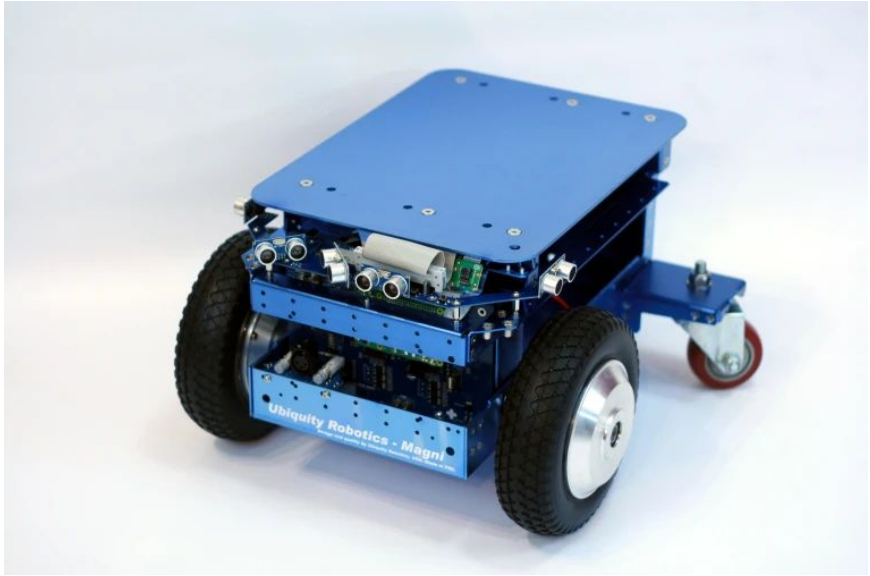
Magni - Reinforcement Learning

Abhilesh Borode, Rachel Breshears, Casey Duncan, Emerson Ham, and Mohammed Aun Siddiqui

Intro

- Magni Robot
- RL learns to correct for motor “damage”
  - RL Input: Position, Orientation
  - RL Output: Prescribed motor velocities
    - Linear in X & Angular in Z
- Goals of RL:
  - Stay as close to the line as possible
  - Increase the learning rate for convergence.

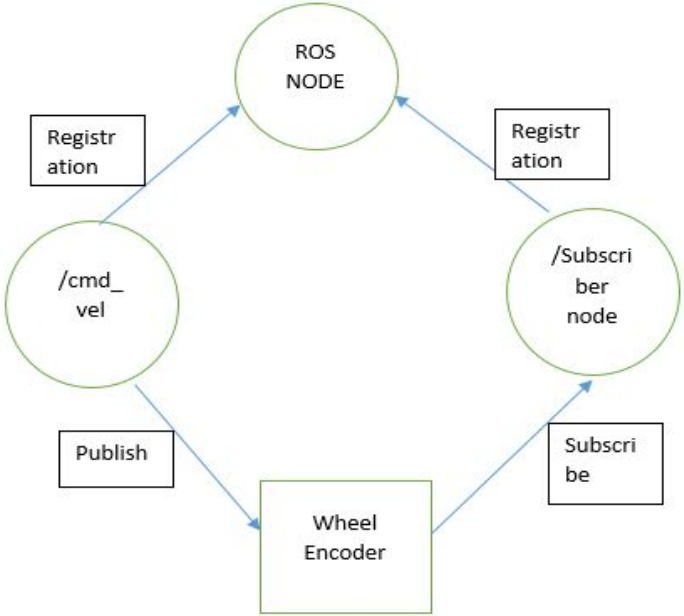


Robot Problems

- Odometry- Inconsistent
- Motor Damage- Inherent Speed variance
- Wheel Encoder- Requires resetting

Working With ROS & C++

- How ROS works
- Magni's topics
- Functions for reading/writing to topics



Basic RL Algorithm - Training

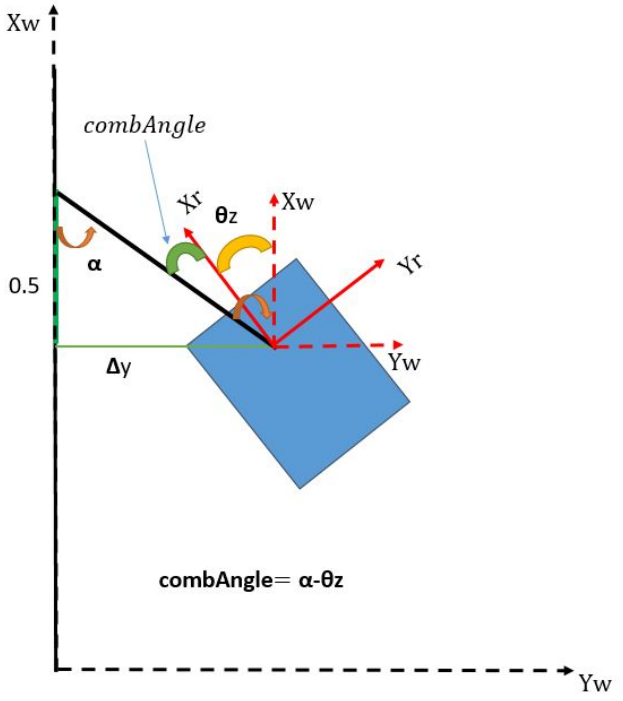
Similar to grid world homework assignment but better.

Algorithm:

1. Set the reward, learning rate, actions and states
2. Initialize Q matrix to 0 if first iteration or read the from file that saves the Q matrix.
3. Till bot reaches 4m:
   a. Calculate old state
   b. Select a random action * extreme state
   c. Calculate new state
   d. Compute reward " `reward = -abs(4.5-currentState);`"

$$Q_{k+1}(s,a) = (1-\alpha)Q_k(s,a) + \alpha \sum_{s'} T(s,a,s') \left[R(s,a,s') + \gamma \max_{a'} Q_k(s',a')\right]$$

Run 2 iterations for exploring and then 1 run for implementing from the learnt Q matrix and check performance.



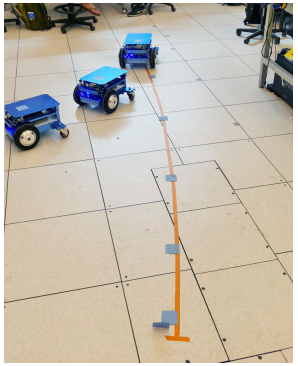
Basic RL Algorithm - Performance

Success!

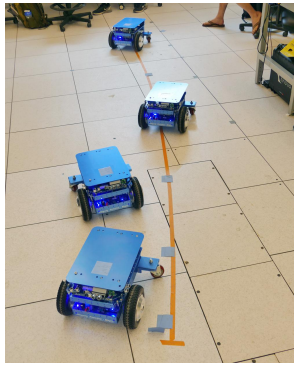
- Sometimes

Issues:

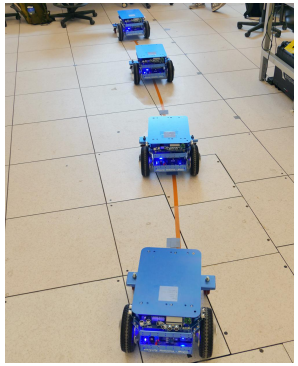
- Slow convergence
- Bad q-values can stay for a long time because of random updates.(stuck in local optima).
- Noise from trials were as large as the variation in Q-Values. Leads to noisy results, even once converged



No Training



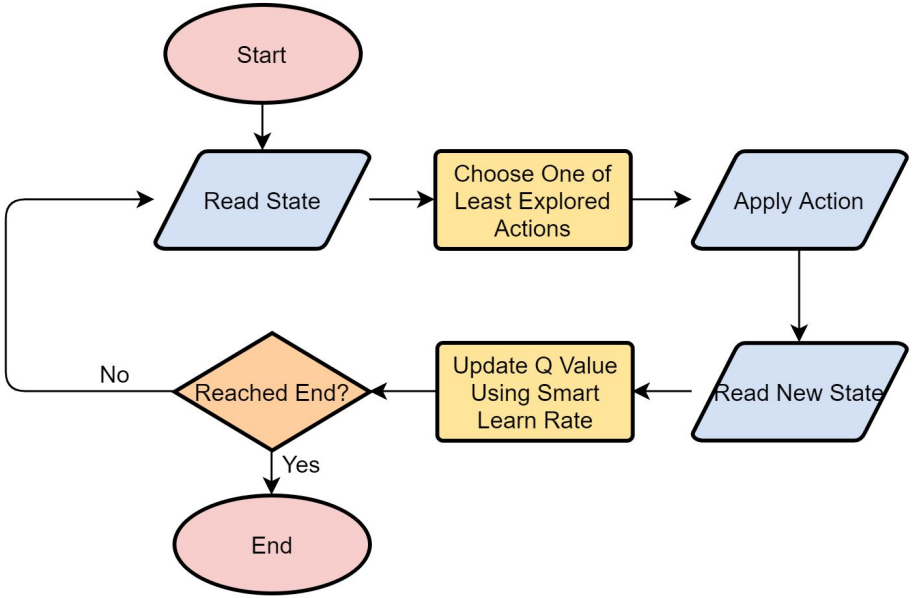
Training



Success!

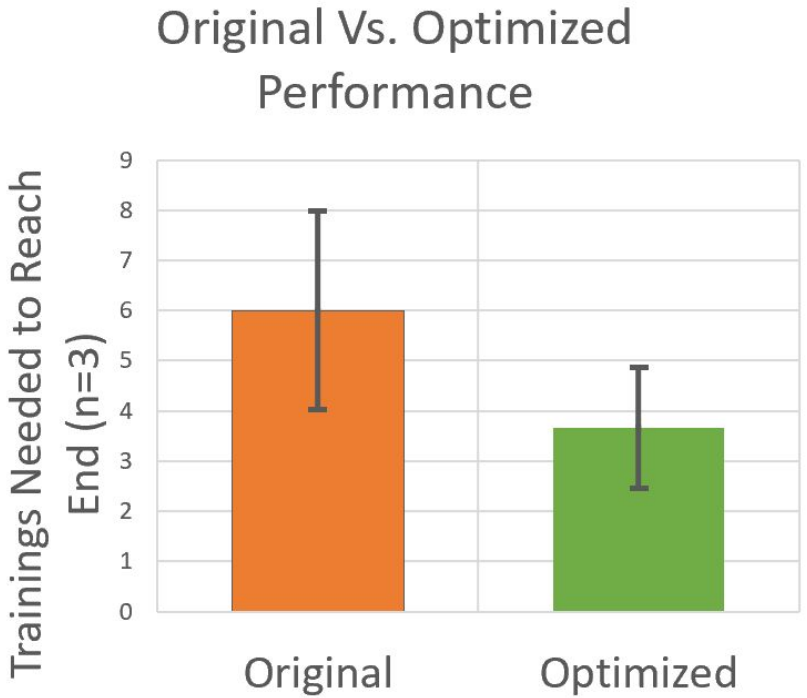
Optimizations

- Smart exploration
- Learning rate as a function of #times learned
- Adjusting state range sizes & action strengths
- Adding discounting to reduce preference for poorly learned actions



Optimizations - Performance

- 39% improvement in cumulative error
- Three trials (learned Q-Matrices) each
- Smoother movements
- Issue with “forgetting” when trained long enough for learning rate to get tiny



Conclusion

Success!

Possible further work:

- Intentionally choosing actions that arrive in states we haven't learned yet
- Collecting more data to see which optimizations were most effective