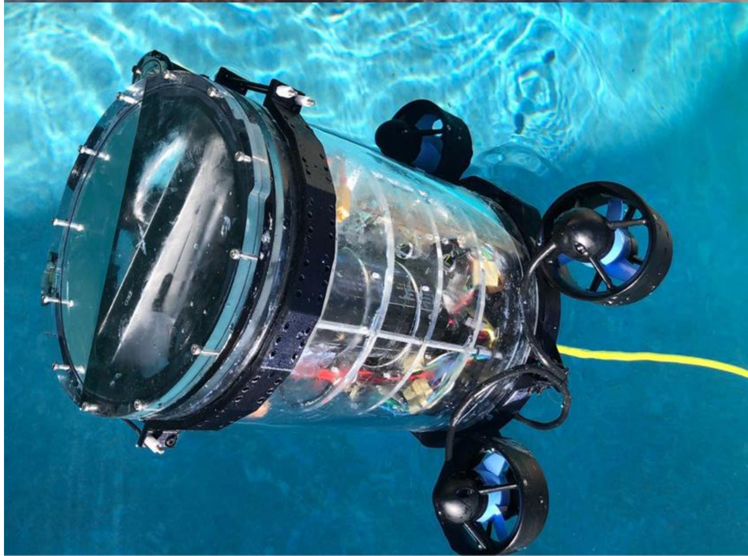
uDrone Optimality

Team 1:
Alex Goldman, Evan Lamb, Jason Newton

Introduction

- Novel, Non-holonomic, Underwater Drone
- Used for reef mapping and monitoring
- Mimics quadcopter dynamics
- “Flys” over reef at constant speed and distance, following a given path
- In development by Distributed Robotic Exploration and Mapping Systems Laboratory (**DREAMS Lab**) in partnership with The Center for Global Discovery and Conservation Science (**GDCS**)

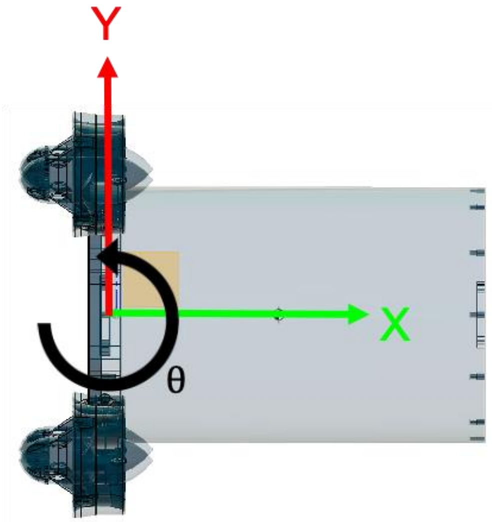


Assumptions and Simplifications

- Two dimensional world and vehicle
- Constant forward velocity (ν = 1 m/s)
- Control input of pitch rate only
- Hydrodynamic drag built into motion model
- Neutrally buoyant drone
- Assume Q = 1 and vary R
- $\dot{\theta}$ limits based on $\ddot{\theta} * \Delta t$
- Assume we use θ to aim and then move
  - (i.e. our path is a collection of step functions)
- Maintaining fixed distance over reef is modeled by a predetermined path
- Assume known world & no stochasticity

System Model

- State Space Model $X = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}$ $u = [\dot{\theta}]$ $\begin{aligned} \dot{x} &= v\cos\theta \\ \dot{y} &= v\sin\theta \\ \dot{\theta} &= u \end{aligned}$



- Cost Function $J = \int_{t_0}^{t_f} \left\| f\big(x(\tau)\big) - y(\tau) \right\|^2 + \left\| g\big(u(\tau)\big) \right\|_R^2 \, d\tau$

$f\big(x(t)\big) \triangleq$ desired y at $x(t)$

$g(u(t)) \triangleq$ control cost based on motor parameters

- Constraints

$0 \le x \le 10$

$0 \le y \le 4$

$-\frac{\pi}{2} < \theta < \frac{\pi}{2}$

$-70 < \ddot{\theta} < 70$

Discretization of System Model

- State Space Model

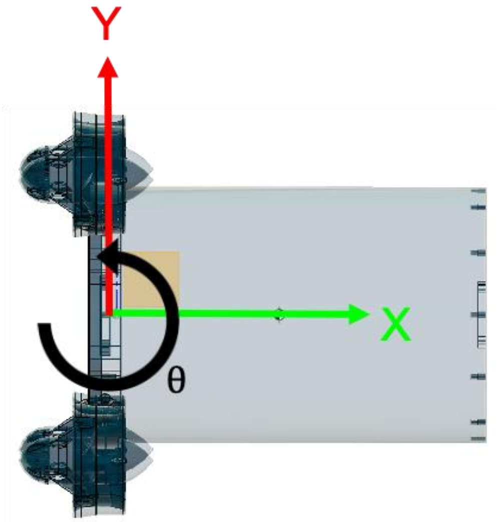
$$\begin{aligned} \theta_{k+1} &= \dot{\theta}_k * \Delta t + \theta_k \\ x_{k+1} &= v\cos\theta_{k+1} * \Delta t + x_k \\ y_{k+1} &= v\sin\theta_{k+1} * \Delta t + y_k \end{aligned}$$

- Cost Function

$$J = \sum_{\tau=t_0}^{t_f} \underbrace{\boxed{\| f(x(\tau)) - y(\tau) \|^2}}_{\text{Tracking cost}} + \underbrace{\boxed{\| g(u(\tau)) \|_R^2}}_{\text{Control cost}}$$

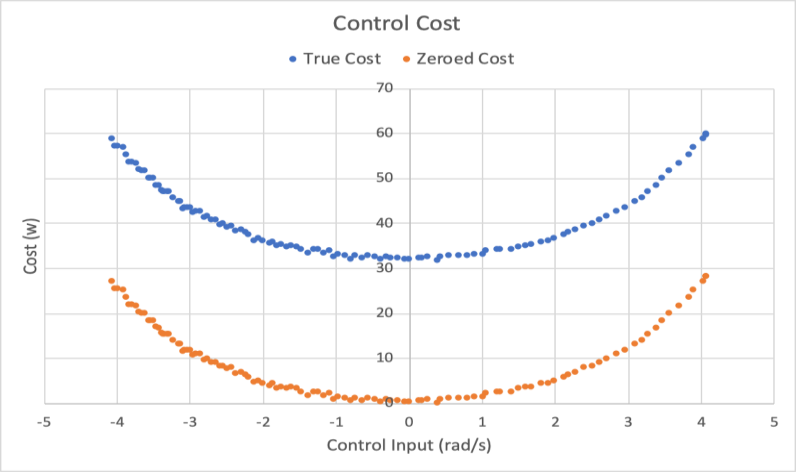
- Constraints

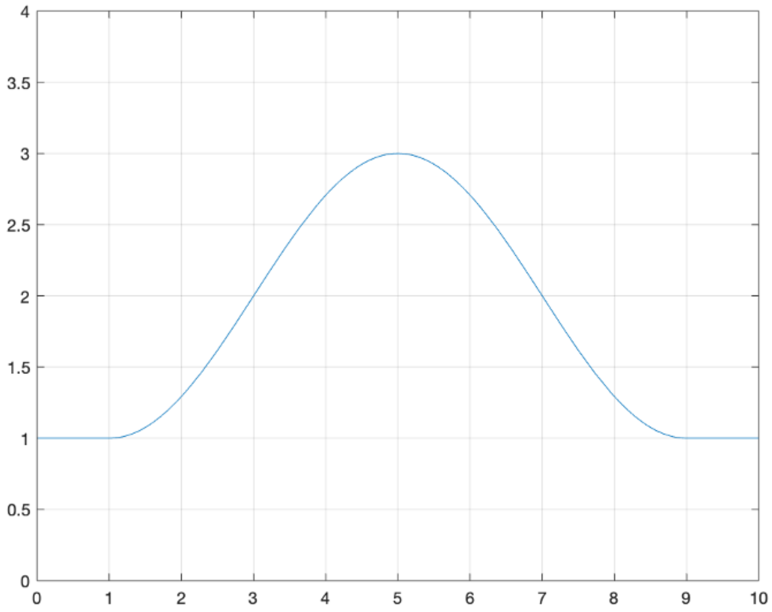
$$0 \le x \le 10$$
$$0 \le y \le 4$$
$$-\frac{11\pi}{24} < \theta < \frac{11\pi}{24}$$
$$-3.5 < \dot{\theta} < 3.5$$



Δx	0.005 m
Δy	0.01 m
$\Delta\theta$	π/24 rad (or 7.5°)
Δt	0.05 s (or 20 Hz)
Δu	0.5 rad/s

Estimation of Cost



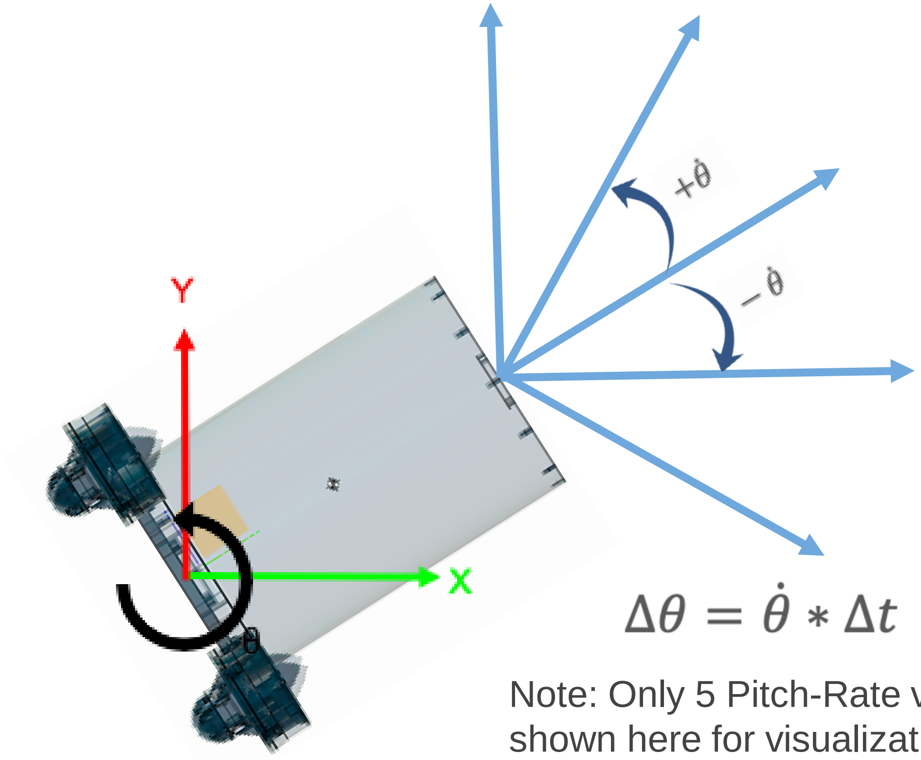


Above: Control cost as a function of control input based on motor parameters.

Right: Desired height given x. Used to calculate cost as a function of deviation from desired path.

$$y = \begin{cases} 1 & \text{if } x \in (0,1) \\ 2 - \cos\left(\frac{\pi\ (x-1)}{4}\right) & \text{if } x \in (1,9) \\ 1 & \text{if } x \in (9,10) \end{cases}$$

Conceptualization of Possible Actions

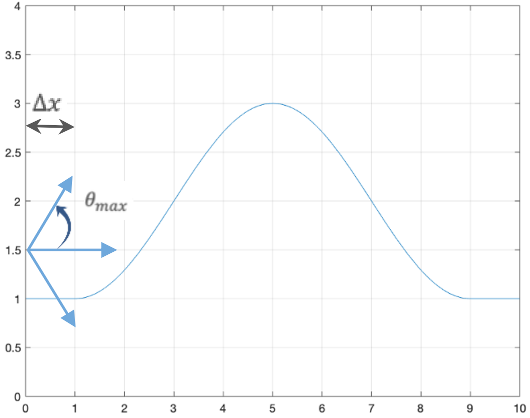


$$\Delta\theta = \dot{\theta} * \Delta t$$

Note: Only 5 Pitch-Rate values shown here for visualization, 15 values were used in simulation

In order to make sure that the uDrone travels at least one grid point in the x direction, the team determined that

$$\Delta x \leq v * \cos\theta_{max} * \Delta t$$



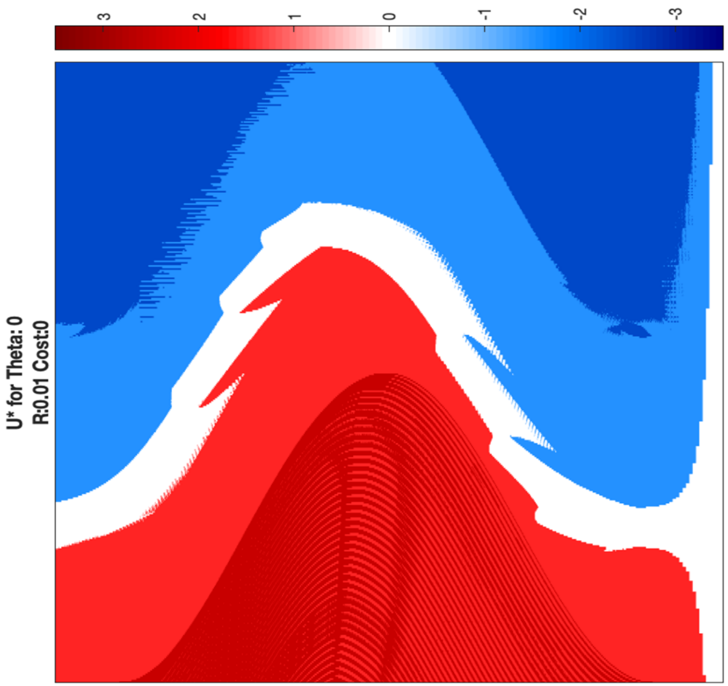
Various θ values from initial condition

Solution Method -- Dynamic Programming

- The solution began at the end.
- Order of the problem
- Steps:
    1. Initialize all variables and U*, J* matrices
    2. Initialize cost (J*) at final X state
    3. For X from End to Beginning, and each Y and θ:
        - Calculate state tracking cost
            - Implement high cost for going too far below path (i.e. hitting reef)
        - For each U
            - Find next state for each U, control input
            - Find control cost for each state
            - Find J* for each next state
                - use “nearest neighbor” approach
            - Add next J* to state and control cost
        - Find U* that minimize J over all U

- X
    - Range: 0 to 10
    - Step: 0.005
    - 2001 grid-points
- Y
    - Range: 0 to 4
    - Step: 0.001
    - 401 grid-points
- θ
    - Range: -11π/24 to 11π/24
    - Step: π/24
    - 23 grid-points
- U
    - Range: -3.5 to 3.5
    - Step: 0.5
    - 15 steps
- Total
    - States: 18,455,223
    - Actions: 276,828,345

Results - U*



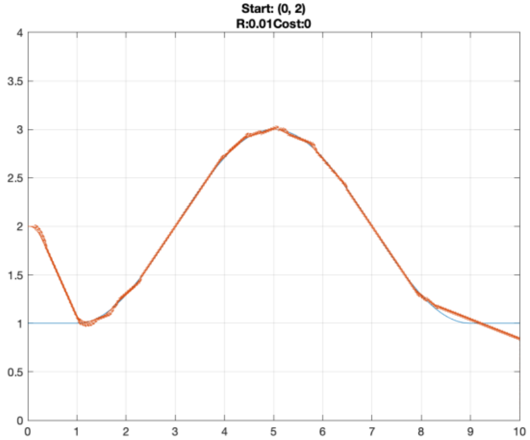
Control Law:

- Optimal Control (U*) for given X & Y positions with θ = 0.
    - Red is positive U
    - Blue is negative U

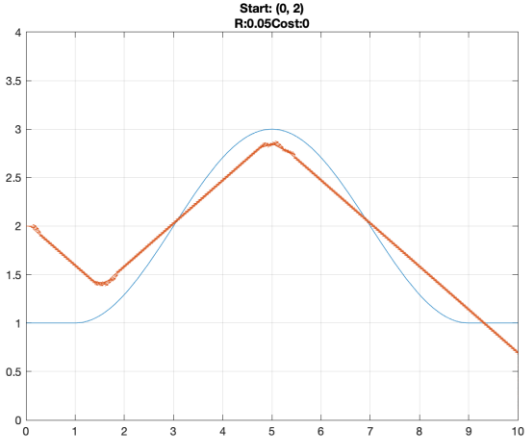
Slice

- R = 0.01
- Theta = 0
    - (1 of 23 for this R value)
- Cost Method: Zeroed

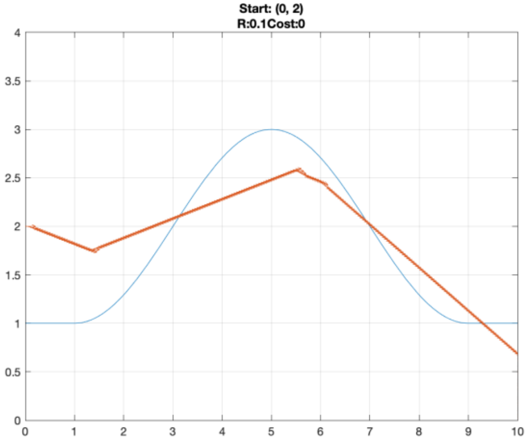
Results - Optimal Paths, Zeroed Cost



R = 0.01



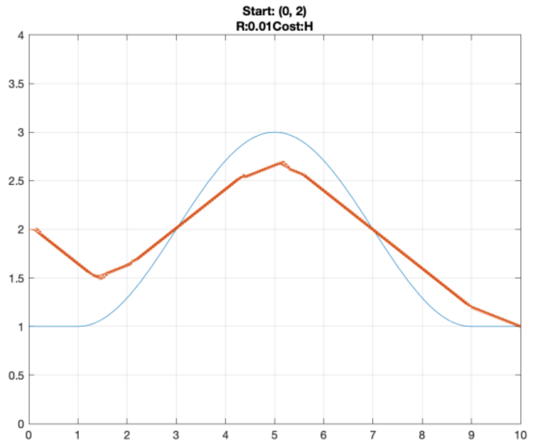
R = 0.05



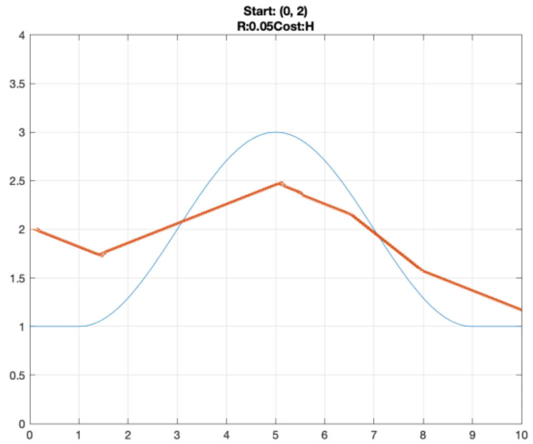
R = 0.1

All example paths show vehicle starting at y = 2, above desired path

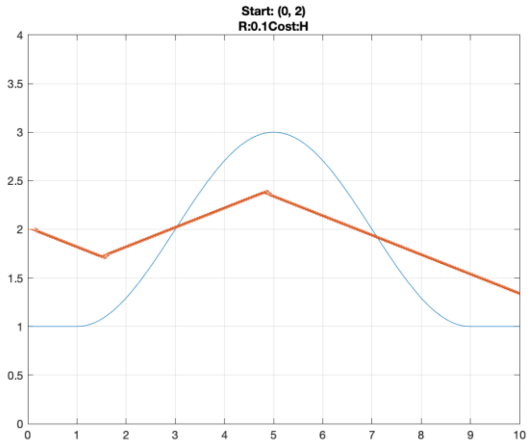
Results - Optimal Paths, True Cost



R = 0.01



R = 0.05



R = 0.1

All example paths show vehicle starting at y = 2, above desired path

Conclusions

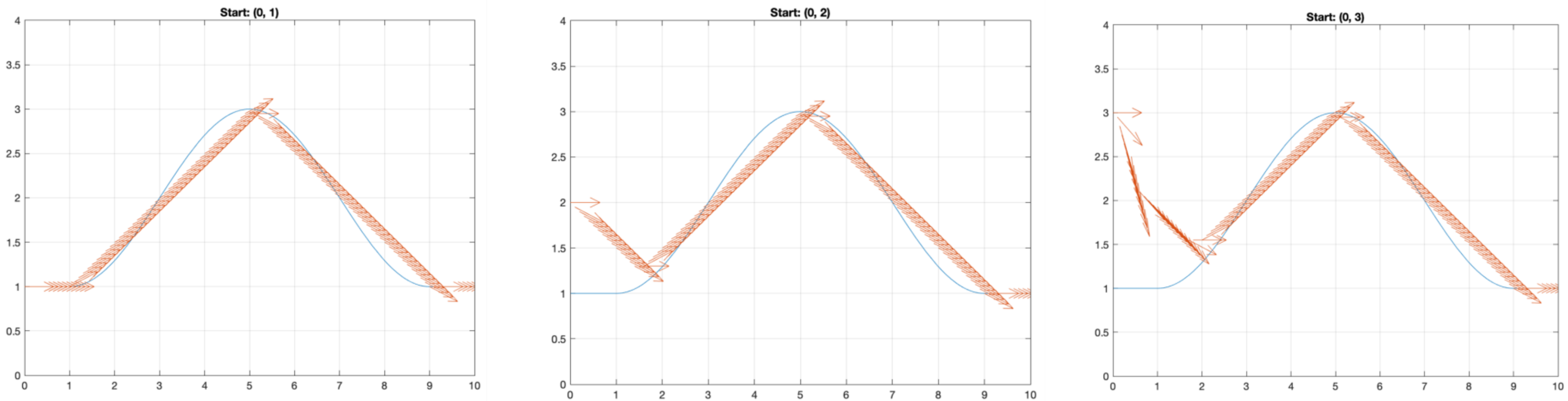
- Nearest neighbor is good for speed but decreases accuracy
- Discretization and constraints limit smooth action
- Model is simplistic but yields helpful intuition
- Able to vary R to see how results change
  - Able to weigh control cost vs. path following depending on application
- Zeroing cost permits greater path following capacity.
- For full autonomy, this would need to run onboard with a finite horizon
  - Would need to pre-compute control lookup tables for various scenarios
  - Other methods, like MPC might work better

References

[1] Fossen, T. I. (2011) *Handbook of Marine Craft Hydrodynamics and Motion Control*. West Sussex, UK: Wiley.

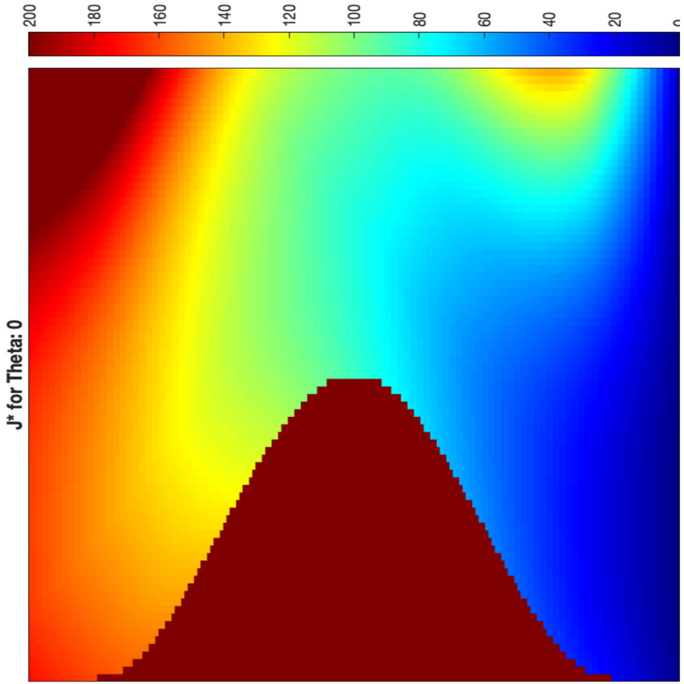
[2] Kirk, D. E. (1970). *Optimal Control Theory, An Introduction*. Englewood Cliffs, NJ: Prentice-Hall.

Appendix A - Example Paths at 10 Hz (R = 0.005)



Optimal Paths given various starting points

Appendix B - J* (R = 0.01, Theta = 0)



Path Cost (J*) for given X & Y positions with θ = 0.
Dark red is from very high cost