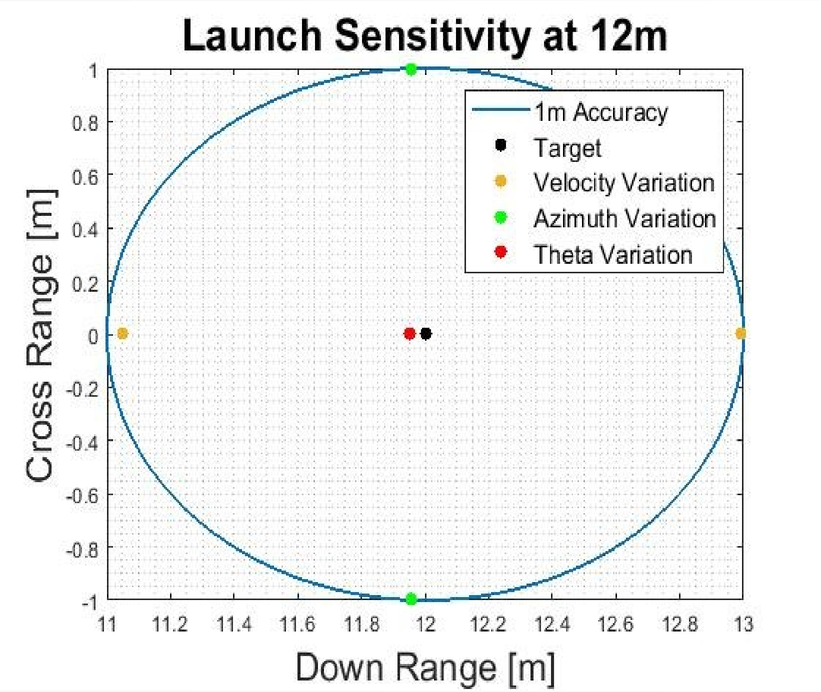
Method

1. Calculate azimuth and elevation angle tolerance for 1m radius error
2. Check for COTS motors/actuators that will allow deployment to stay within required tolerance

Feasibility - Deployment Location

Sensitivity Analysis Results





	Elevation	Azimuth	Velocity
Maximum Allowable Deviation	±2.57°	±4.78°	±0.44 m/s

Feasibility - Deployment Location

Motor Tolerance Verification

NEMA step motor

https://www.omega.com/

Specification	Degree Accuracy
200 steps per full revolution	1.8° ✓
51,200 steps/revolution with microstepping	0.007°

Conclusion

D3.3 DM shall have deployed pods land within 1m radius of software commanded location

With a necessary accuracy of 2.57° elevation and 4.78° azimuth, there is a feasible option for adjusting these angles such that the pod is launched to within 1m of the desired location.

Feasibility - Stopping Method

Requirement

D4.1.2 Pod shall be designed to maintain position upon impact at final landing position, eg prevent bouncing outside of accuracy tolerance.

Feasibility Test Rationale

Verify that a feasible stopping mechanism exists and will prevent the pod from bouncing or rolling out of the tolerance zone.

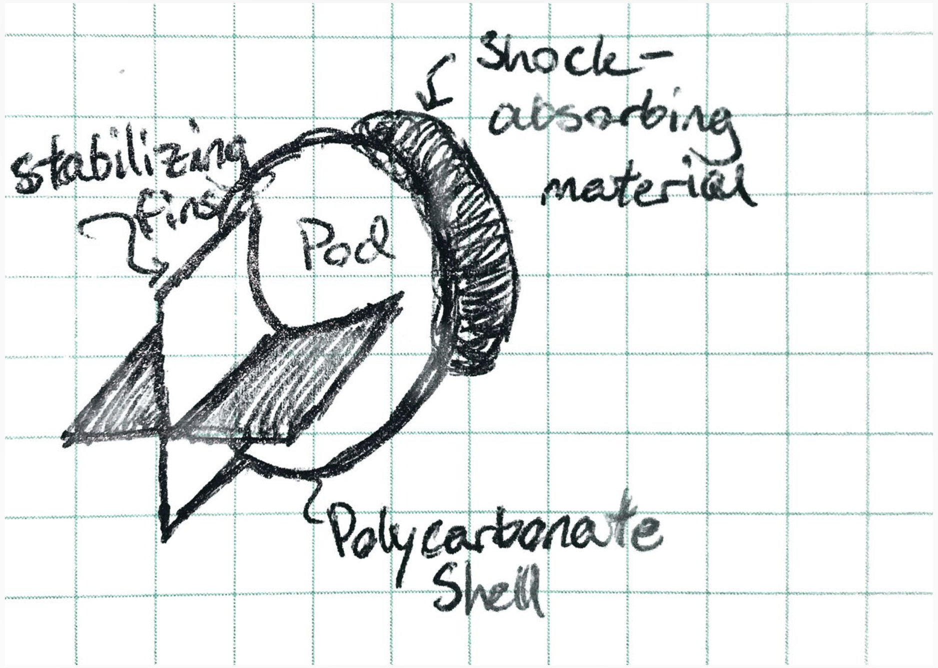
Method

1. Prototype stopping methods to determine ideal and non-ideal dynamical responses
2. Improve upon stopping method to better ensure desired orientation upon landing

Feasibility - Stopping Method

Stopping Method Design

- Spherical shell
- Crumple-zone type stopping mechanism
- Stabilizing fins



Feasibility - Stopping Method

Conclusion

D4.1.2 Pod shall be designed to maintain position upon impact at final landing position, eg prevent bouncing outside of accuracy tolerance.

- Based on preliminary testing, the design is feasible
- More testing is needed to ensure landing orientation of shock absorber

✓

Feasibility - Landing Force

Requirement

P5.4.2 Pod shells and internal components shall be durable enough to withstand at least 30 deployments

Feasibility Test Rationale

Explore the effects of damping mechanisms/materials and necessity of internal damping for electronics. Verify that there are materials that will not break upon impact.

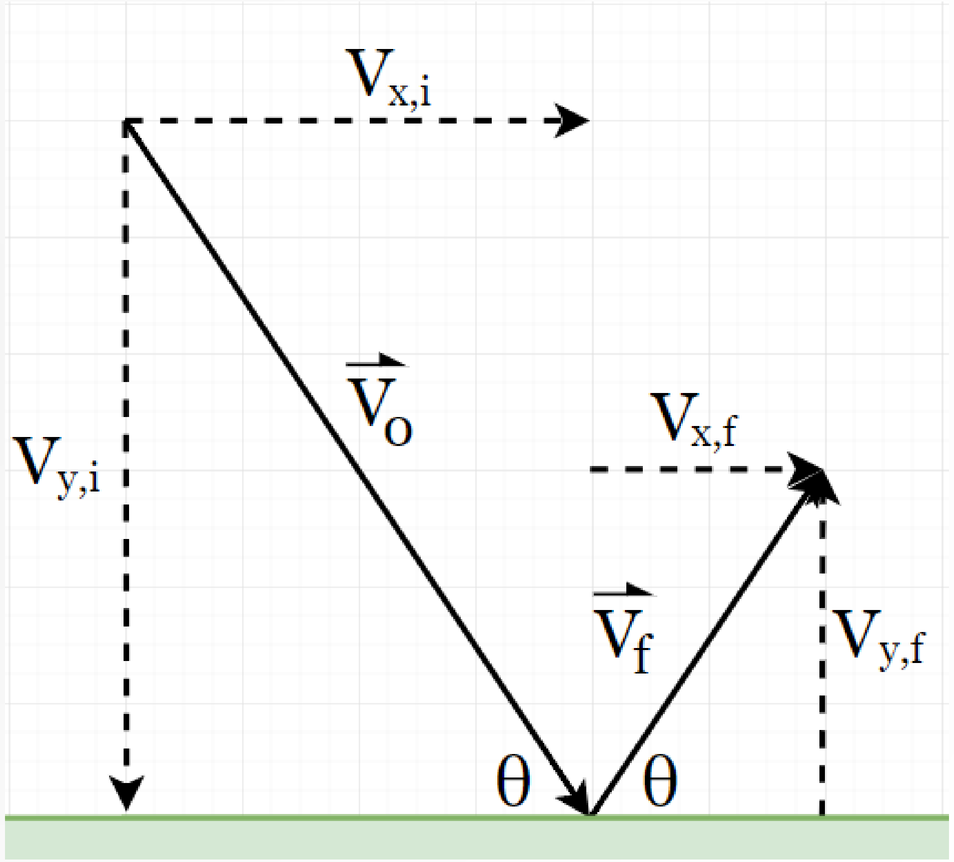
Method

1. Calculate landing forces with damping
2. Check stress on shell
3. Ensure internal components are mounted correctly.

Feasibility - Landing Force

Landing Force Assumptions

- No wind or drag
- Initial velocity at launch = final velocity at landing
- Change in momentum is partially elastic



Feasibility - Landing Force

Landing Force Governing Equations

$$V_f = -eV_0$$

$$F_x = \frac{m_p(V_{x_i} - (-V_{x_f}))}{\Delta t} = \frac{m_p(V_0 cos\theta - (-eV_0 cos\theta))}{\Delta t}$$

$$F_x = \frac{(1+e)m_p V_0 cos\theta}{\Delta t} \qquad F_y = \frac{(1+e)m_p V_0 sin\theta}{\Delta t}$$

$$|F| = \frac{(1+e)m_p V_0}{\Delta t}$$

- Coefficient of restitution, e = 0.5
- m = 600 g = 0.0411 slugs
- Δt = 0.04 s
- Vo = 10.85 m/s = 35.597 ft/s

F = 244 N = 55 lbf

n = 24.9 g’s

Feasibility - Landing Force

Shell Strength Analysis

Yield strength of polycarbonate: 9137 psi

$$\sigma = \frac{F}{A}$$

Necessary impact area: 0.006 in^2

Internal Component Strength Analysis

Compact and tight mounting of electronics with most sensitive components on bottom.

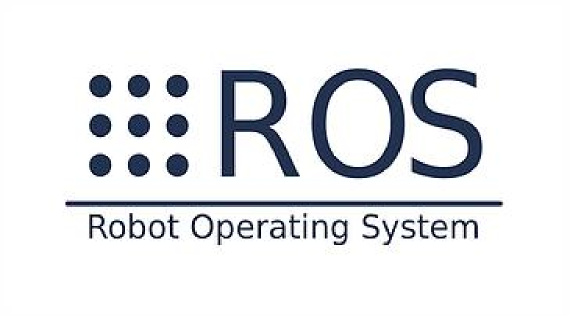
Further testing and analysis will be completed upon more accurate component strength values

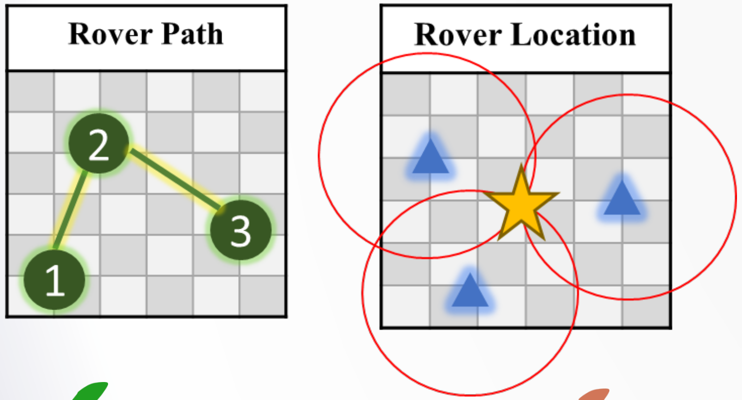
Conclusion

P5.4.2 Pods shall be durable enough to withstand at least 30 deployments ✓

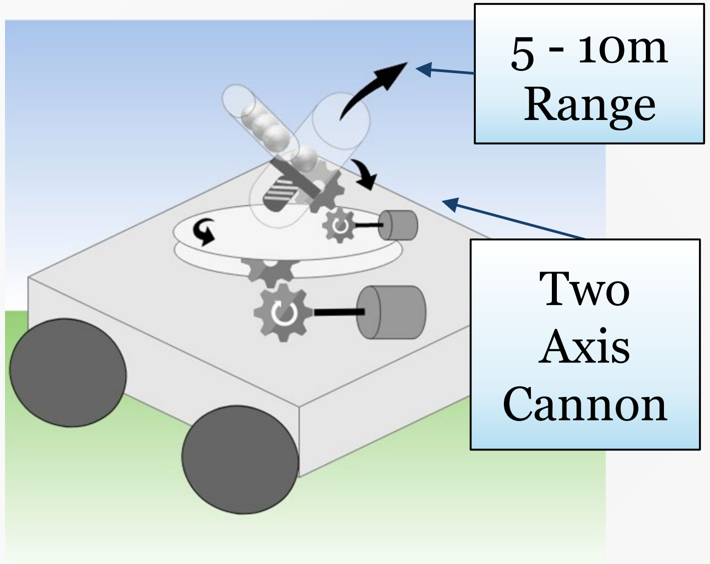
Critical Project Elements

Software:

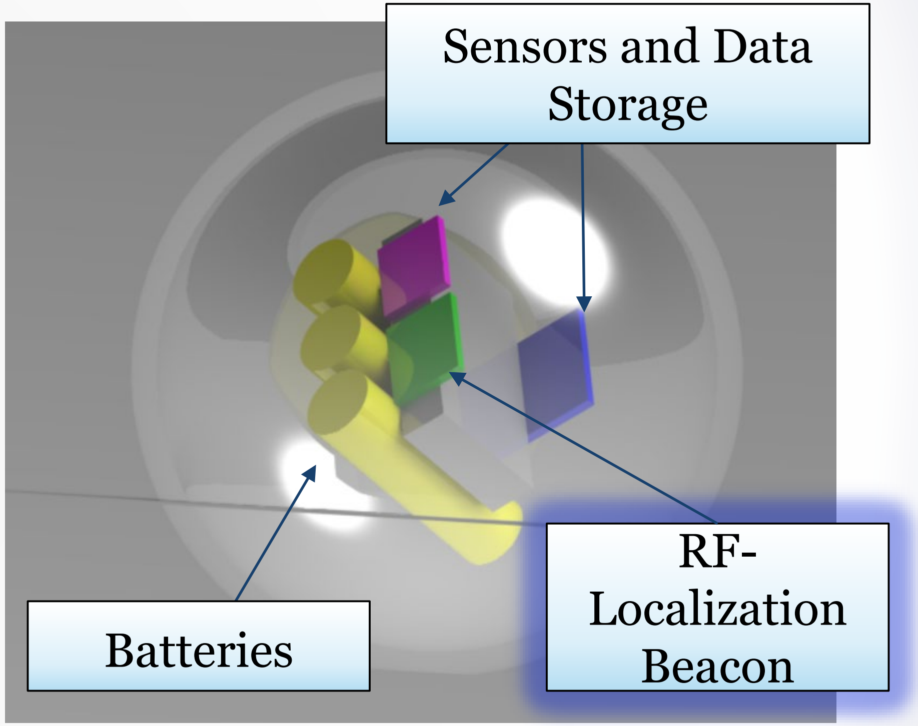




Deployment:



Pods:

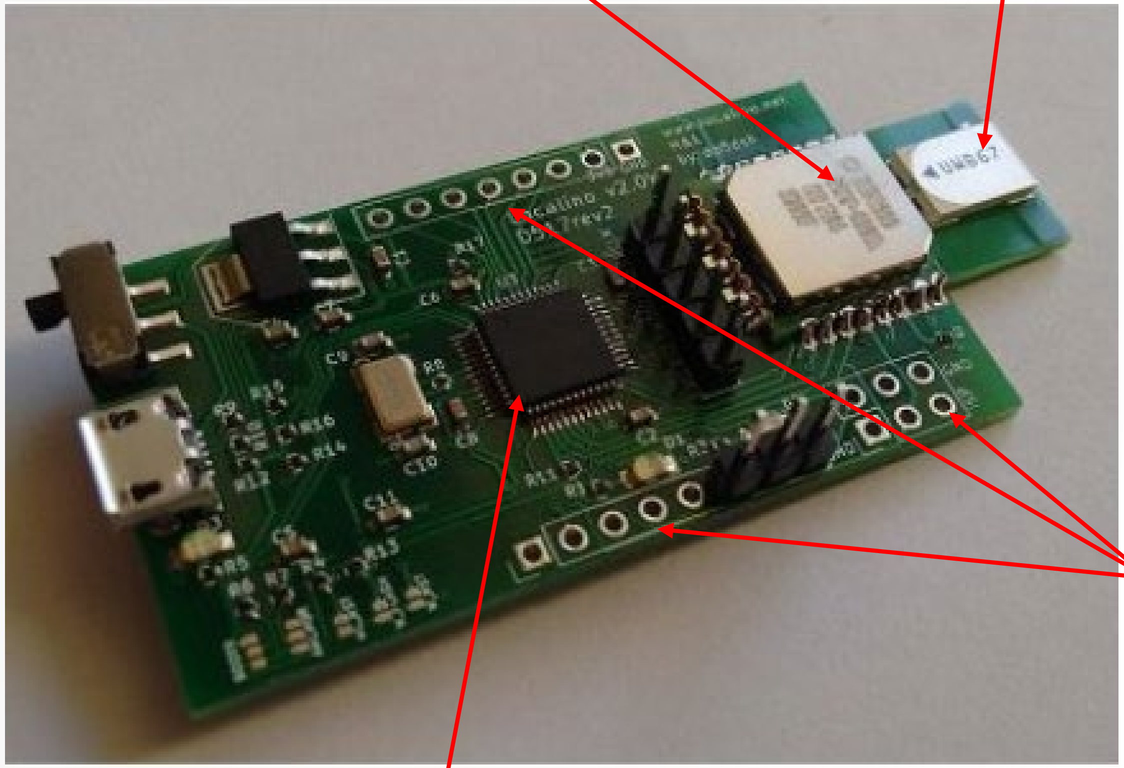


- ✓ Generate path between waypoints
- ✓ Estimate rover position using ranging data
- ✓ Deploy pods from rover
- Get ranging data from RF beacons
- Power pods for 2 Hours

Ranging Feasibility

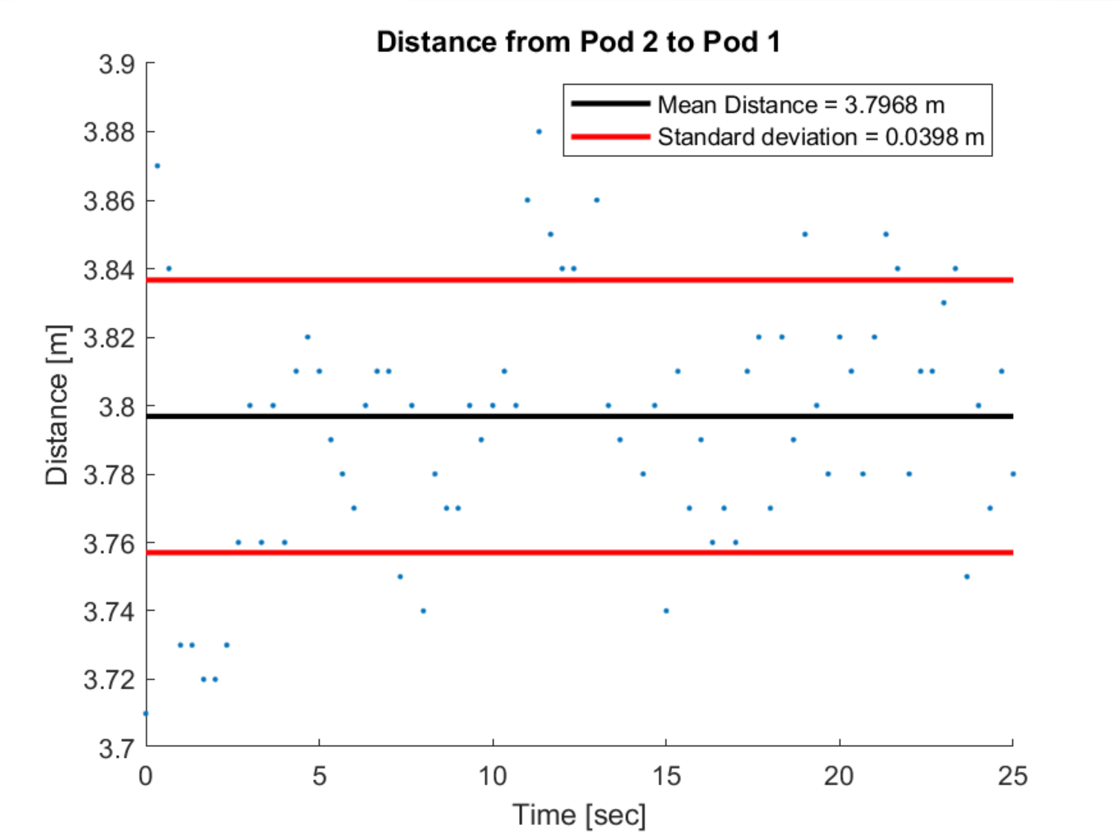
Requirements:
P5.2 P - The pods shall communicate data to the rover and amongst themselves
P8.2 P - Pods shall contain DWM1000 and STM 32 for RF-localization and communication

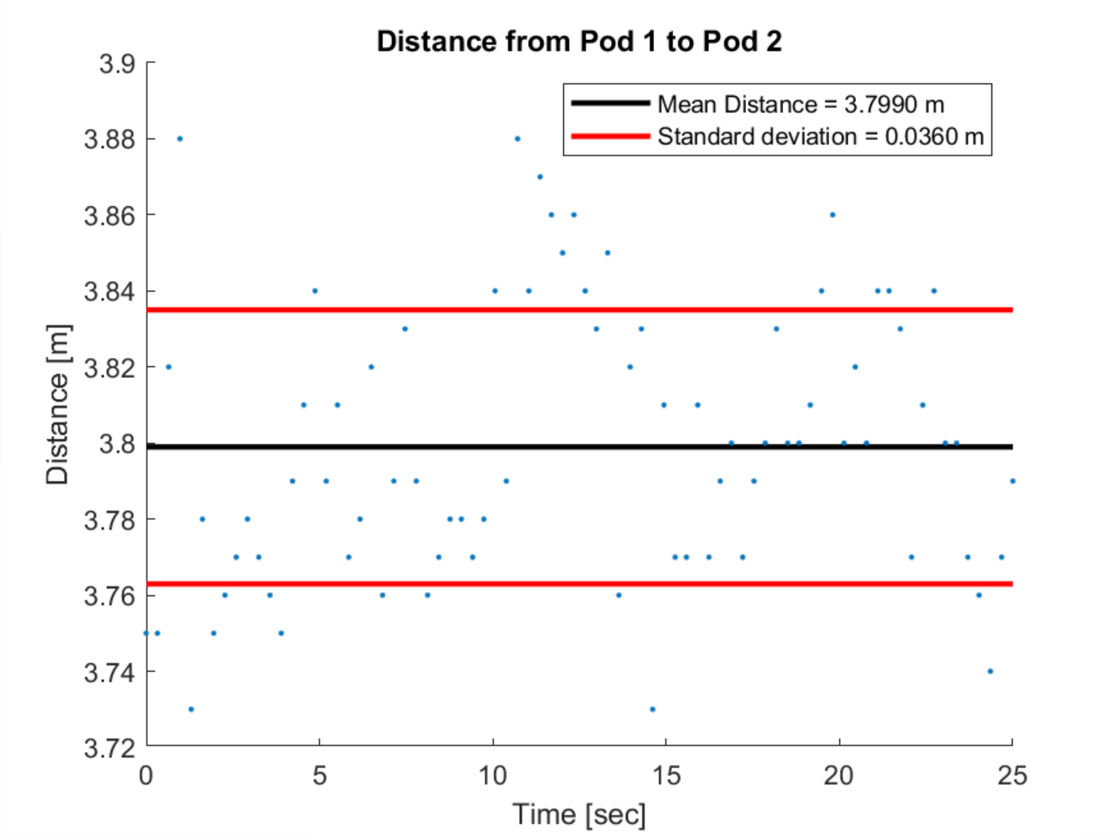
Design Approach:
Use customer provided heritage baseline ranging software and hardware to jumpstart design



Ranging Test

P5.2 P - Ranging modules are able to ping each other and calculate distance
P8.2 P - Both STM32 and DWM1000 is used for ranging ✓





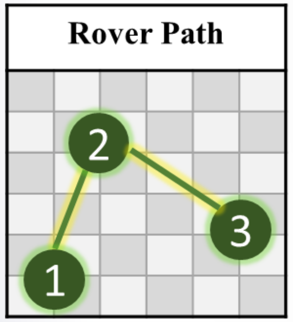
- Ranging test was performed by keeping putting between two Localinos on opposite sides of a room
- **0.0021 meter difference** in average distance which is within deviation for both

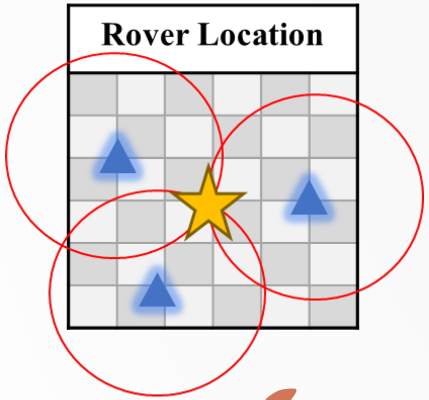
Critical Project Elements

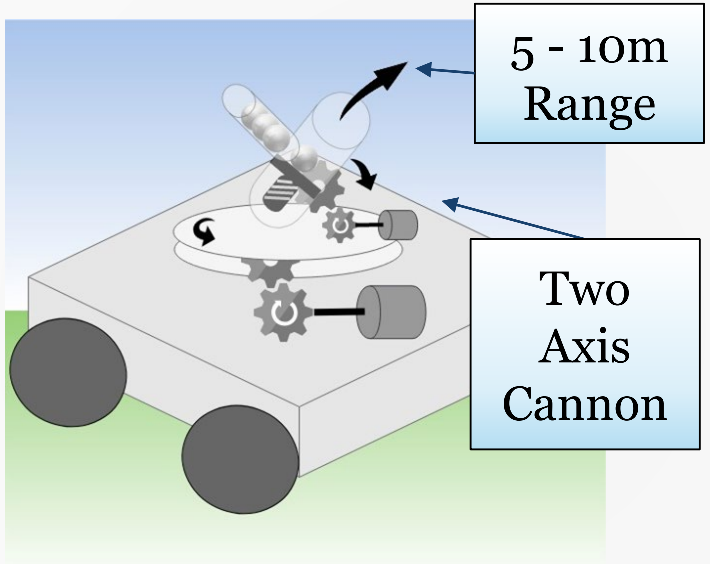
Software:

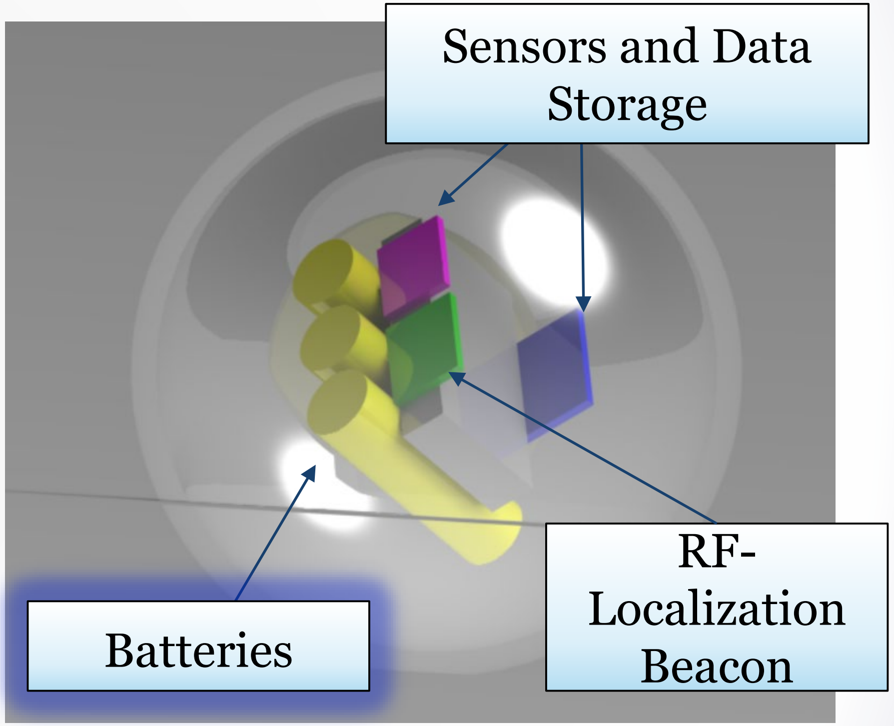
Deployment:

Pods:









✓ Generate path between waypoints

✓ Estimate rover position using ranging data

✓ Deploy pods from rover

✓ Provide RF ranging data

Power pods for 2 Hours

Power Feasibility

Requirements:

P5.3.2 - Battery shall have sufficient capacity to meet a 5% duty cycle between low and high power mode for 2 hour duration test

Design Approach:

Use NiCd for Power Source. NiCd AA batteries commonly have at least 1000 mAh of battery capacity which is well above the required capacity if all components were running for the full 2 hour duration test.

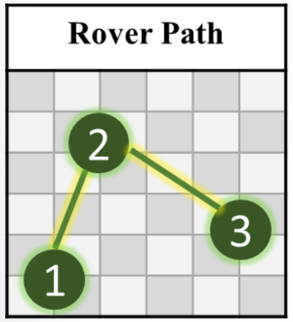
Component	Current	Duration	Capacity Needed
Localino	100 [mA]	120 [min]	200 [mAh]
DWM1000	100 [mA]	120 [min]	200 [mAh]
Sensor Breakout Board	40 [μA]	120 [min]	80 [μAh]
Accelerometer	145 [μA]	120 [min]	290 [μAh]
SD Card Reader	100 [mA]	120 [min]	200 [mAh]
Total	**300.245 [mA]**		**600.37 [mAh]**

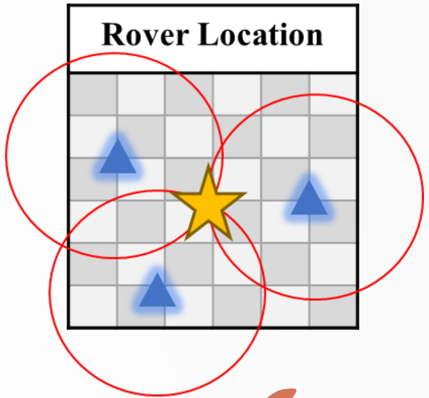
P5.3.2 - Battery shall have sufficient capacity to meet a 5% duty cycle between low and high power mode for 2 hour duration test ✓

Critical Project Elements

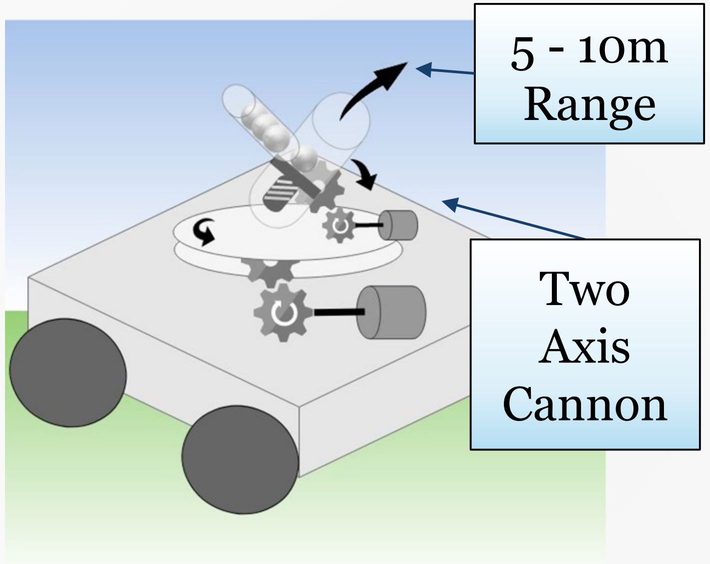
Software:

ROS
Robot Operating System

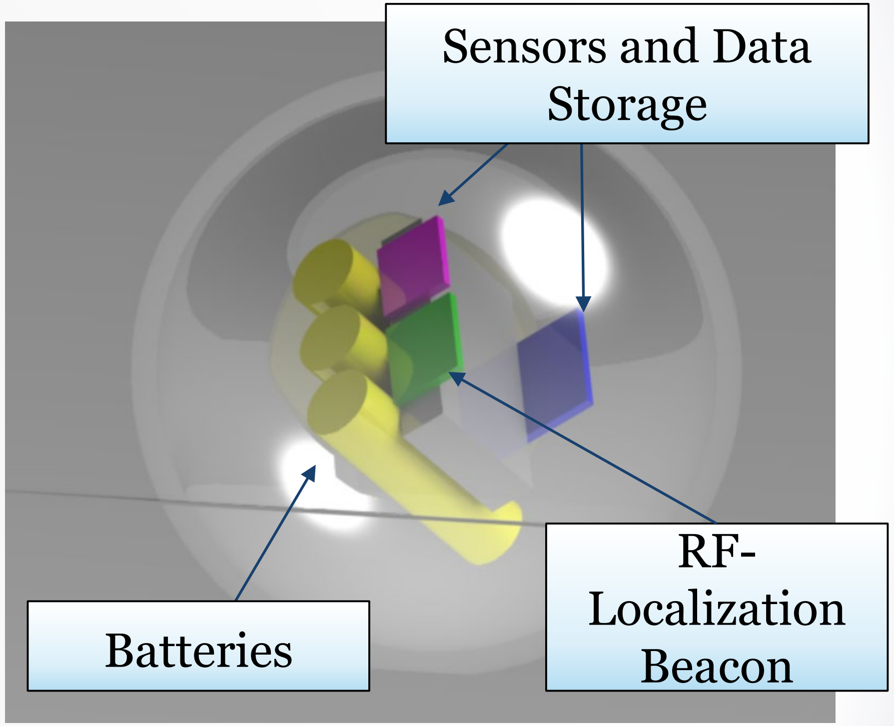




Deployment:



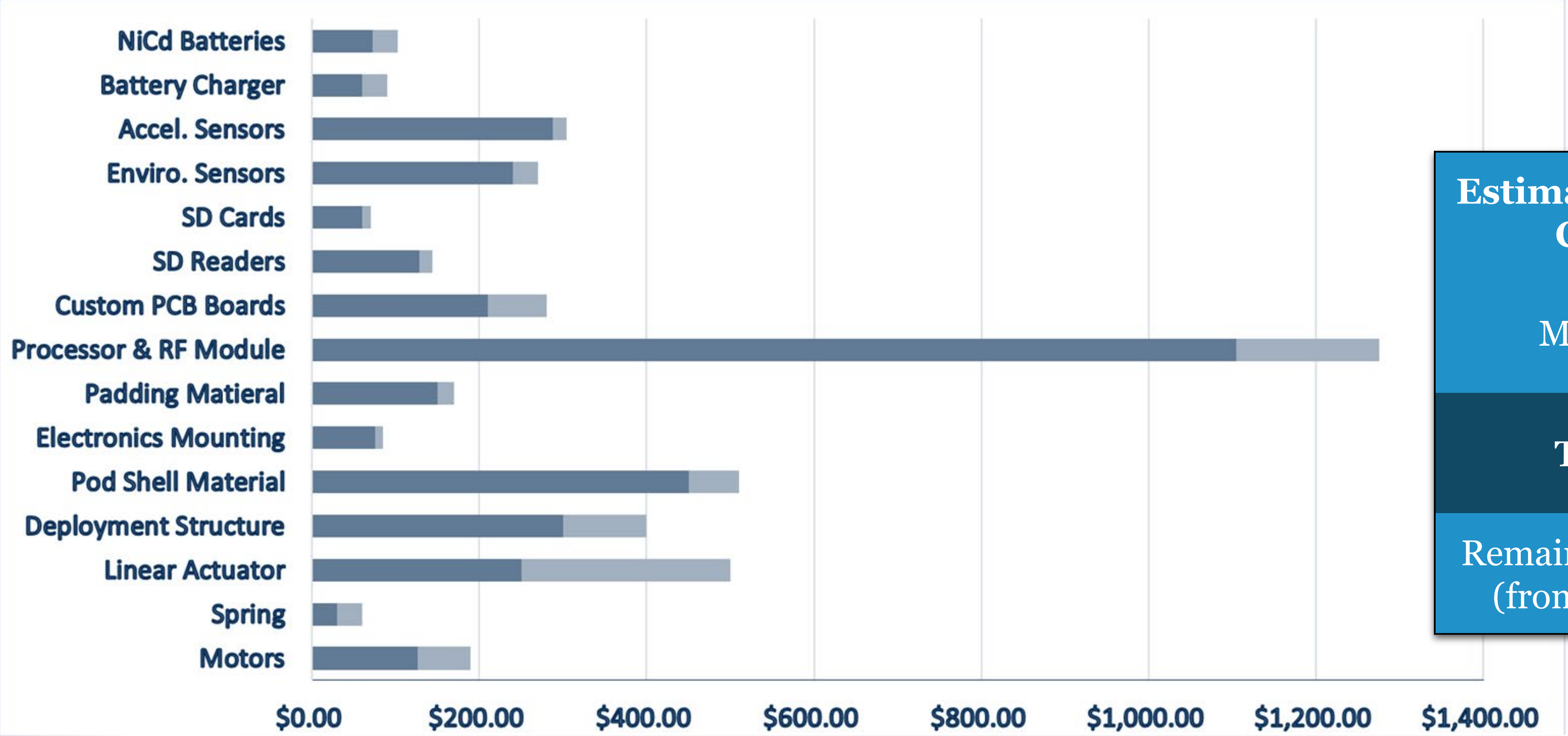
Pods:



- ✓ Generate path between waypoints
- ✓ Estimate rover position using ranging data
- ✓ Deploy pods from rover
- ✓ Provide RF ranging data
- ✓ Power pods for 2 Hours

Status Summary

Budget

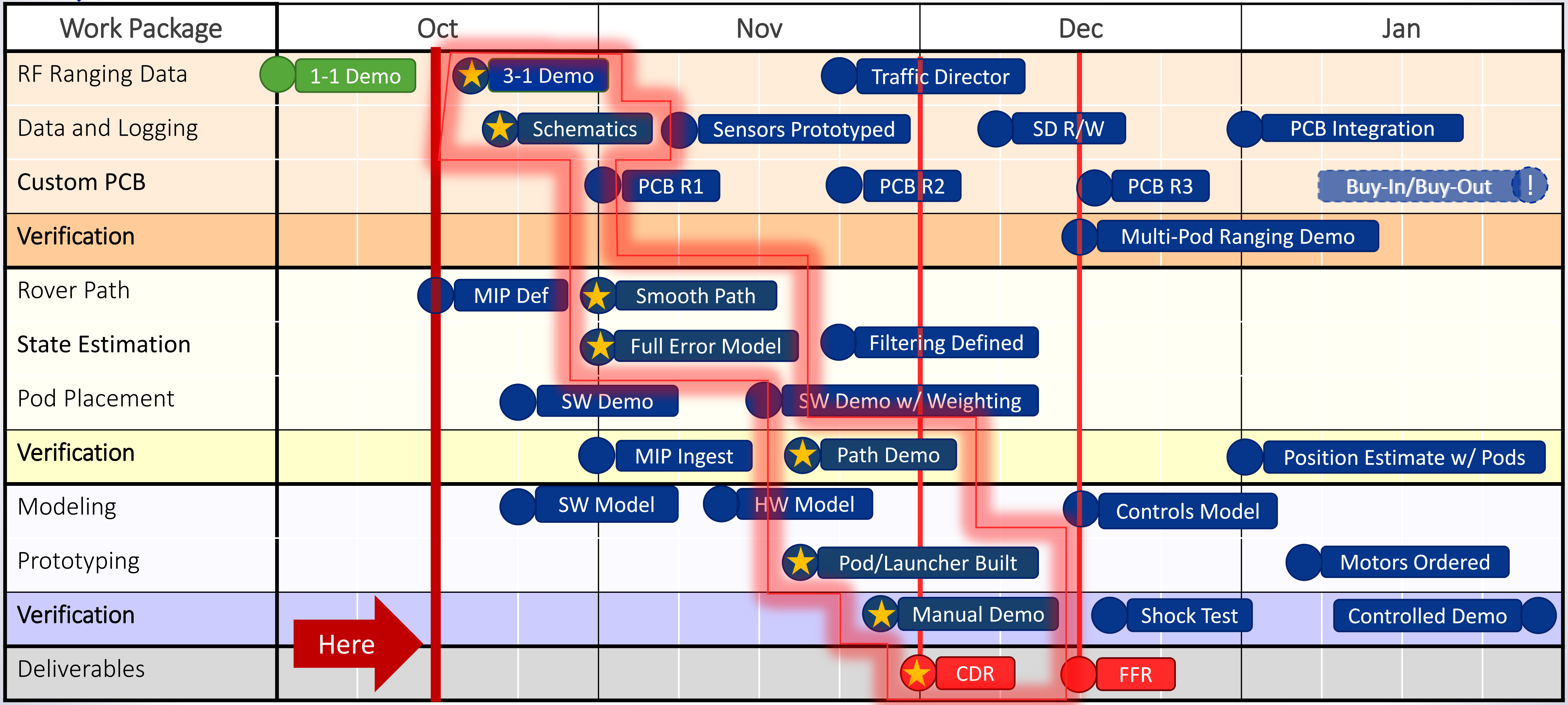


Estimated Total Cost:	**$3544.00**
Margin:	$905.00
Total	**$4449.00**
Remaining Funds (from $5000)	$551.00

Notes:

- Budget based on estimated **12 pods + 4 pods for prototyping**
- Jackal Rover provided by customer

Schedule


Work Package
Oct
Nov
Dec
Jan
RF Ranging Data
1-1 Demo
3-1 Demo
Traffic Director
Data and Logging
Schematics
Sensors Prototyped
SD R/W
PCB Integration
Custom PCB
PCB R1
PCB R2
PCB R3
Buy-In/Buy-Out
!
Verification
Multi-Pod Ranging Demo
Rover Path
MIP Def
Smooth Path
State Estimation
Full Error Model
Filtering Defined
Pod Placement
SW Demo
SW Demo w/ Weighting
Verification
MIP Ingest
Path Demo
Position Estimate w/ Pods
Modeling
SW Model
HW Model
Controls Model
Prototyping
Pod/Launcher Built
Motors Ordered
Verification
Manual Demo
Shock Test
Controlled Demo
Here
Deliverables
CDR
FFR

Next Steps

ID	Description	Feasible?	Next Steps for CDR
M1	Rover shall autonomously navigate along software generated path within 1m accuracy using RF-Localization Beacon correction to inertial navigation	✓	• Develop pod placement algorithm • Apply smooth path algorithm on rover
M2	The rover shall estimate its absolute position	✓	• Full error model for position estimation • Perform 3-1 Beacon-to-Rover Test
M3	The deployment mech shall have capability to deploy pods to software defined locations	✓	• Full error model for deployment • Prototype launching mechanism

Deployed **RF A**ntennas for **G**PS-denied **O**ptimization and Environmental **N**avigation

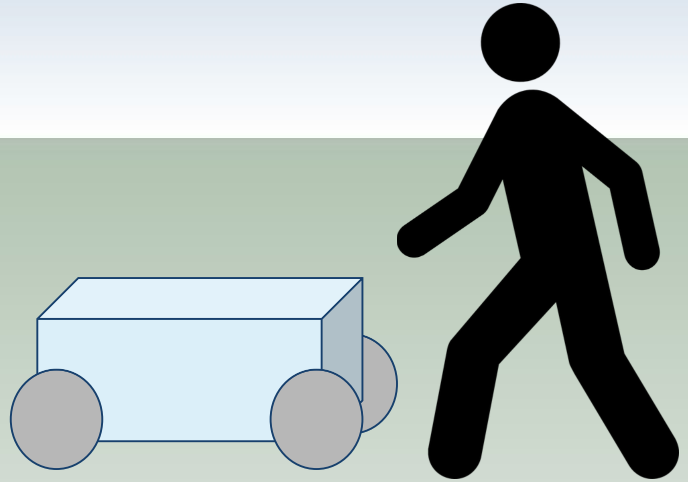
Thank you for your time --
We hope to see you at CDR!

Appendix

A: CONOPS

Step One:

- User uploads MIP to rover
- Rover ingests data and begins mission sequence

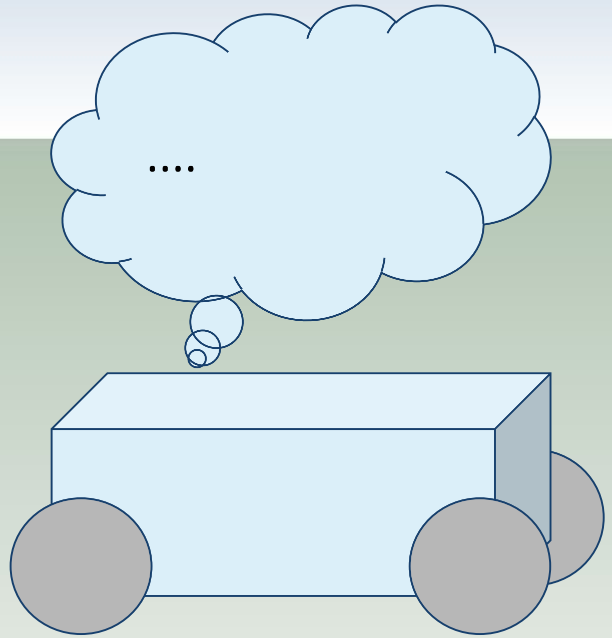


Problem Givens:

-Terrain Map
-Science Map
-Obstacle Map
-Waypoints

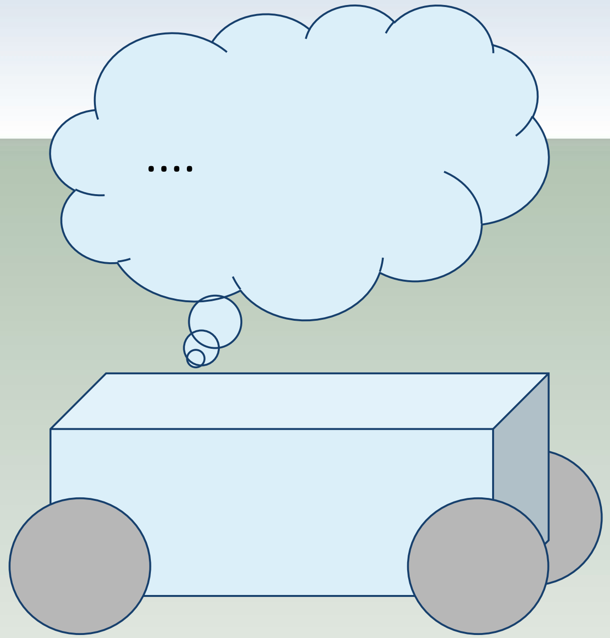
Step Two:

- Rover collects final GPS location
- GPS Cut OFF



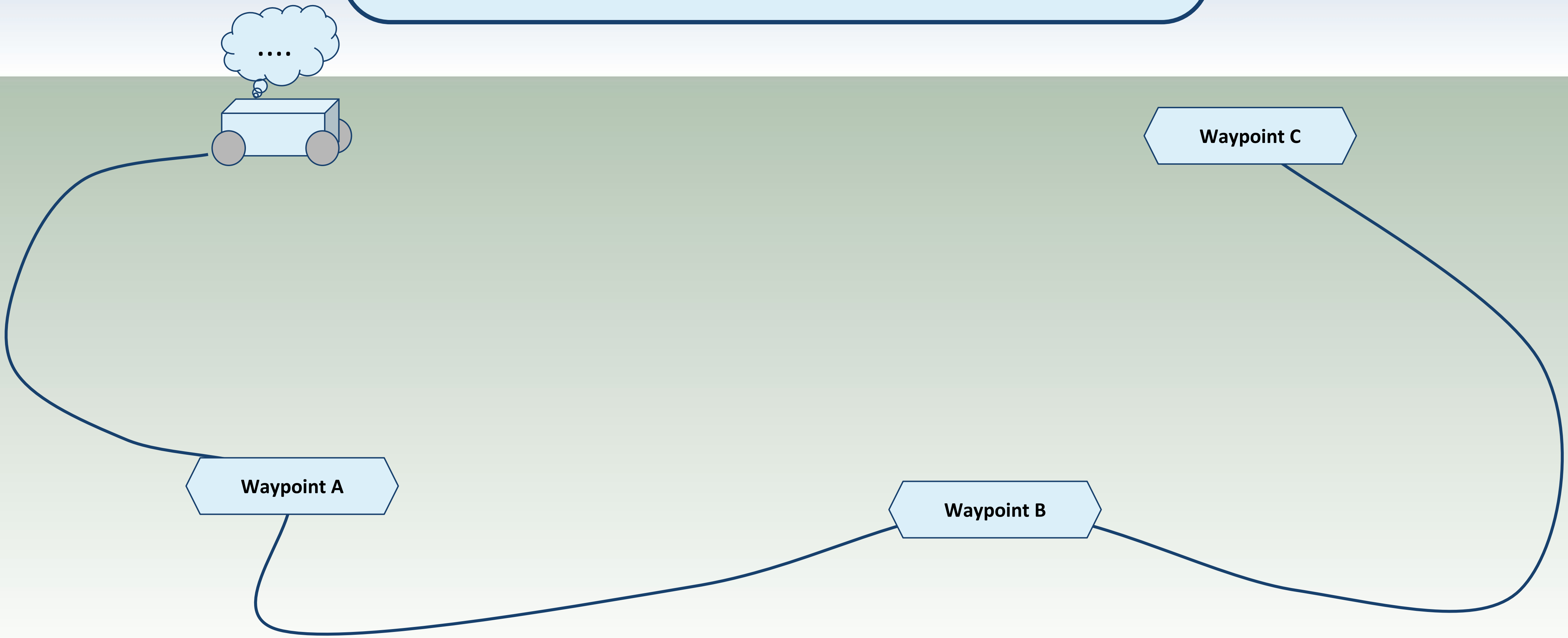
Step Two:

- Rover collects final GPS location
- GPS Cut OFF
- *Secretly, GPS still recorded for post-test analysis*



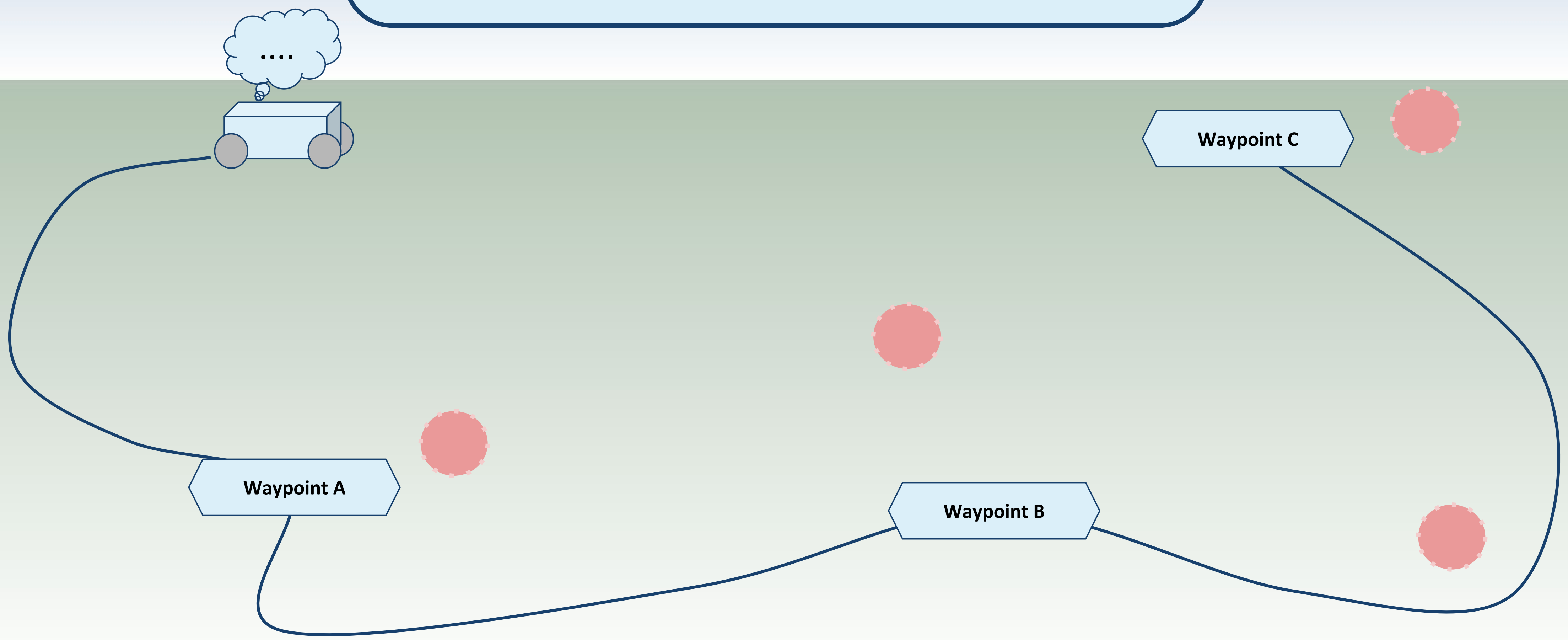
Step Three:

- Software combines maps and last GPS point
- Software generates path to waypoints (avoiding obstacles)



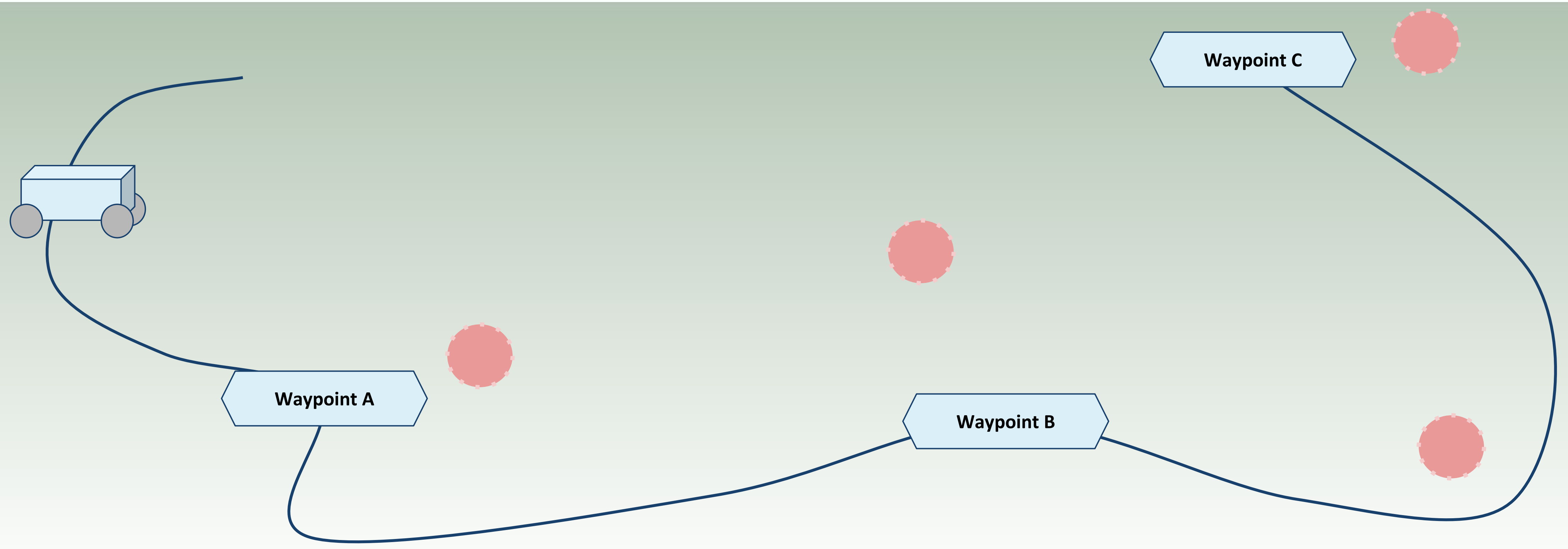
Step Three:

- Software generates pod placement locations
- Software generates rover speeds and deployment timing



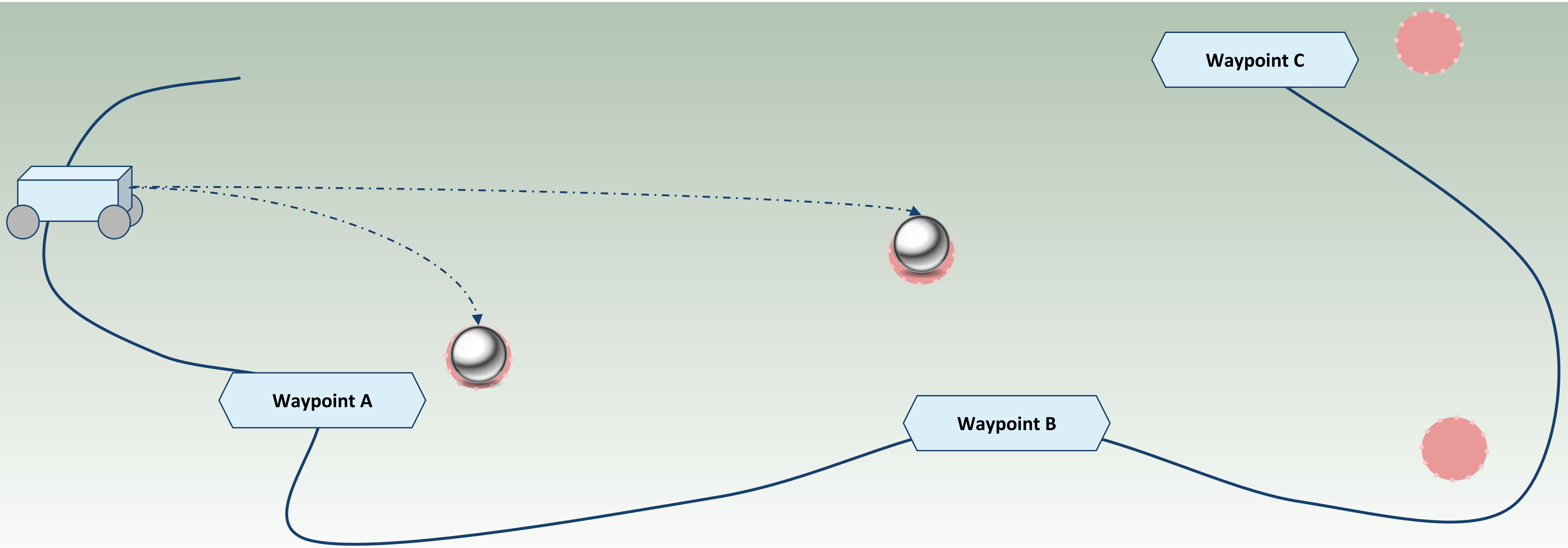
Step Four: Autonomously chooses one of three options

- Rover moves along path
- Rover deploys pod
- Rover communicates with pod(s), corrects location



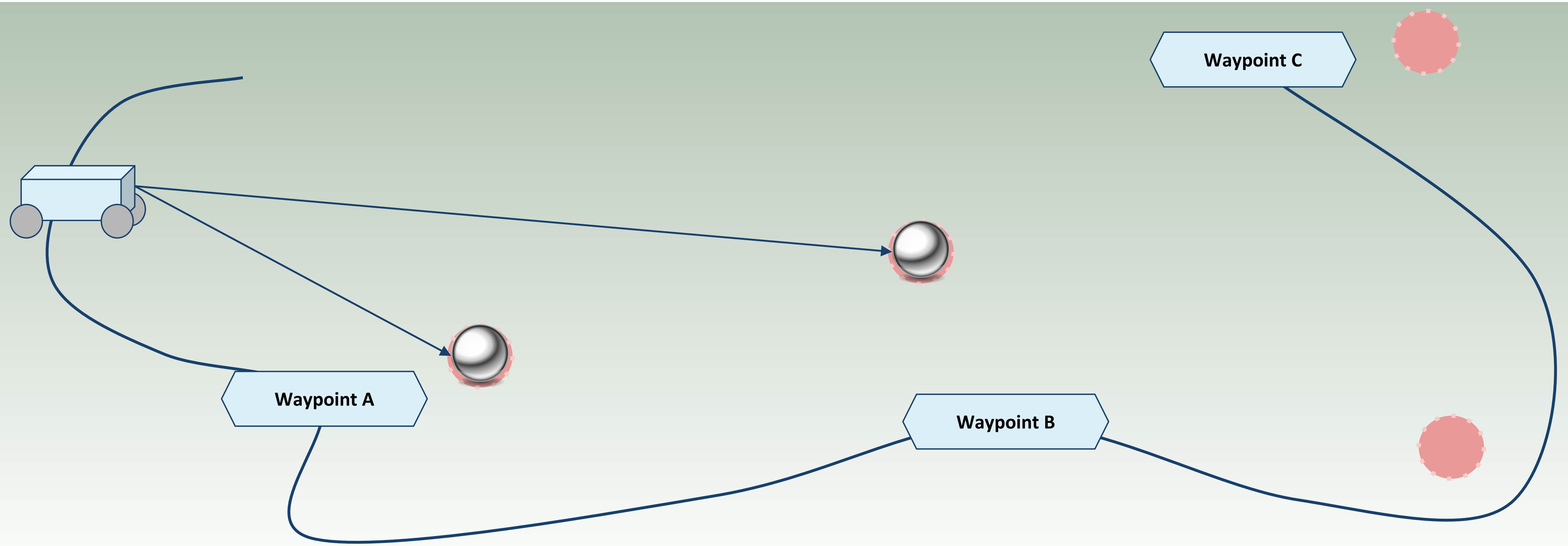
Step Four: Autonomously chooses one of three options

- Rover moves along path
- Rover deploys pod
- Rover communicates with pod(s), corrects location



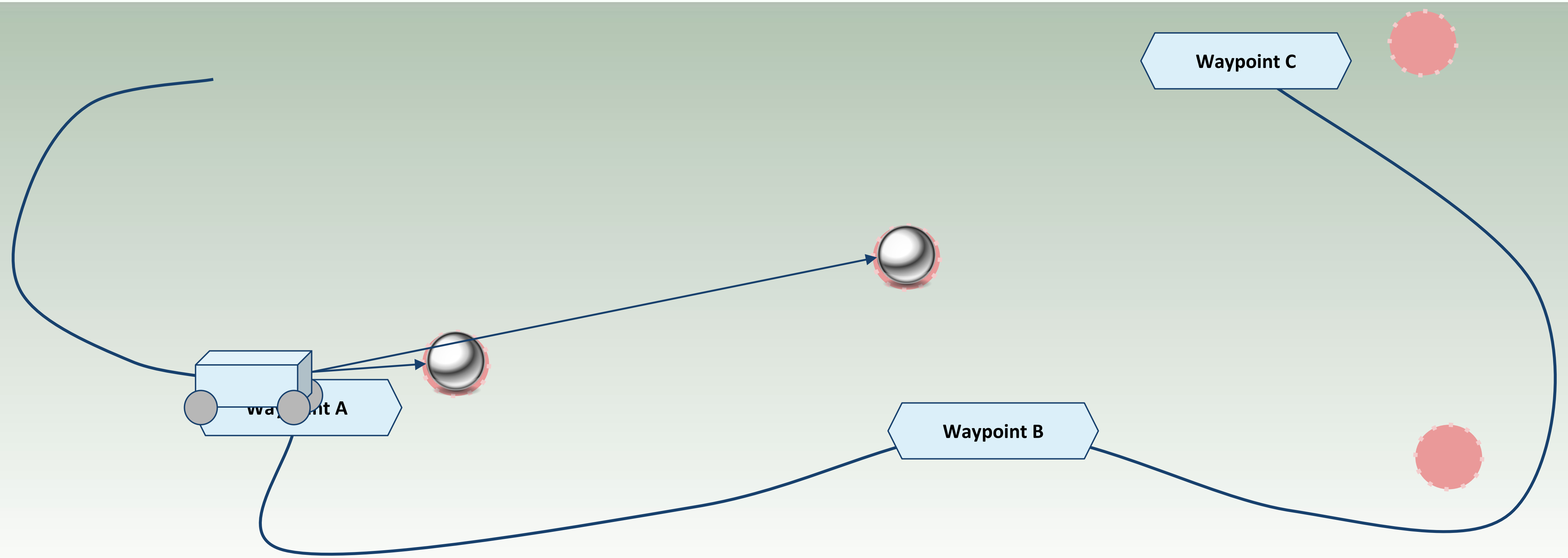
Step Four: Autonomously chooses one of three options

- Rover moves along path
- Rover deploys pod
- Rover communicates with pod(s), corrects location



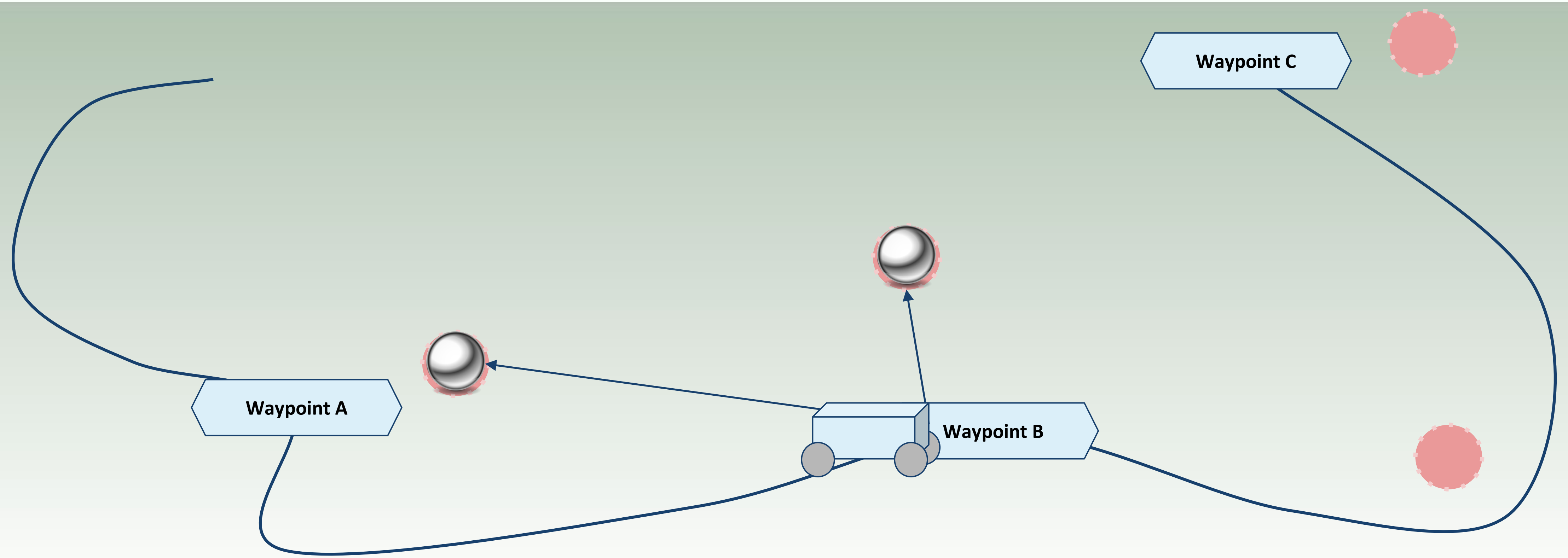
Step Four: Autonomously chooses one of three options

- Rover moves along path
- Rover deploys pod
- Rover communicates with pod(s), corrects location



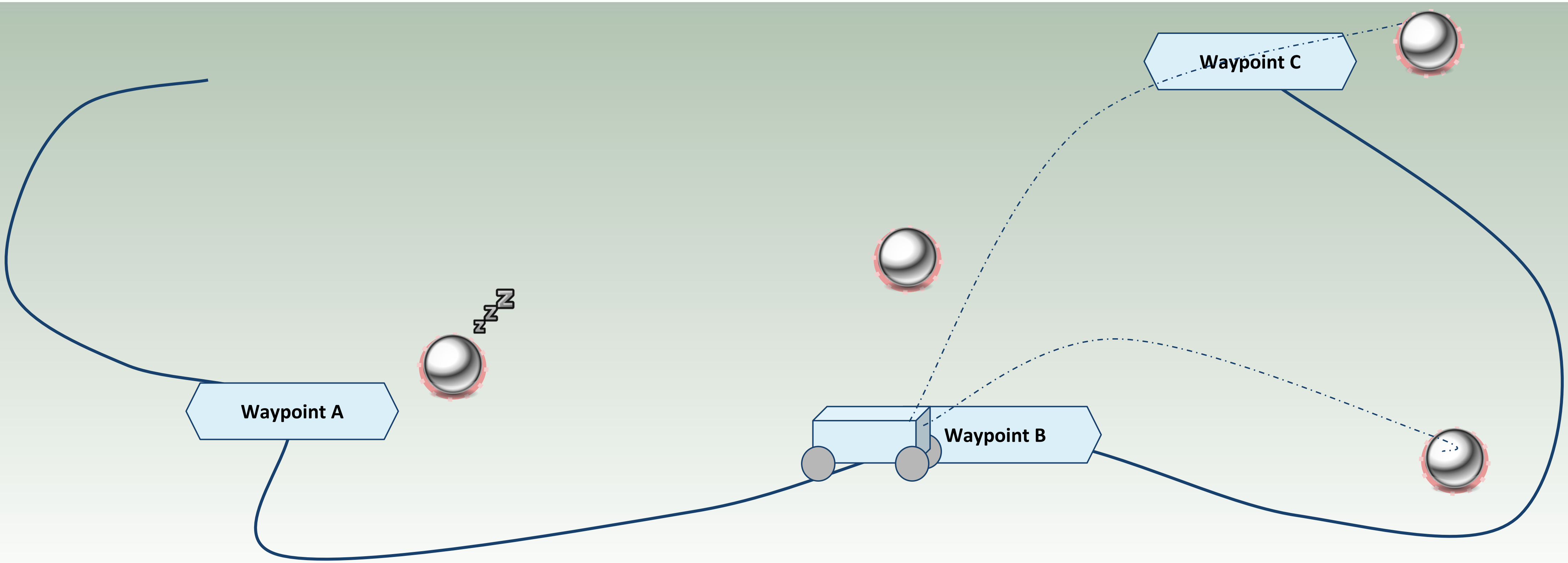
Step Four: Autonomously chooses one of three options

- Rover moves along path
- Rover deploys pod
- Rover communicates with pod(s), corrects location



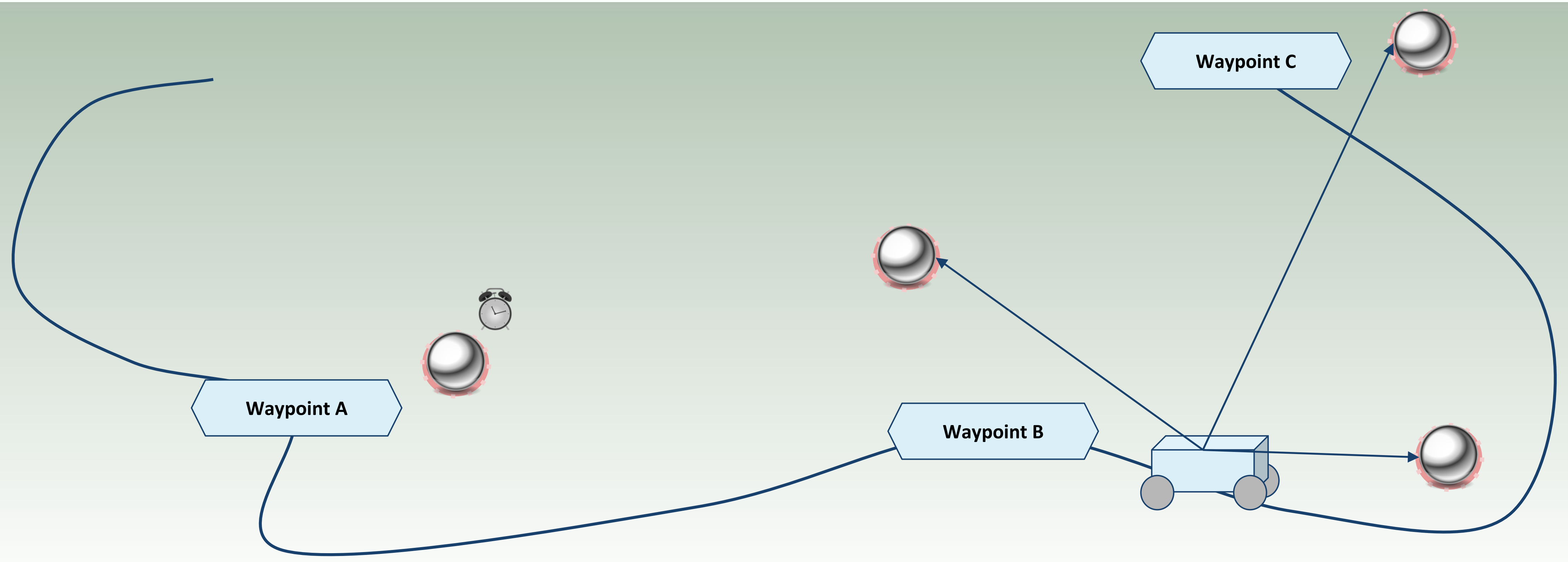
Step Four: Autonomously chooses one of three options

- Rover moves along path
- Rover deploys pod
- Rover communicates with pod(s), corrects location



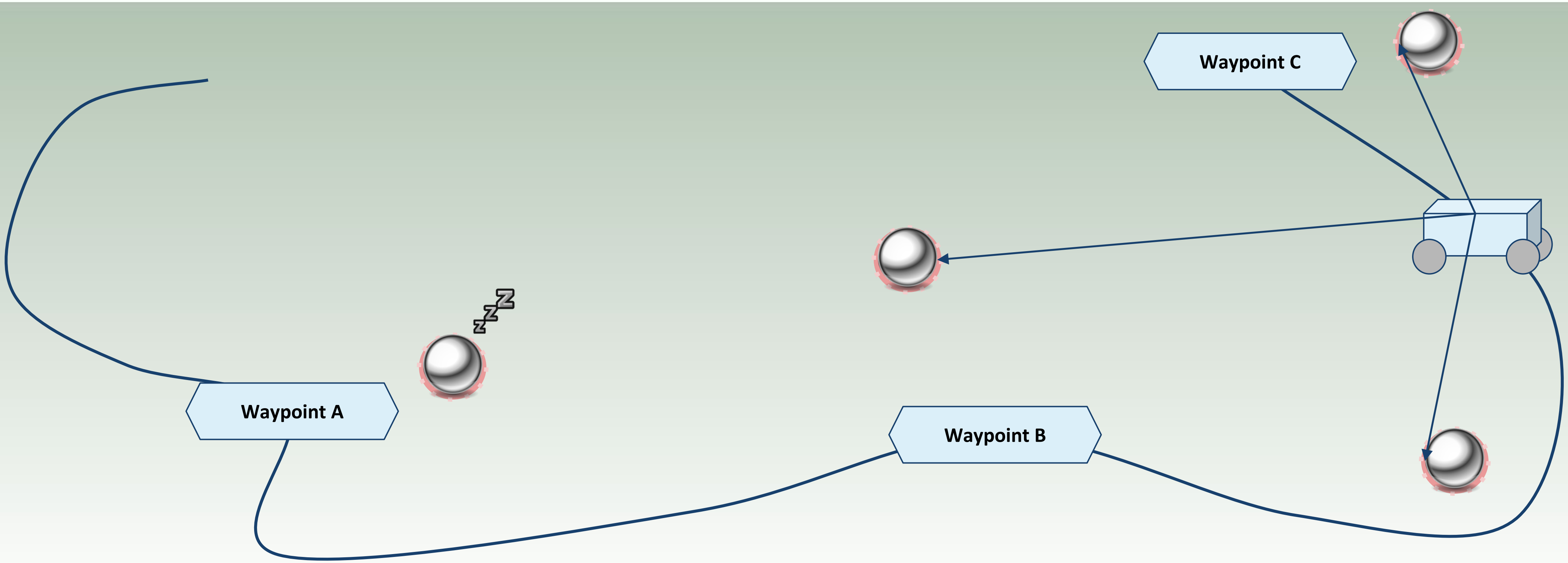
Step Four: Autonomously chooses one of three options

- Rover moves along path
- Rover deploys pod
- Rover communicates with pod(s), corrects location



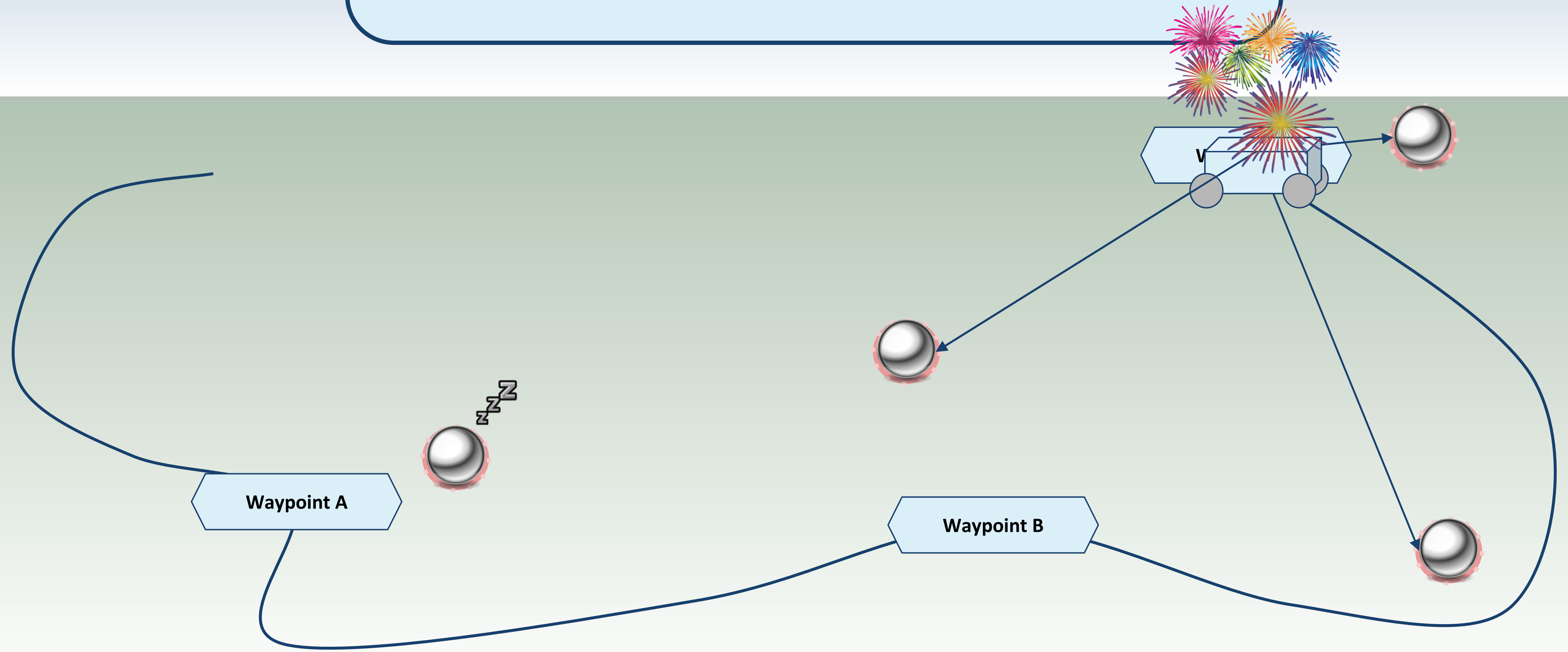
Step Four: Autonomously chooses one of three options

- Rover moves along path
- Rover deploys pod
- Rover communicates with pod(s), corrects location



Step Four: Autonomously chooses one of three options

- Rover moves along path
- Rover deploys pod
- Rover communicates with pod(s), corrects location



Four Step Summary:

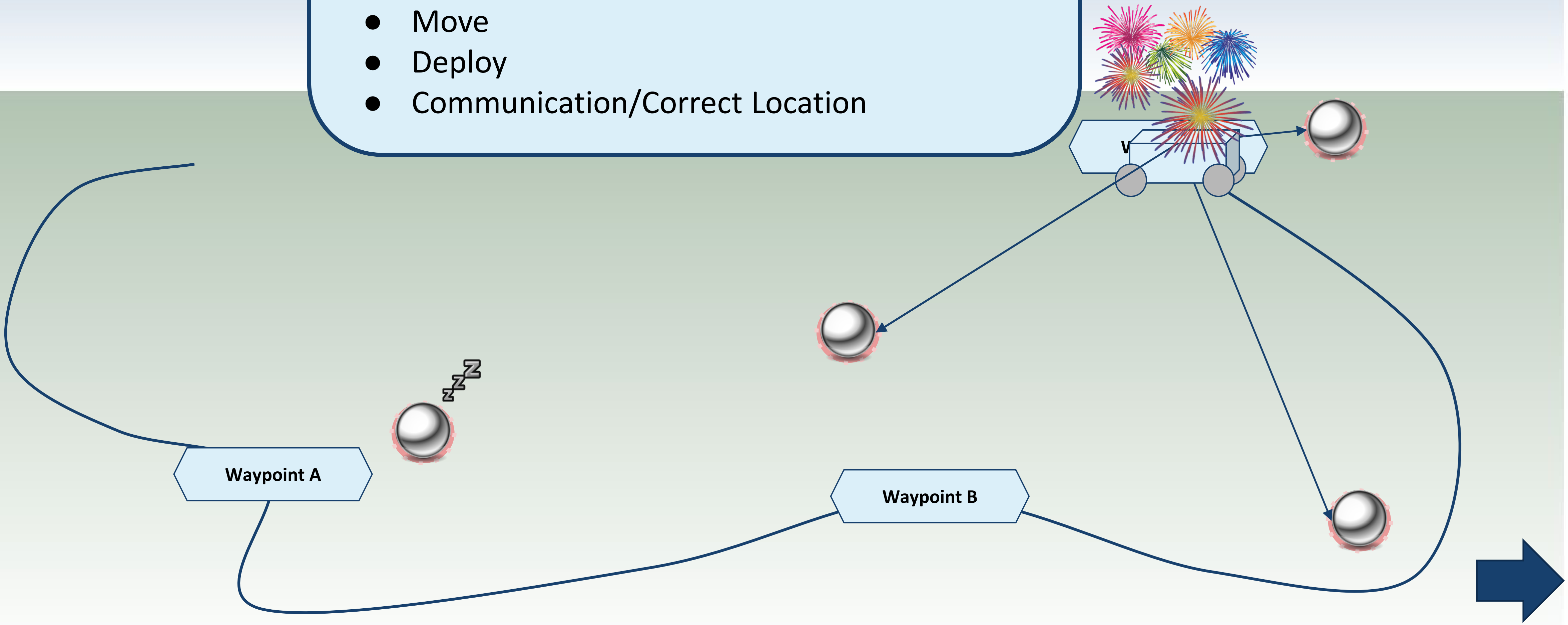
1 - User uploads MIP

2 - Ingest and collect last GPS

3 - Software predetermines path, pod locations, etc.

4 - Begin closed loop:

- Move
- Deploy
- Communication/Correct Location



B: Navigation

Path Algorithm - Proof of Accuracy (backup)

- Algorithm has been used since 1950s and the optimality has been proven before
- This assumes an admissible heuristic (never overestimates cost)
- A* designed as start to end, needs to be extended to include more than one end point

Lemma 3

Let $(n_1, n_2, \cdots, n_t)$ be the sequence of nodes closed by A^*. Then, if the consistency assumption is satisfied, $p \leq q$ implies $\hat{f}(n_p) \leq \hat{f}(n_q)$.

Proof: Let n be the next node closed by A^* after closing m. Suppose first that the optimum path to n does not go through m. Then n was available at the time m was selected, and the lemma is trivially true. Then suppose that the optimum path to n does, in fact, go through m. Then $g(n) = g(m) + h(m, n)$. Since, by Lemma 2, we have $\hat{g}(n) = g(n)$ and $\hat{g}(m) = g(m)$,

$$\begin{aligned}\hat{f}(n) &= \hat{g}(n) + \hat{h}(n)\\ &= g(n) + \hat{h}(n)\\ &= g(m) + h(m, n) + \hat{h}(n)\\ &\geq g(m) + \hat{h}(m)\\ &= \hat{g}(m) + \hat{h}(m)\end{aligned}$$

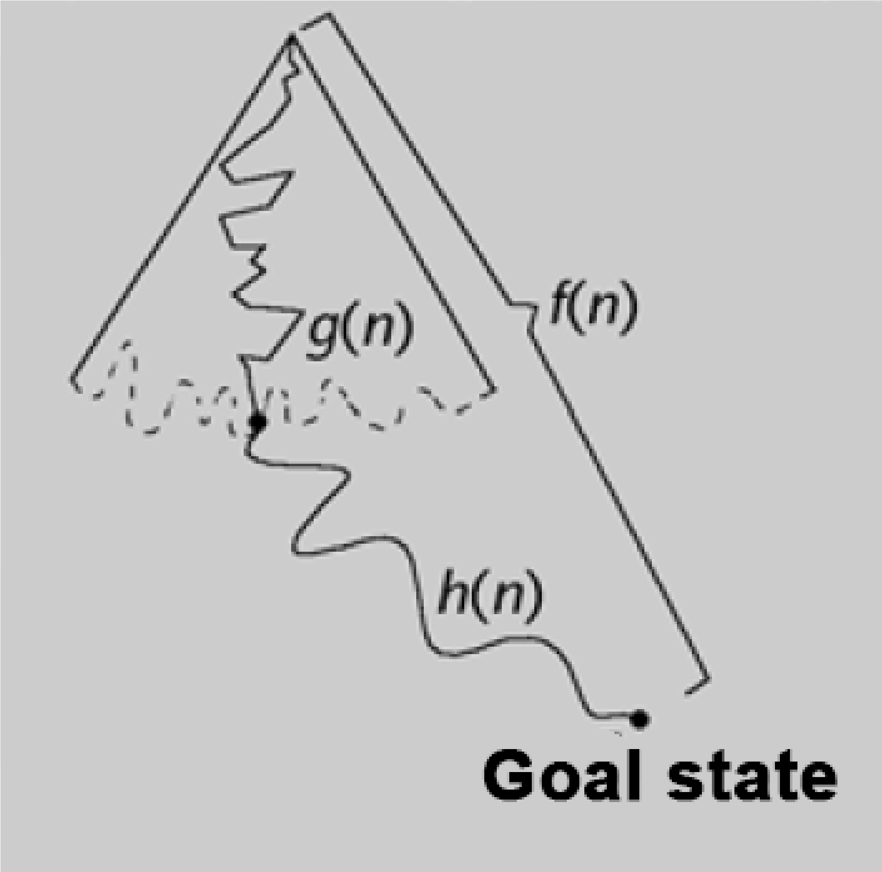
where the inequality follows by application of (5). Thus we have

$$\hat{f}(n) \geq \hat{f}(m).$$

Since this fact is true for any pair of nodes n_k and n_{k+1} in the sequence, the proof is complete.

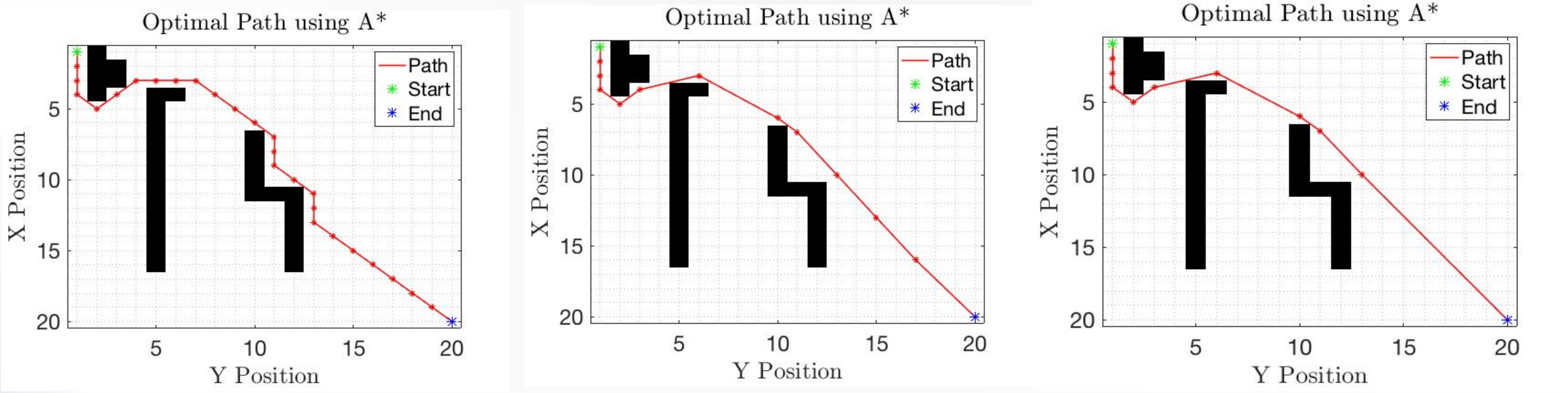
Path Algorithm - Heuristic (backup)

- Heuristic functions are “means to an end” approaches to problems that revolve around practicality
- For search algorithms, this means estimating the lowest cost from point A -> B
- Includes **Euclidean distance**, Manhattan distance (absolute difference between X and Y coordinates), etc.
- Must be admissible (underestimate of actual distance)



Path Algorithm - Neighbor Search (backup)

- Vanilla A* will search 8 nearest neighbors for least cost path
- Valid solution but provides grid-dependent paths that are not continuous
- Increasing number of searched neighbors “extends” A*’s range and allows for shorter, continuous paths to be developed that do not “hug” the grid (at cost of computation time)



20 x 20 Grids with 8, 128, and 4096 neighbors

Path Feasibility Backup- Comp Time for Dijkstra's

- A* and Dijkstra's Algorithm were final choices and deciding factor was computation time
- Dijkstra's is just a special case of A* where the heuristic cost function is set to zero

Function Name	Calls	Total Time	Self Time*	Total Time Plot (dark band = self time)
Astar_DRAGON_navigation_main	1	18.148 s	2.089 s	
pdist2	134797	15.213 s	9.191 s	

- A* Profiler output
- Note the difference in calls to pdist2

Function Name	Calls	Total Time	Self Time*	Total Time Plot (dark band = self time)
Astar_DRAGON_navigation_main	1	134.548 s	15.599 s	
pdist2	1048973	112.787 s	67.808 s	

- Dijkstra's Algorithm Profiler output (no heuristic)
- Almost 7.5x slower!

Path Feasibility Study - Path Generation Simulation (backup save)

- Requirements Addressed:

	S1.3		S - Software shall generate a path through the terrain to reach up to 10 waypoints	*The rover needs to follow some path to hit MIP desired waypoints, and must stay safe during traversal*
		S1.3.1	S - Software shall record initial GNSS position of rover	*An initial point of reference is required for inertial navigation, and to indicate starting position on MIP maps*
		S1.3.2	S - SW shall be capable of ingesting MIP, data types must be compatible	*MIP defines the mission, and so the software must be compatible in order to use it.*
		S1.3.3	S - Software path shall meet MIP time requirements, 'on-	*Per customer, mission success is not clear if the rover arrives late or early to*
M8			**The team shall use the customer-provided hardware**	*Customer requirement*
	S8.1		S - The software shall run on the rover	*Customer requirement*
		S8.1.1	S - Programming language used shall be ROS compatible/portable	*Customer requirement*

Path Feasibility Study - Path Sim Approach (backup save)

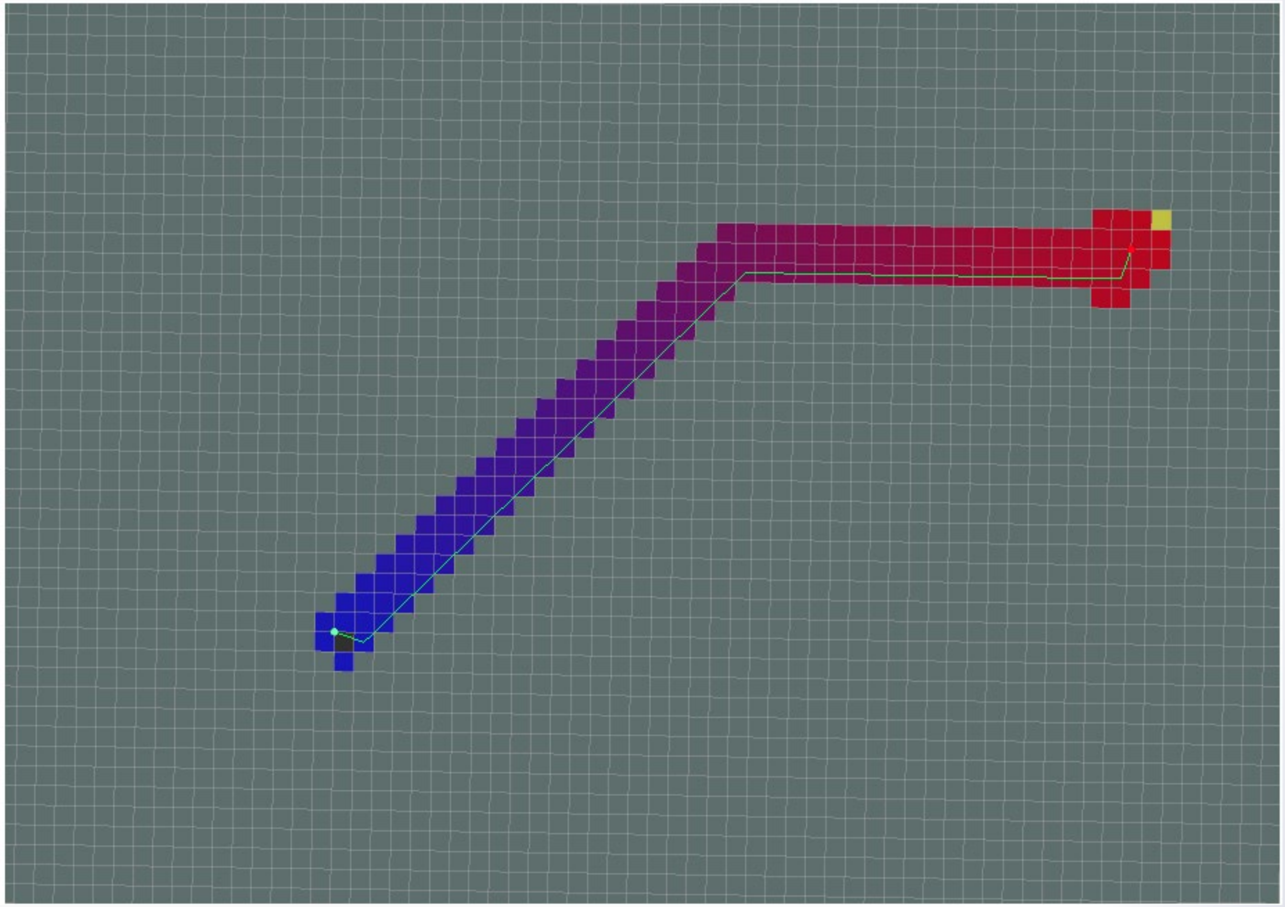
- Use high level MATLAB simulation just to show capability to generate shortest path from start to end while avoiding objects
- Assumptions/Simplifications: no science weighting, no path smoothing, no parameterized function for curve, matrix-based grid map with set values for obstacle, currently no waypoints

Backup Slide - Path Algorithm Adjustments

- Still needs to address:
  - Multiple waypoint path generation
  - Curve smoothing for actual testing (i.e., need FOS for distance from obstacles)
  - Integrating timing requirement for waypoints
  - Conversion to low level language (ROS Compatible)
- Need to ascertain format of MIP from customer and how to overlay and implement the provided maps

Path Feasibility - Rover Porting

- A* and other path planning nodes exist within ROS library `global_planner`
- Vanilla 8 neighbor search but still useful for porting from MATLAB to ROS compatible language
- Looking to convert MATLAB code directly to Python for adaptability (MATLAB API also exists)



ROS A* Example

Path Feasibility - Onboard Computer

- Using MATLAB, maximum memory used at one time was never greater than 1.06 MB, well within available memory on rover
- MATLAB is much more costly in terms of RAM due to UI, low level ROS compatible languages will not be nearly as expensive
- Plenty of heritage and help offered from COHRINT team in porting and implementing autonomously on Jackal

COMPUTER	Standard		Performance	
	Celeron J1800 Dual core, 2.4GHz 2 GB RAM	WIFI Adapter 32 GB Hard Drive	Intel Core i5 4570T Dual core, 2.9GHz 4 GB RAM	WIFI Adapter 128GB Hard Drive

Jackal Datasheet Specs

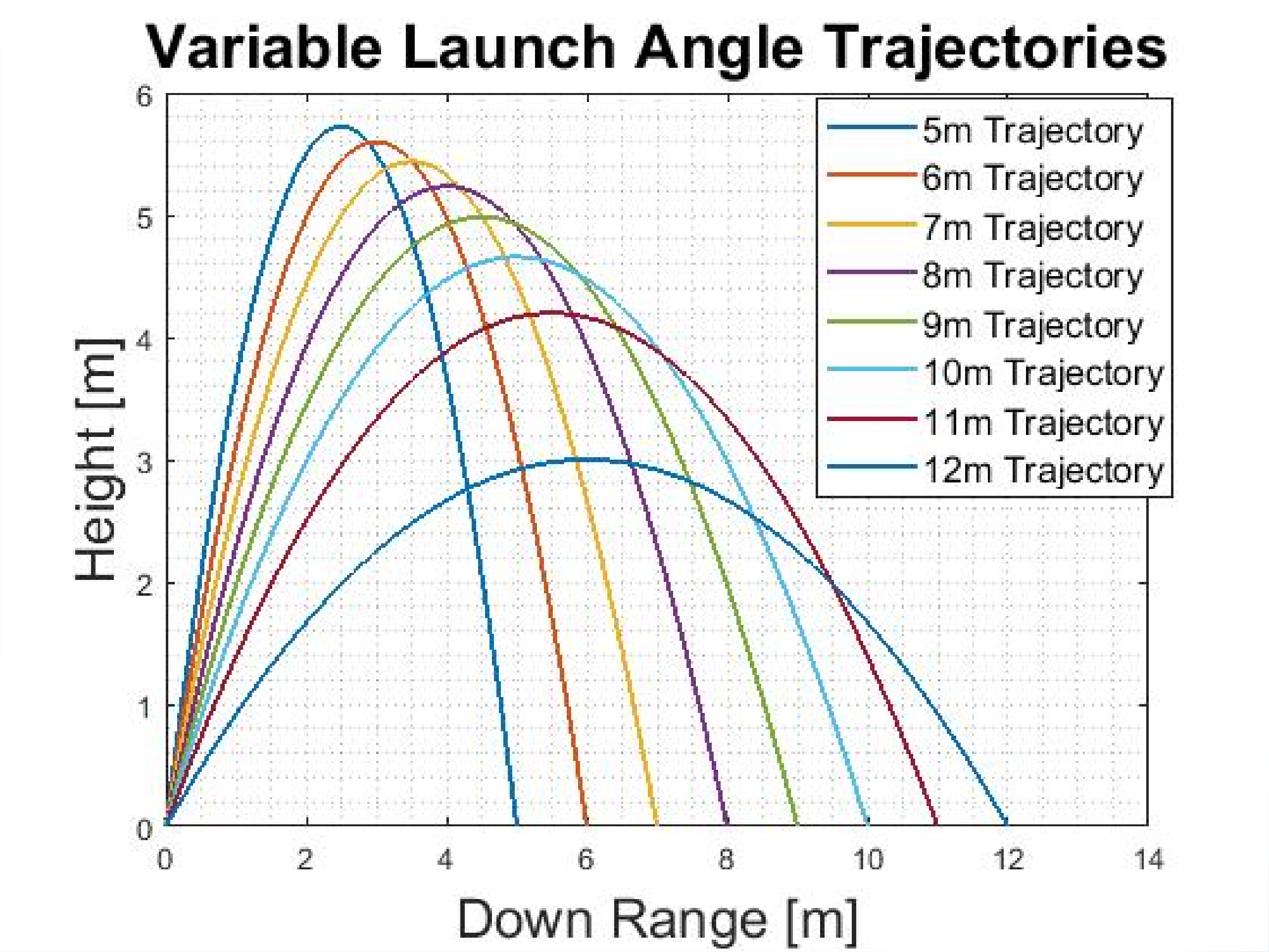
C: Deployment

CONOPS

1. Receive angle and distance to deploy pod from navigation subsystem team
2. Pod is loaded into “deployment tube”
3. Rover stops
4. On pod deployment controller, compute azimuth and elevation angle
5. Command motor to turn to azimuth angle
6. Command motor to turn to elevation angle
7. Command linear actuator to pull back spring until it reaches release mechanism
8. Release mechanism unlatches and deploys pod
9. Linear actuator moves back into relaxed position and latches into release mechanism
10. Pod is reloaded into “deployment tube”
11. Repeat

Trajectory Model

Distance	Launch Angle
5m	77.68°
6m	75°
7m	72.16°
8m	69.09°
9m	65.70°
10m	61.78°
11m	56.78°
12m	45°



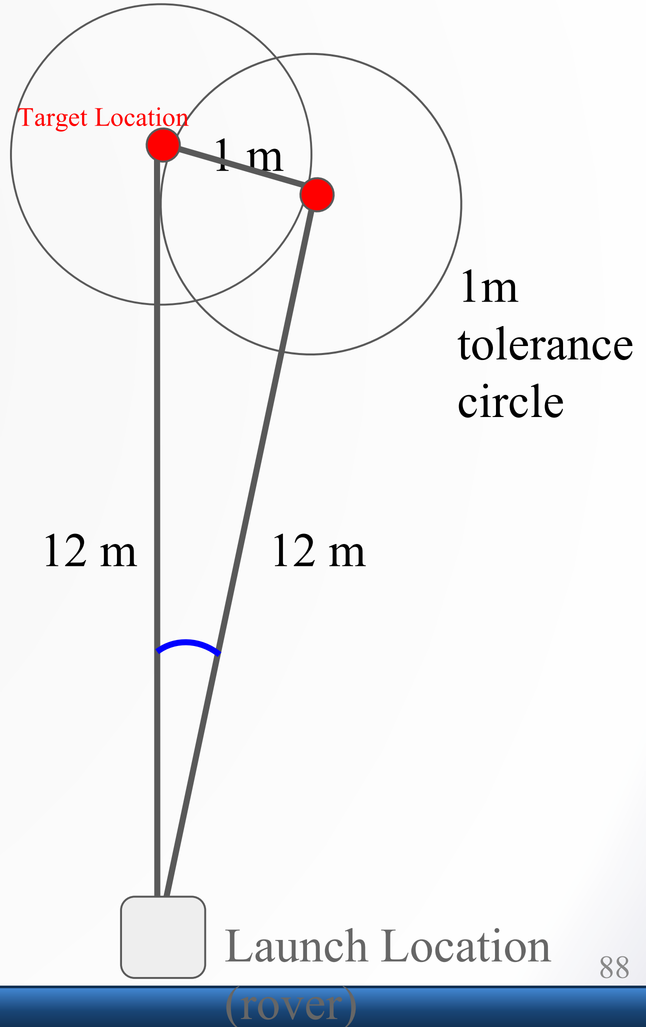
Azimuth Angle Adjustment

D3.3 DM shall have deployed pods land within 1m radius accuracy of SW commanded location

$$\theta = \cos^{-1}\left(\frac{a^2+b^2-c^2}{2ab}\right)$$

a = 12
b = 12
c = 1

Required Angle: 4.78°



Linear Actuator Cost and Force

Linear Actuator	Force	Cost	Weight	Stroke
PA-17	850lbs	$305.00	11.55lbs-13.25lbs	4”-8”
Figelli Automations	1000lbs	$219.99	4.5lbs-5.5lbs	3”-9”
LACT-1000APL	1000lbs	$425.95-$431.95	10.2lbs-11.3lbs	4”-8”

Pitching Machine

Drawbacks

- Weight requirements
- Modeling difficulty
- Inaccurate assumptions
- Friction forces
- Rotational energy imparted on the pod

Pitching Machine Analysis

Assumptions:

- Conservation of Energy
- Spinning wheels decrease to launch velocity
- Wheels are cylinders with constant density

ρ = 1200 kg m²

t = 8 cm

r_w = 10 cm

m_w = 3.02 kg (each)
ω_i= 463 rpm

$$2KE_i = 2KE_{0,w} + KE_{0,p}$$

$$I_w\omega_i^2 = I_w\omega_0^2 + \tfrac{1}{2}m_pV_0^2$$

$$I_w\omega_i^2 = I_w(r_wV_0)^2 + \tfrac{1}{2}m_pV_0^2$$

$$I_{cyl} = \tfrac{1}{2}m_wr_w^2 = \tfrac{1}{2}(\pi r_w^2t\rho)r_w^2$$

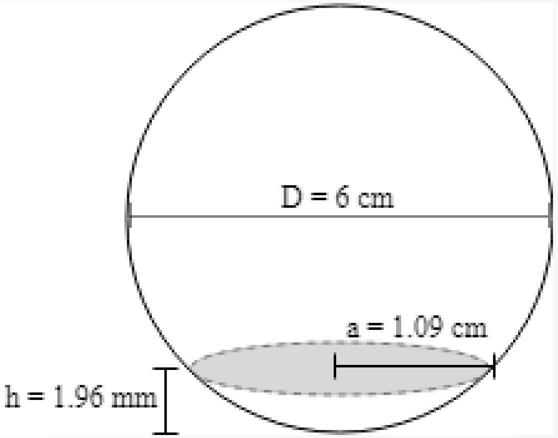
$$\omega_i = \sqrt{(r_wV_0)^2 + \frac{m_pV_0^2}{2I_w}}$$

$$\omega_i = V_0\sqrt{r_w^2 + \frac{m_p}{\pi\rho t r_w^4}}$$

Feasibility - Landing Force

Stress Equations

$$\sigma = \frac{F}{A} = \frac{F}{\pi a^2}$$

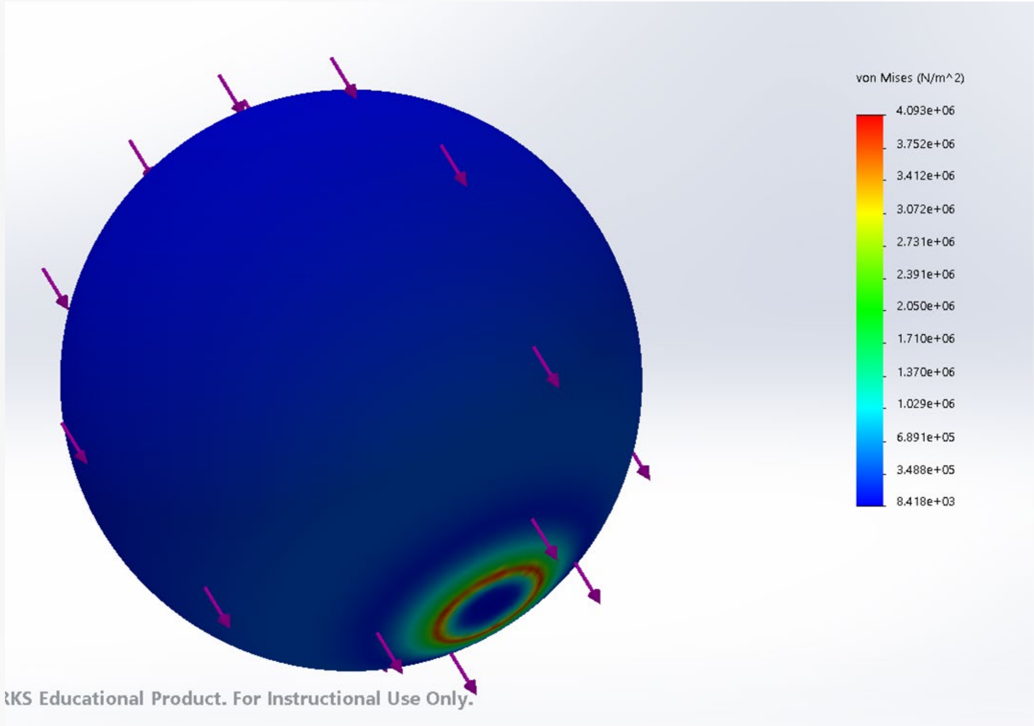


F = 244 N

A = 3.7325 cm^2

σ = 653 MPa = 94.7 psi

SolidWorks Stress Analysis



σ_{max} = 4.093 MPa

Landing Stress Analysis

Material Selection Analysis

Material	Average Yield Strength [MPa]
Polycarbonate	56.5 MPa
Polyimide	103 MPa
Polysulfone	95.2 MPa

Functional Requirement Met

Requirement	Addressed with...	Met?
M3: DM shall have the following range: No less than 5m, at least 10m, no more than 20m	Deployment method, trajectory models	✓
M4: The rover and ground inputs shall prevent damage to all hardware systems (To be addressed in CDR)	Reloading mechanism, safety mechanism	✓
M5: The pods shall function as RF navigation beacons and as environmental data monitors, to the rover	Landing/stopping mechanism, stress analysis, material selection	✓

Other Deployment Options

- Drone fleet
  - Drones are equipped with all necessary sensors and boards
  - Drones sit on launch platform attached to rover
  - When signal is received a single drone will fly up, point in the calculated azimuth direction, and fly to determined range
  - The drone will land and begin functioning as an RF beacon
- Pneumatic cannon
  - Similar to spring canon in aiming and actuation
  - Uses pressurized chamber to deploy
  - **Added complexity and weight** is unnecessary for this project

Pod Reloading

D - DM shall be capable of deploying 10 pods within a 20 minute (TBR) duration

D - DM shall be capable of reloading and deploying a new pod every 2 minutes

Potential Concepts

- Magazine
- Gravity fed
- Link fed system

Finer Details Dependent Upon Mechanism Structure

- Exact location for reloading mounting system
- Organization of pods in the reloading mechanism

Safety

D - DM shall have a mechanical safety inhibit

- Cap to cover cannon when not in use
- Procedures in place on proper operation and stowage
- When not in use, have a pin that prevents deployer from latching to linear actuator

D - DM shall have a remote safety inhibit, such as an arm/disarm system, to enable safe approach to rover

- Software immediately stops linear actuator and then begins to move towards decompressed location

D - DM shall have indicated keep out range/FOV

- Sign/sticker to warn users to never stand near the front of the cannon

Deployment Controls

D - DM shall have interface with rover SW for commanding deployment mechanism to deploy pods

- Rover SW provides bearing and distance for desired landing location
- Pod deployment controller calculates azimuth and launch angle
- Controller commands step motors
- Step motors rotate gears
- Azimuth and launch angle are achieved and pod is launched

D - DM shall not interfere with rover's GNSS or pod communication

- Barrel and reloading mechanism will be constructed from material that does not interfere with GNSS or pod communication

Weight Budget

Object	**Weight [lbs] (Total Estimates)**
Linear Actuator	6
Cannon Barrel	4
Base Platform	4
Motors	3
Reload Mechanism	4
Gears	3
Pods	20
Total	**44**

D: POD

Customer assistance with PCB design

- Initially after the trade studies from the CDD, PCB design was out of scope.
- After further discussion with our customer, he has agreed to help with the design and provide tutorials to aid PCB design.
- Hardware and software development plan allows for descope to use Localino v2.0 and breakouts if PCB design fails.

Pod Electronics Design Plan

Hardware

Use Localino v2.0, sensor suite, accelerometer, and SD card reader reference designs to develop PCB with help from customer.

Software and Buy Out

Test sensor suite, accelerometer, GPS, and SD card reader with Localino 2.0

Test ranging and data transmission/reception using Localino v2.0. Continue developing provided software/firmware

Transfer existing software and power system to PCB hardware.

↓

Test. Verify.
Verify. Test.

↓

Procure additional PCB/Hardware for 10 pods total.

↓

Full system testing

↑

'Buy Out' date:
Feb 1 2019

PCB Timeline

10/12 Begin PCB Drawings

+3 Weeks

11/2 Finalize Initial PCB Design

+1 Week

11/9 Receive v1.0 Board

+2 Weeks

11/23 Fix bugs and order new PCB

+1 Week

11/30 Receive v2.0 Board

+2 Weeks

12/14 Fix bugs and order new PCB

Winter Break Debug and order v3.0

1/18 Receive v3.0 Board

+1 Week

1/25 Fixed bugs and order v4.0

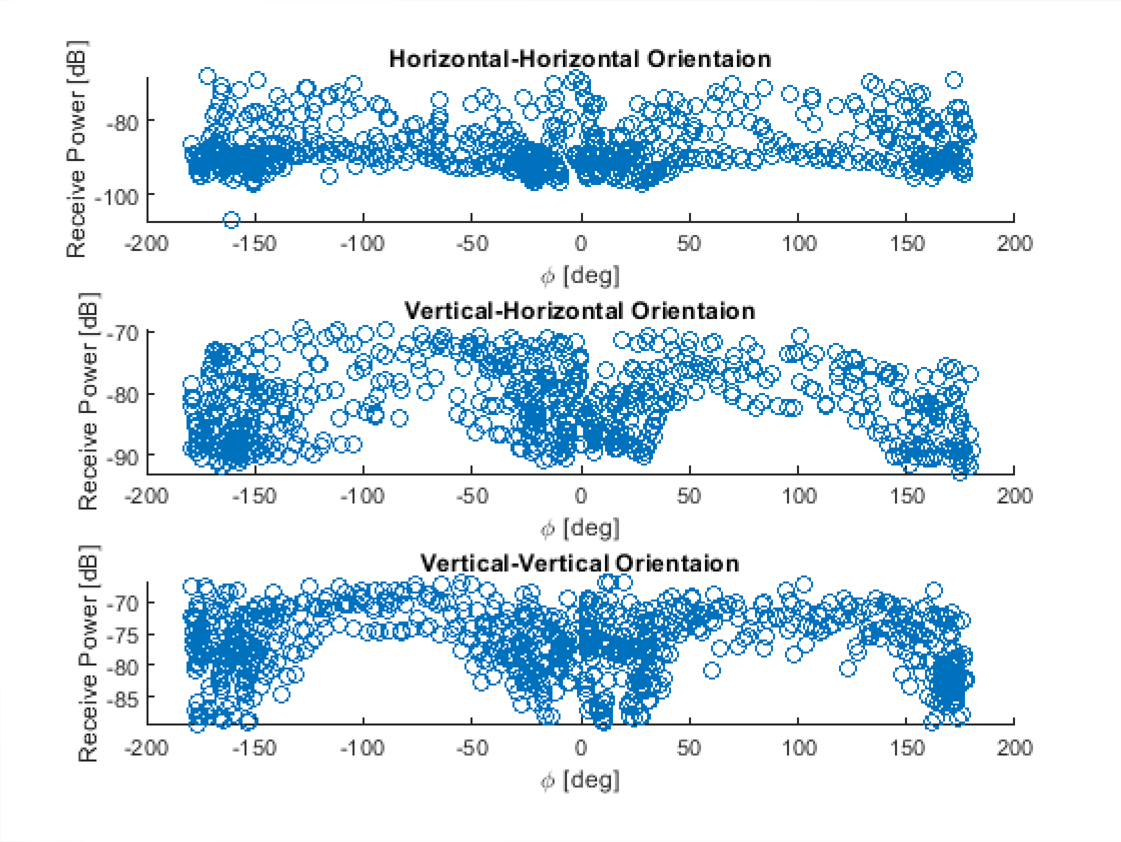
+1 Week

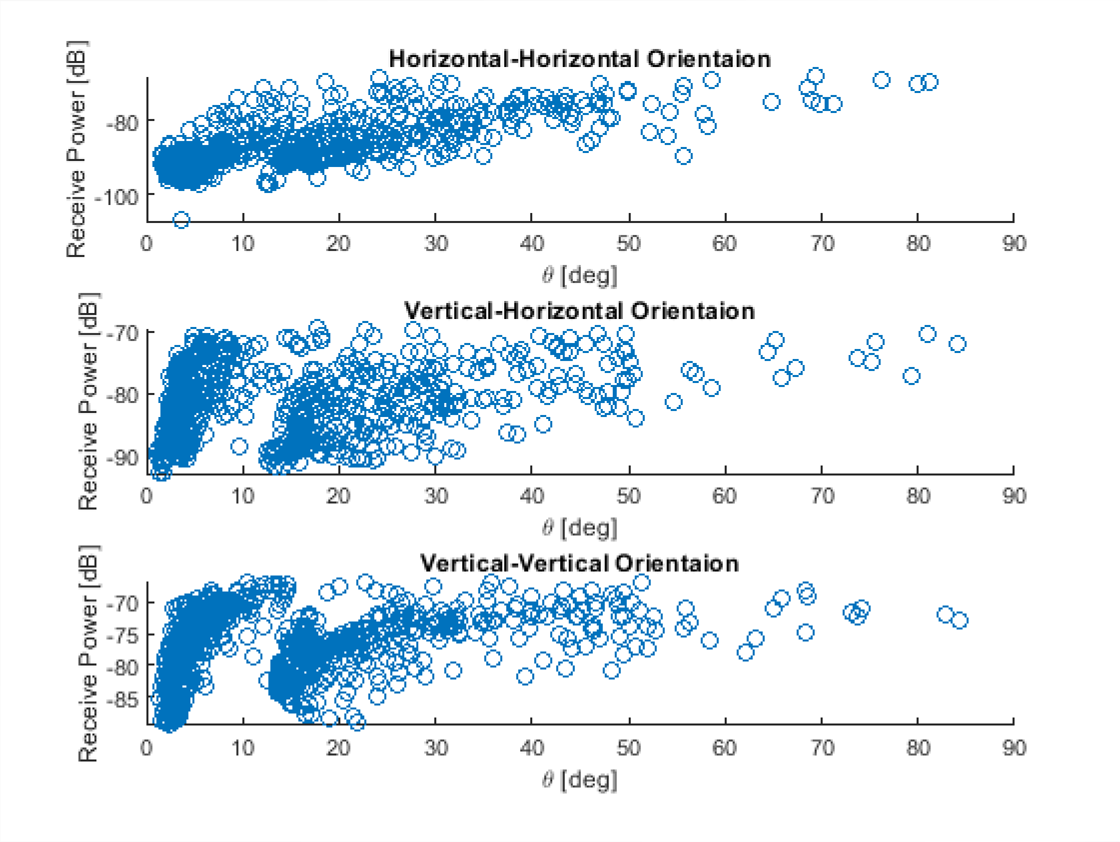
2/1 Receive Final Board and if it doesn’t work order Localino Boards

Provided RF testing

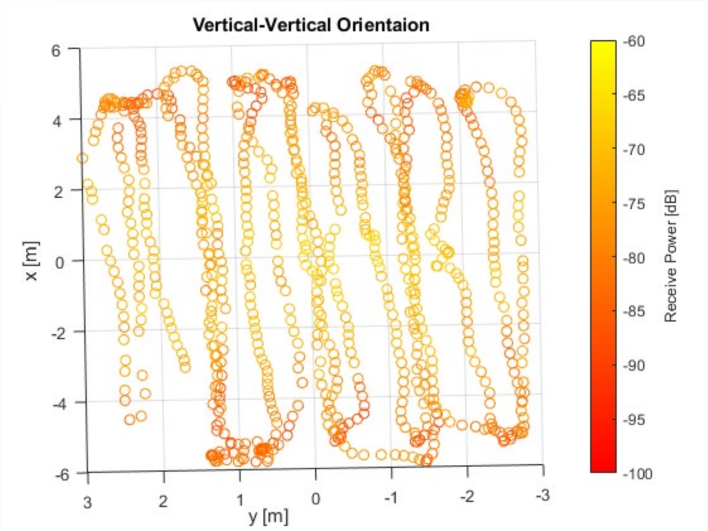
- A full github repo is being provided with developed software and firmware to jumpstart Localino v2.0 development.
- Provided software and firmware runs off ROS which will be compatible with the rover
- DW1000 antenna transmit pattern is provided
- DW1000 transmit range from 20 m to 90 m depending on orientation of the antennas

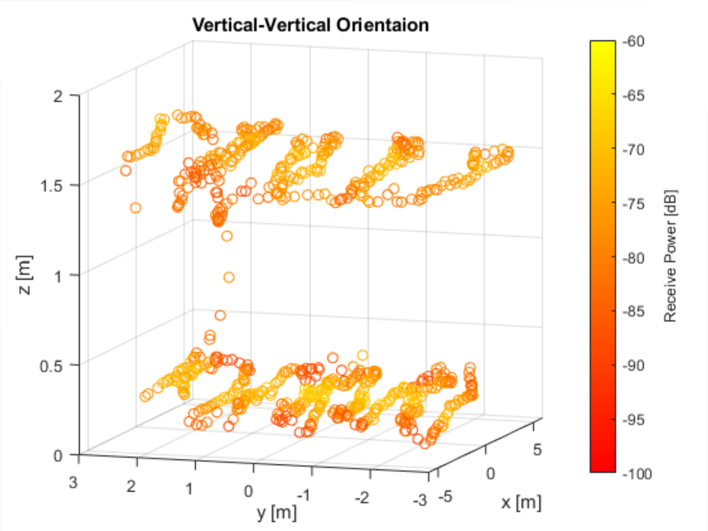
Antenna Transmit Pattern

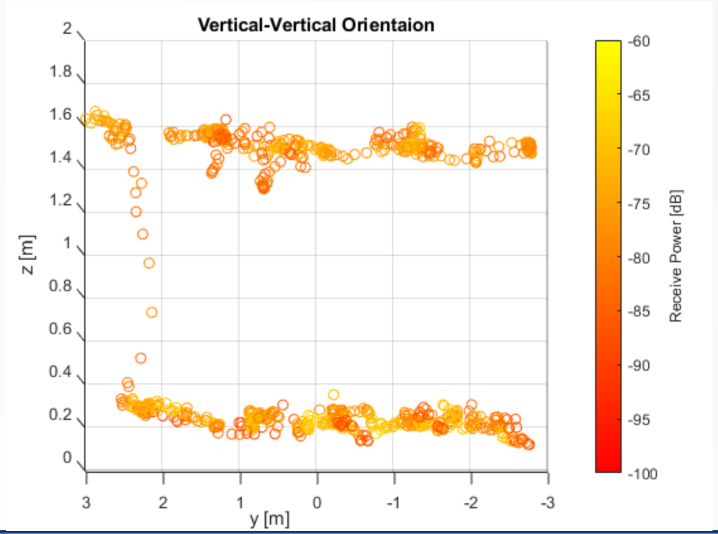


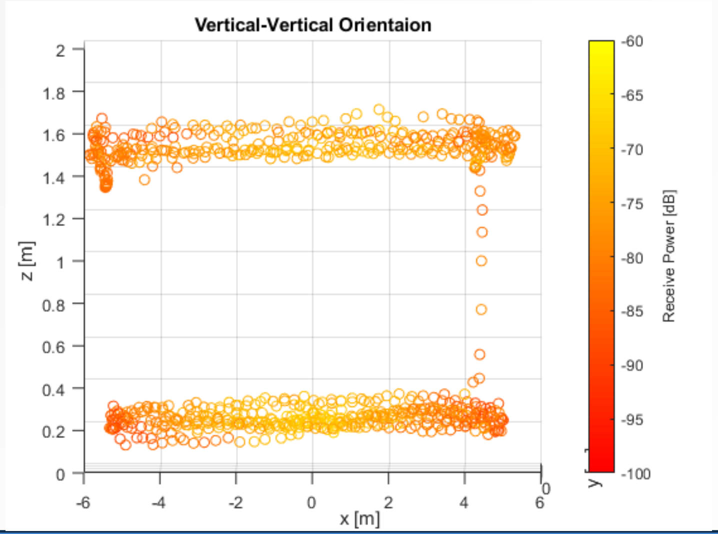


Antenna Pattern Test Setup







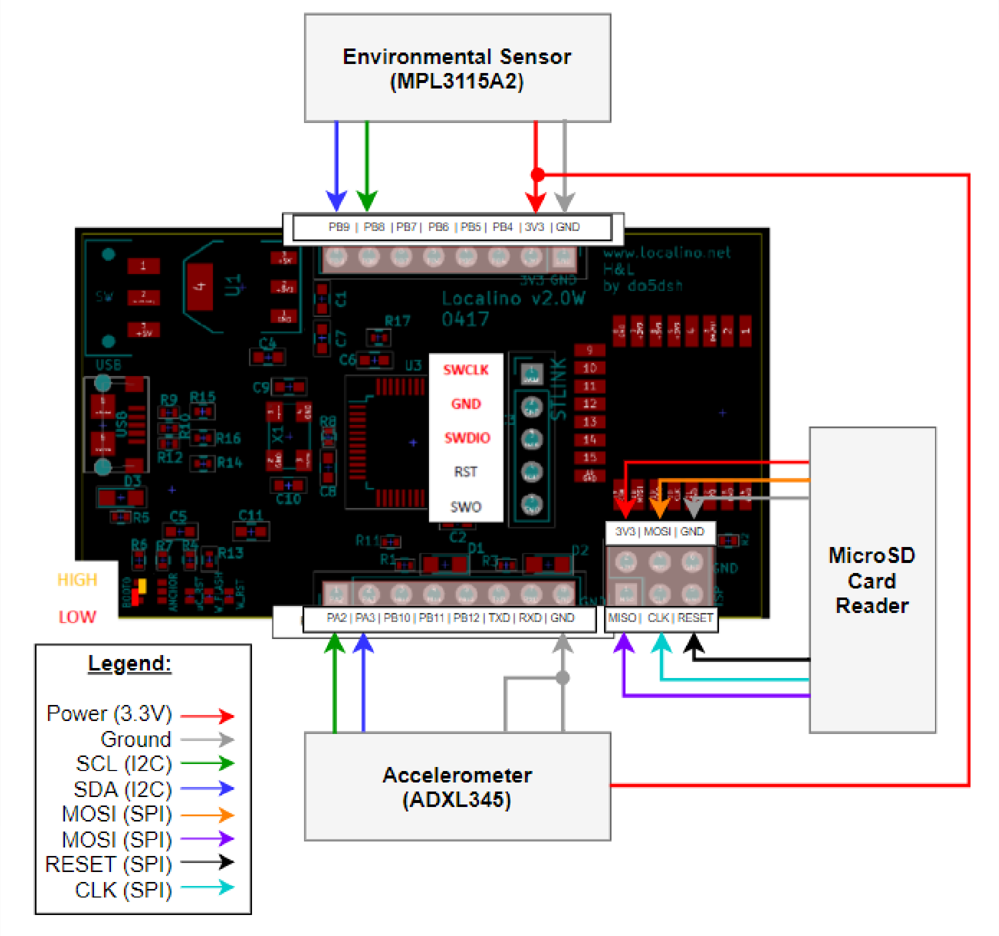


Interfacing Design

Requirements:
P6.1 - Pod shall have environmental sensor package
P6.1.1 - Temperature Sensor
P6.1.2 - Altimeter
P6.1.3 - Accelerometer

Design Approach:
Integrate environmental accelerometers, and SD Card Reader breakout boards onto the Localino microcontroller and write data to the SD card to prove interface feasibility/.

Rationale:
Prove that all sensors and SD card reader can be integrated and function as needed. Also have heritage with the environmental sensor suite and MicroSD card reader.



P6.1 - Pod shall have environmental sensor package ✓

Pin Labels on Localino

I2C	
Pin	**Usage**
3v3	Power
GND x2	GND
PA2	USART_TX, 12-bit ADC, Timer
PA3	USART_RX, 12-bit ADC, Timer
PB4	Timer, SPI_MISO
PB5	Timer, SPI_MOSI, I2C_SMBAI
PB6	I2C_SCL, Timer, USART_TX
PB7	I2C_SDA, Timer, USART_RX
PB8	Timer, I2C_SCL
PB9	Timer, I2C_SDA
PB10	I2C_SCL, USART_TX, Timer
PB11	I2C_SDA, USART_RX, Timer
PB12	SPI_NSS, I2C_SMBAI, USART, Timer

SPI	
Pin	**Usage**
3v3	Power
GND	GND
MOSI	SPI Connection
MISO	SPI Connection
RESET	Reset
CLK	Clock

Chosen Component Datasheets Sensors

ENVIRONMENTAL SENSOR:

MPL3115A2 (Temperature/Pressure/Altititude):
https://cdn.sparkfun.com/datasheets/Sensors/Pressure/MPL3115A2.pdf

ACCELEROMETER SENSOR:

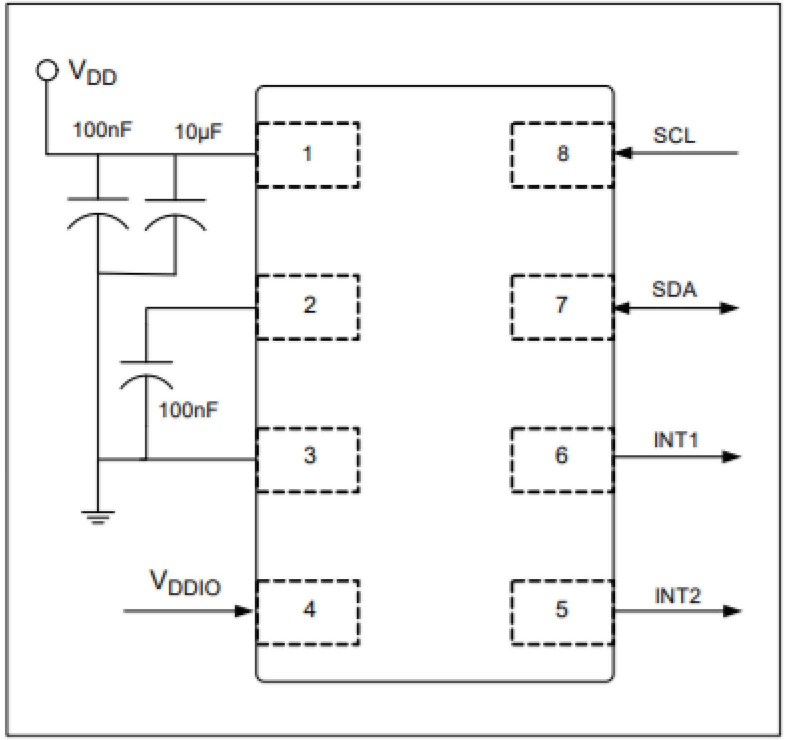
ADXL345 (+/- 16g Digital): https://www.sparkfun.com/datasheets/Sensors/Accelerometer/ADXL345.pdf

SD CARD READER:

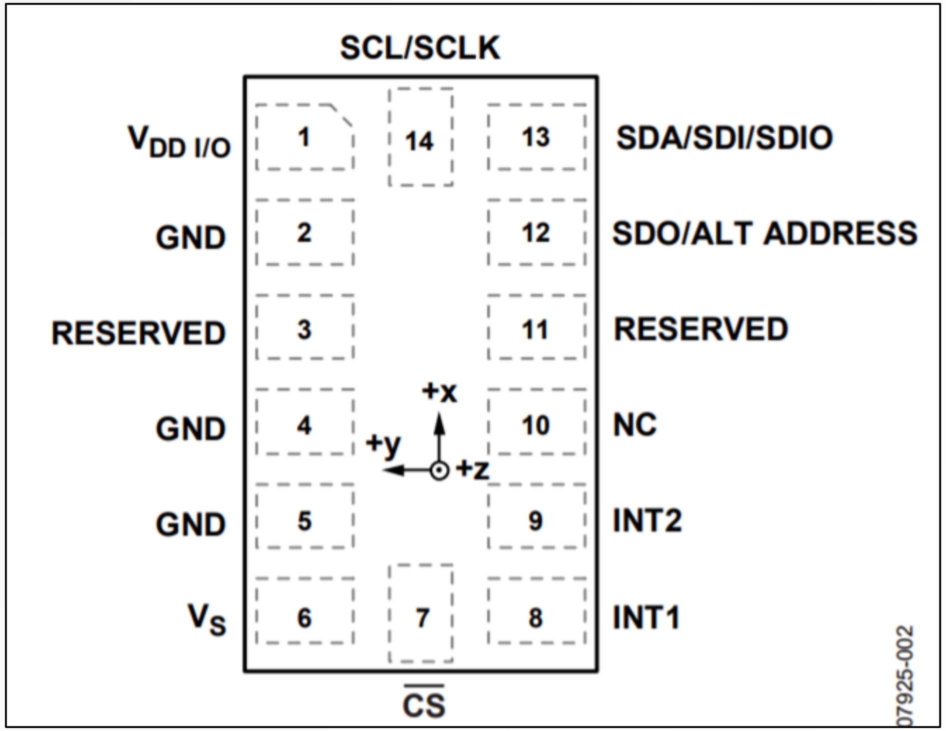
Micro SD Card Reader (Adafruit): https://cdn-learn.adafruit.com/downloads/pdf/adafruit-micro-sd-breakout-board-card-tutorial.pdf

Pinout on Sensor Suite and Accelerometer

Environmental Sensor (MPL3115A2)



Accelerometer Sensor (ADXL345)



Chosen Environmental Suite Electrical Characteristics

MPL3115A2

Table 3. Electrical Characteristics @ VDD = 2.5V, T = 25°C unless otherwise noted.

Ref	Symbol	Parameter	Test Conditions	Min	Typ	Max	Unit
1	V_{DDIO}	I/O Supply Voltage		1.62	1.8	3.6	V
2	V_{DD}	Operating Supply Voltage		1.95	2.5	3.6	V
3	I_{DD}	Integrated Current 1 update per second	Highest Speed Mode Oversample = 1		8.5		µA
4			Standard Mode Oversample = 16		40		
5			High Resolution Mode Oversample = 128		265		
6	I_{DDMAX}	Max Current during Acquisition and Conversion	During Acquisition		2		mA
7	I_{DDSTBY}	Supply Current Drain in STANDBY Mode	STANDBY Mode selected SBYB = 0		2		µA
8	VIH	Digital High Level Input Voltage SCL, SDA		0.75			V_{DDIO}
9	VIL	Digital Low Level Input Voltage SCL, SDA				0.3	V_{DDIO}
10	VOH	High Level Output Voltage INT1, INT2	I_O = 500 µA	0.9			V_{DDIO}
11	VOL	Low Level Output Voltage INT1, INT2	I_O = 500 µA			0.1	V_{DDIO}
12	VOLS	Low Level Output Voltage SDA	I_O = 500 µA			0.1	V_{DDIO}
14	T_{ON}	Turn-on time(1)	Highest Speed Mode(2)			60	ms
			Highest Resolution Mode(3)			1000	
16	T_{OP}	Operating Temperature Range		-40	25	+85	°C

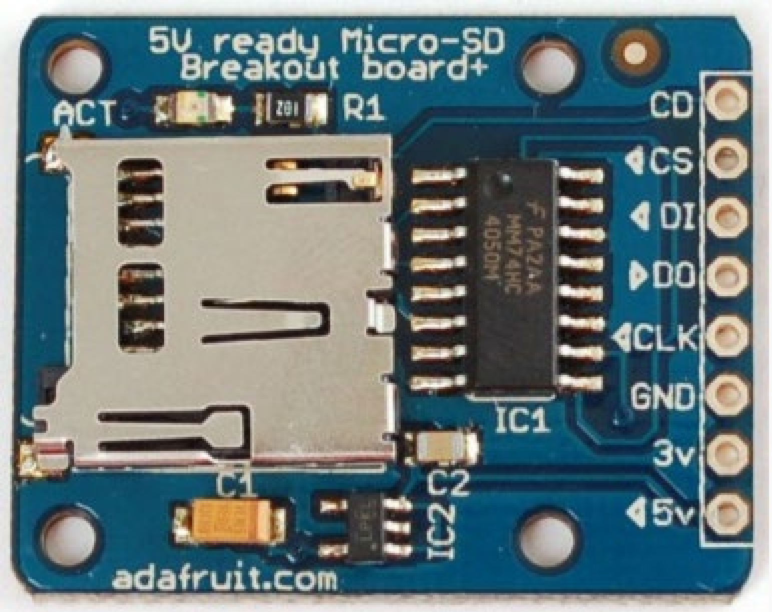
Chosen Accelerometer Electrical Characteristics

ADXL345

POWER SUPPLY					
Operating Voltage Range (V_S)		2.0	2.5	3.6	V
Interface Voltage Range ($V_{DD\ I/O}$)	$V_S \leq 2.5$ V	1.7	1.8	V_S	V
	$V_S \geq 2.5$ V	2.0	2.5	V_S	V
Supply Current	Data rate > 100 Hz		145		µA
	Data rate < 10 Hz		40		µA
Standby Mode Leakage Current			0.1	2	µA
Turn-On Time[5]	Data rate = 3200 Hz		1.4		ms
TEMPERATURE					
Operating Temperature Range		−40		+85	°C
WEIGHT					
Device Weight			20		mg

SD Card Reader pinout and characteristics

Localino to MicroSD
Card Reader Wiring



→ NC
→ RESET
→ MOSI
→ MISO
→ CLK
→ GND
→ NC
→ 3.3V

NOTE: Onboard ultra-low dropout regulator that convert voltages ranging from 3.3V-6V down to ~3.3V

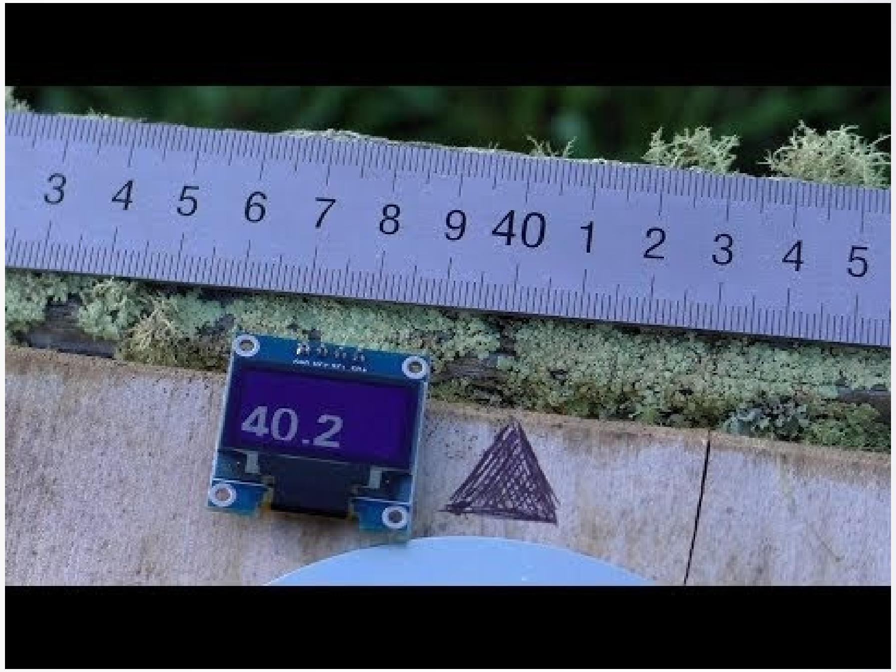
CD - Card Detect Pin - Shorts to ground when a card is not inserted. Not necessary for operation of reader.

Verification Feasibility – GPS Solution

Requirements:
G1.1.1 - GNSS accuracy shall be 1m or less

Design Approach:
Purchase Ublox C94-M8P Kit
Post-test, hand-place receiver next to each pod and record location
Raw GPS data is corrected by vendor-provided RTK (Real Time Kinematic) to provide centimeter level accuracy

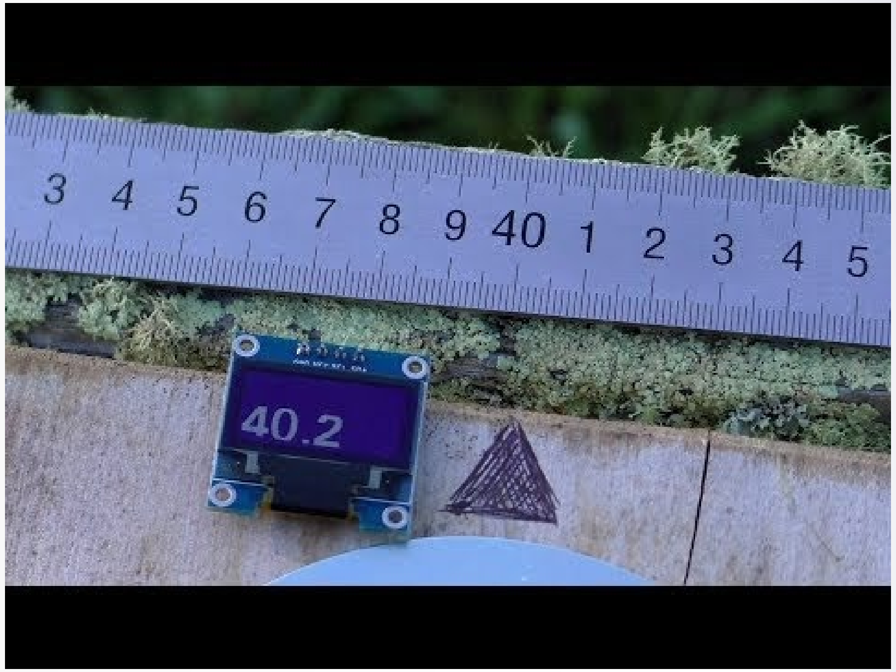
Rationale:
Crucial to verify results
Satisfies requirement of obtaining GPS location of pod to cm accuracy
Data collection post-test is acceptable and meets data analysis needs
Avoid extra size/mass introduced by components, jumpers, wires, and breakout boards
Avoid additional complication including implementation of our own RTK correction
Price is competitive, cost of single Ublox kit is on par with price of ten rtk-gps modules



Verification Feasibility – GPS Solution

Demonstration:
Vendor provided demonstrations can be found online, once purchased can perform landmark test for verification

Conclusion:
This is an obvious choice for the team, reducing GPS functionality from requiring 10 installations and a custom RTK solution to 1 single COTS pre-packaged solution will reduce workload while not evading any requirements nor impacting other subsystems.



G1.1.1 - GNSS accuracy shall be 1m or less	✓

Trade Study Datasheet References (Environmental Sensors)

BME680 (Air Quality/Temperature/Humidity/Pressure/Altitude) : https://ae-bst.resource.bosch.com/media/_tech/media/datasheets/BST-BME680-DS001-00.pdf

CCS811/BME280:

CCS811 (Air Quality):
https://cdn.sparkfun.com/assets/learn_tutorials/1/4/3/CCS811_Datasheet-DS000459.pdf

BME280 (Temperature/Humidity/Pressure/Altittude):
https://cdn.sparkfun.com/assets/learn_tutorials/4/1/9/BST-BME280_DS001-10.pdf

Trade Study Datasheet References (Accelerometer Sensors)

ADXL377 (+/- 200g Anolog Sensor):
https://cdn.sparkfun.com/datasheets/Sensors/Accelerometers/ADXL377.pdf

MMA8452Q (+/- 2, 4, 8g Digital Sensor):
https://cdn.sparkfun.com/datasheets/Sensors/Accelerometers/MMA8452Q-rev8.1.pdf

State Estimation Monte Carlo with Radial Error and compounding error over mission time/rover movement

https://drive.google.com/file/d/1F4js4WjIPjW0bSAoFkRklraXYxKOr9jg/view