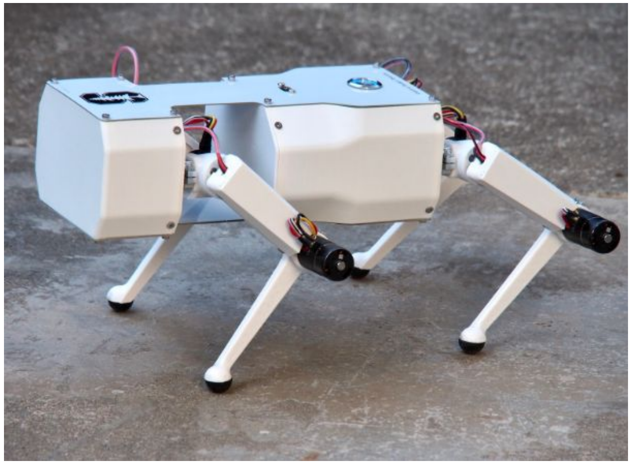
Brandeis University

Stanford PupperV2: *Visual Obstacle Detection, Boundary Generation, and Navigation*

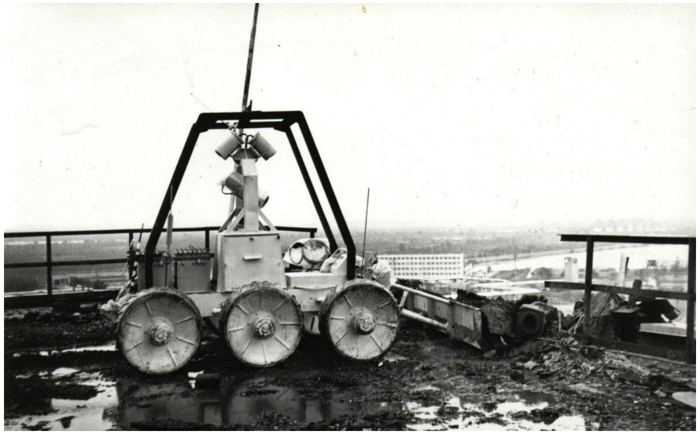
Arjun Albert and Alexion Ramos



Index

1. Context
2. Hardware
3. Limitations
4. Fiducial Obstacle Detection
5. Boundary Generation
6. Motion Planning
7. Controls
8. Conclusion

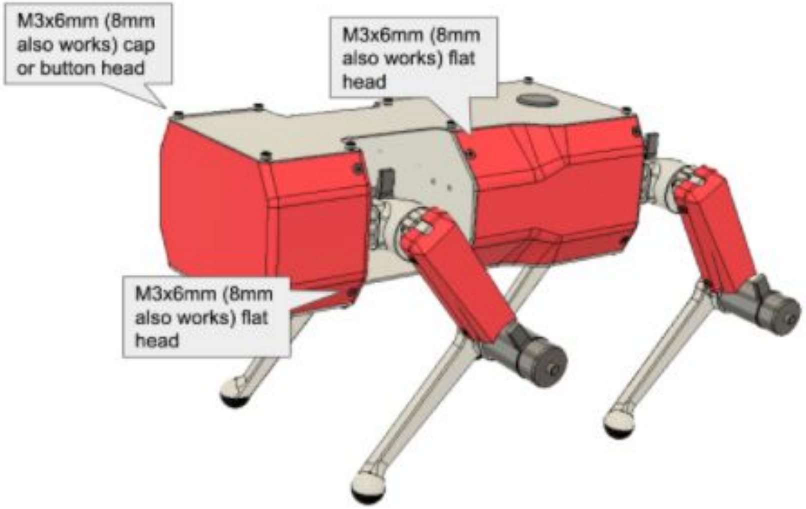
Context

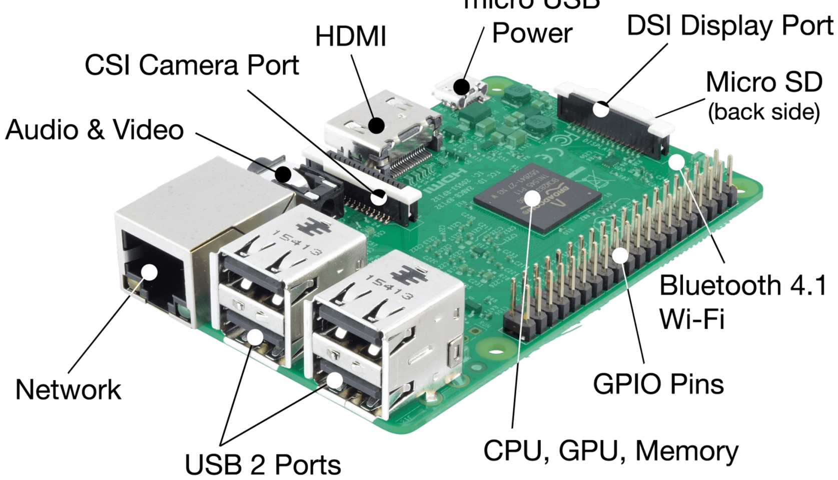


- Applications
  - Mapping and exploration
  - Rescue missions
  - Animatronics
  - Maintenance and Inspection
- Issues
  - Complex dynamics
  - Irregular movement
  - Low weight capacity

Hardware

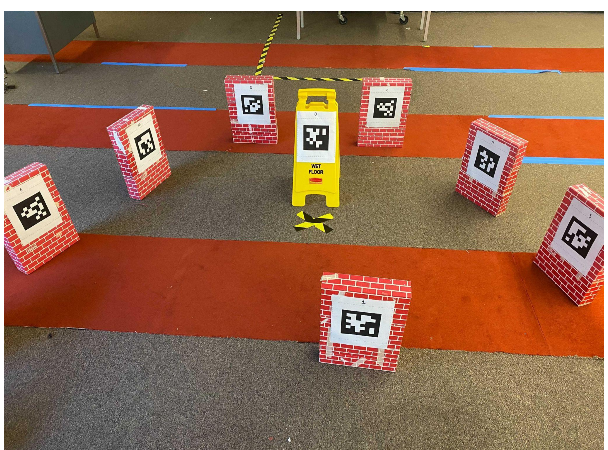
- Brushless motors
- RaspberryPi
- Raspicam
- No IMU, LiDAR, GPS





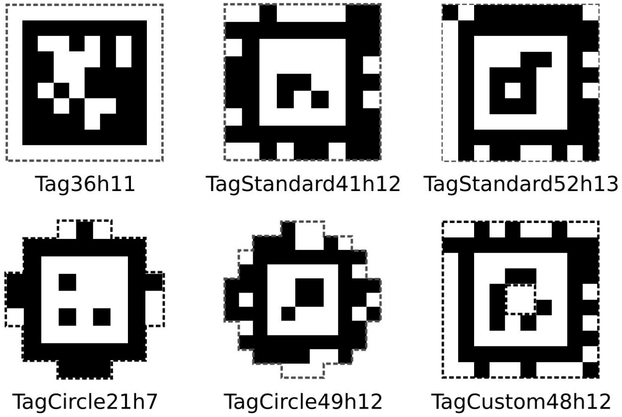
Limitations

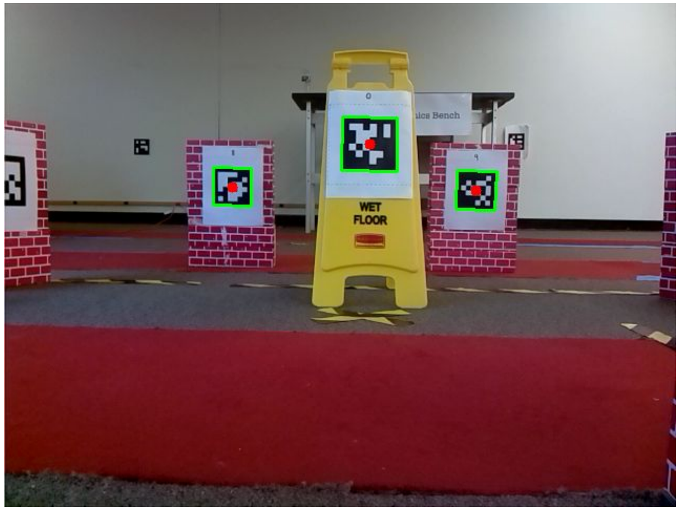
- No ROS
- Limited compute power
- No localization
- Lack of controller software interfaces
- Unknown and rugged environment



Fiducial Obstacle Detection

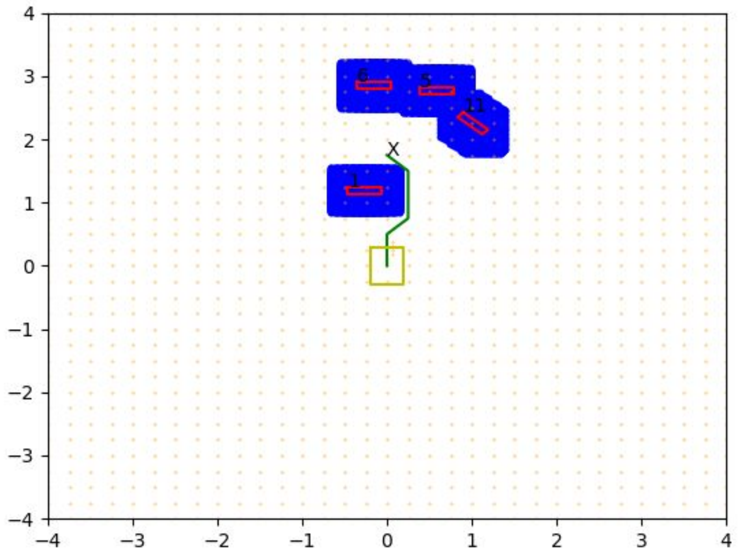
- Calibration
  - Input camera parameters
  - Print matching fiducials
- Detection
  - Pupil AprilTags
  - Raspicam software
- Transform
  - Rotation Matrix
  - Translation Matrix
  - Coordinate frame offset





Boundary Generation

- Polygon representation
  - Interpolation
- Configuration space
  - Minkowski sum
  - Asymmetric ego agent
  - Rotation matters



$$A = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$$

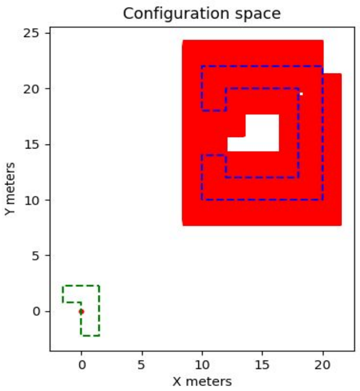
$$B = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$$

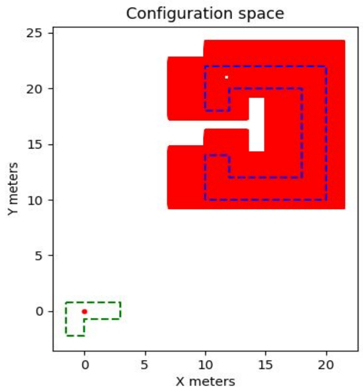
$$A + B = \{\mathbf{a} + \mathbf{b} \,|\, \mathbf{a} \in A,\ \mathbf{b} \in B\}$$

Brandeis University

Boundary Generation Algorithms

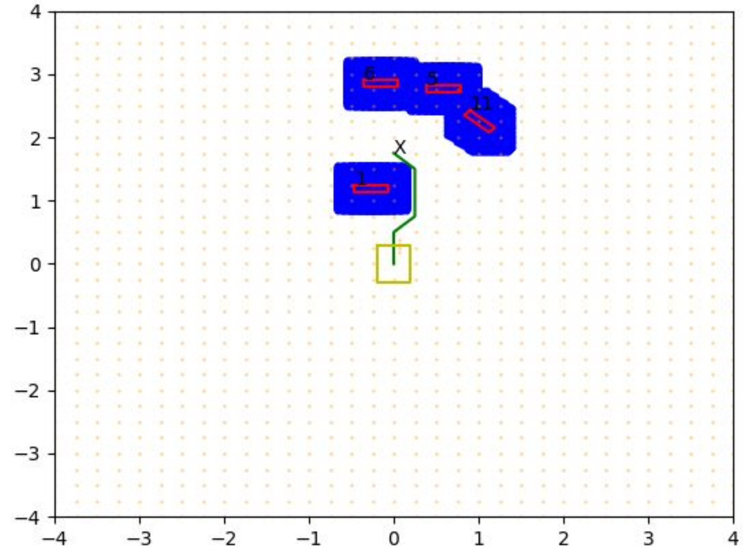
Data: [A] geometric representation of ego robot and [B] geometric representation of environment obstacles
Result: [C] configuration space matrix
i = 0
initialize [C]
while $i < N$ **do**
  apply negative rotation to [A]
  C=[C] | $[A] + [B]$
  i++
end





Data: [A] with (2, N) geometric representation of ego robot and [B] with (2, K) geometric representation of environment obstacles where N < K
Result: [C] configuration space matrix
L = N
while $N < K$ **do**
  append (NULL, NULL) to [A]
end
i = 0
initialize [C]
while $i < N$ **do**
  apply negative rotation to [A]
  C=[C] | $[A] + [B]$
  i++
end
current step = (0,0)
matrix step = (-1, -2)
initialize {C}
j = 0
while $j < L$ **do**
  append C[current step] to {C}
  current step = current step + matrix step
  current step = current step % N
  j++
end

Motion Planning



- Functionality
  - Obstacle inflation
  - Dijkstra graph search
  - Node deletion
  - Euclidean distance as cost
- Performance
  - Fast collision checking
  - Discretization tuning
  - Planning horizon vs. graph density tradeoff

Controls

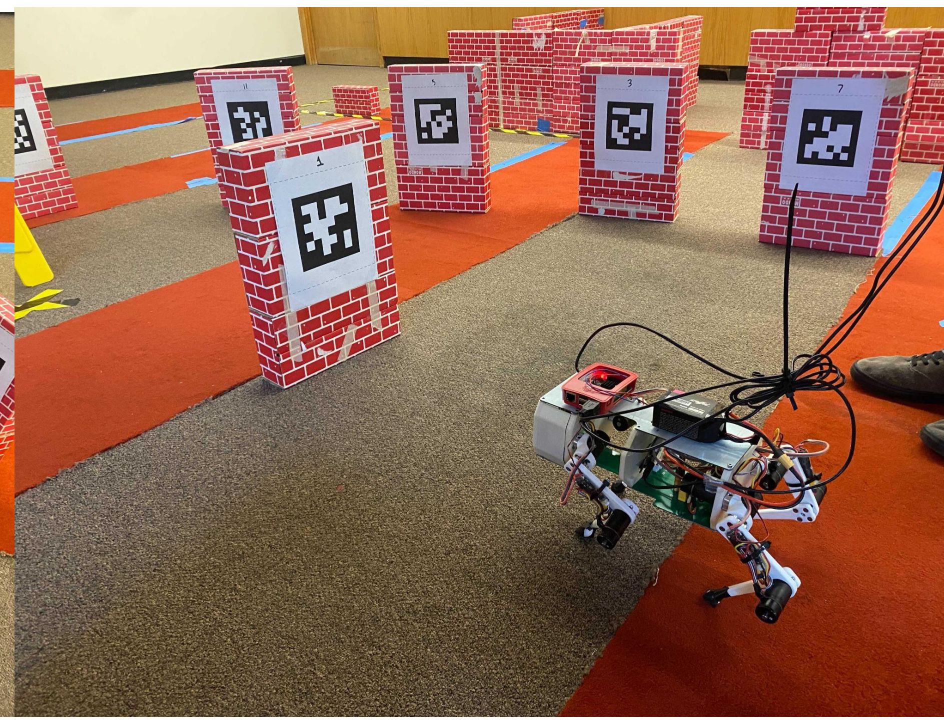
Problems with the original controller

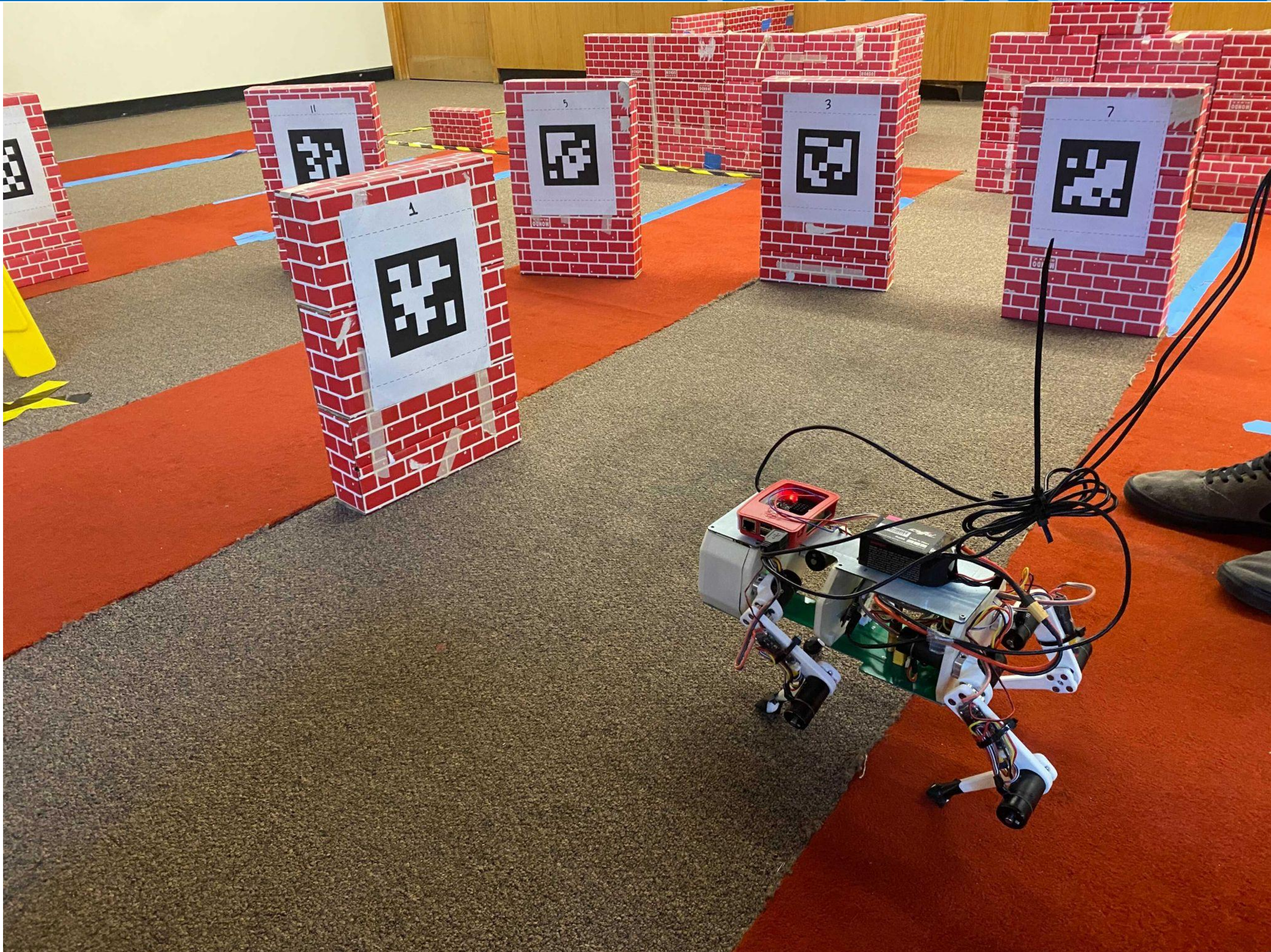
- Designed for manual control using a PS4 controller.
- No existing API.

Features of new controller

- Provides an API to emulate a PS4 controller.
- Supports both autonomous controls and manual controls.
- Does not need a display to receive input.

Testing Setup




11
5
3
7
1

Demo

Conclusion

- Calibration is a fragile process
- Graph-based motion planning is slow
- ROS modules are useful
- Localization heavily affects planning design decisions

Brandeis University

Related Work and References

W. Schwarting, J. Alonso-Mora, L. Pauli, S. Karaman and D. Rus, "Parallel autonomy in automated vehicles: Safe motion generation with minimal intervention," *2017 IEEE International Conference on Robotics and Automation (ICRA)*, Singapore, 2017, pp. 1928-1935, doi: 10.1109/ICRA.2017.7989224.

E. Behar and J. Lien, "Fast and robust 2D Minkowski sum using reduced convolution," *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, San Francisco, CA, USA, 2011, pp. 1573-1578, doi: 10.1109/IROS.2011.6094482.

J. Ward and J. Katupitiya, "Free Space Mapping and Motion Planning in Configuration Space for Mobile Manipulators," *Proceedings 2007 IEEE International Conference on Robotics and Automation*, Rome, Italy, 2007, pp. 4981-4986, doi: 10.1109/ROBOT.2007.364247.

E. Huang, M. Mukadam, Z. Liu and B. Boots, "Motion planning with graph-based trajectories and Gaussian process inference," *2017 IEEE International Conference on Robotics and Automation (ICRA)*, Singapore, 2017, pp. 5591-5598, doi: 10.1109/ICRA.2017.7989659.

https://github.com/campusrover/pupperVisualOdometry/tree/demo