CSE483-Mobile Robotics

End-semester exam

Monsoon 2019

November 18th

Maximum points: 40 Duration: 180 minutes

Instructions

- This is an **open-book** exam. You are allowed to use any paper notes or textbooks that you have brought with you.
- Laptops, tablets, or smartphones are NOT allowed. You also cannot collaborate with other students.
- Your answers must be concise and to-the-point. Verbosity will NOT fetch you additional marks.
- Sufficient space has been provided for each question. Using additional sheets are discouraged, if you need them you're probably doing something wrong.
- You do NOT get credit for replicating whatever is present in the textbook or your notes. Please do not fill your answer scripts with excerpts from such sources.
- Use the last page for rough work or for any of your answers, if necessary.
- State your assumptions clearly if there is any ambiguity with the question(s).

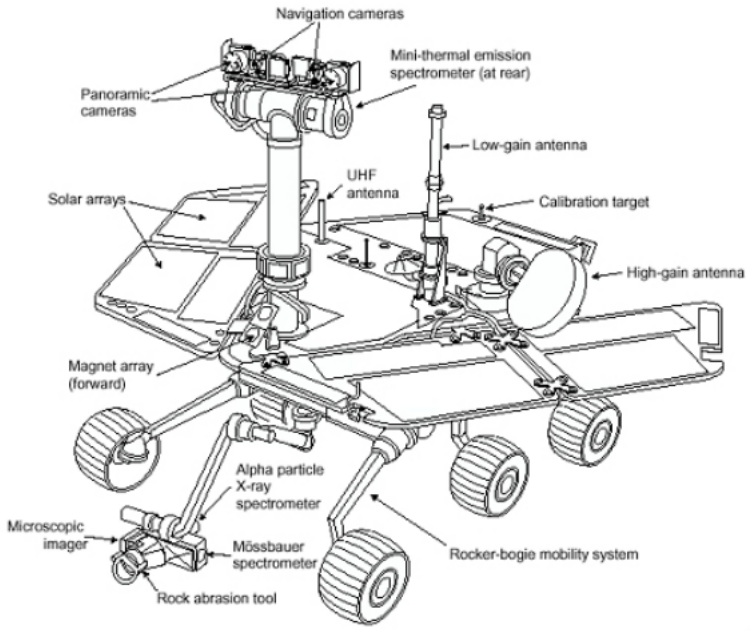
Roll number: TAs

Seat:

Invigilator sign:

Q1	Q2	Q3	Q4	Q5	Q6	Q7	Total

Q1) A mars rover, equipped with cameras and various scientific instruments, is tasked to autonomously explore the surface of mars and collect data. Ignoring all the scientific instruments, and considering only the stereo navigation camera, systematically describe in words how you'd implement the localization, mapping, and planning modules to achieve this task. A high-level description of the algorithms in sufficient. **(5 points)**



Mars Exploration Rover

An over-simplified suggestion that does not take into account the limited computational resources on the rover.

Localization: We can get an estimate of the rover's location by applying a stereo visual odometry algorithm. For every pair of stereo images, we back-project the images to their corresponding 3D point clouds by using disparity information. Successive 3D point clouds are matched and their relative [R|t] transformations can be obtained using least squares minimization. The individual relative transformations are concatenated to recover the whole trajectory of the rover relative to its starting location.

Mapping: We would need to maintain a map of the environment that the rover perceives in order to plan a trajectory, for re-localization, and for later human inspection. We could simply maintain the individual clouds that are generated for visual odometry after transforming them to the starting frame. As the rover reobserves points in the scene, we can perform bundle adjustment to improve both the point cloud and the rover's trajectory.

Planning: Since the rover traverses only on the ground, we could project the 3D map into a 2D occupancy grid map by considering all points that are above a certain height to be the occupied grids or obstacles. Assuming the goal is given, we could run a high level path planner to generate a geometrically feasible path which we then convert into a kinematically feasible trajectory by sampling points along the path and fitting a Bernstein polynomial curve such that it considers the non-holonomic constraints of the rover.

Q2) Consider two calibrated cameras C_1, C_2 that are looking at a scene. The relative orientation R, t between the two cameras, and an up-to-scale 3D point cloud of the scene, $\{X_i\}$, were estimated by applying our well known linear 8-point and triangulation algorithms. We now wish to refine these estimates by applying bundle adjustment. (a) Write down the cost function to minimize. **(1 point)** (b) Why is the cost function squared? **(1 point)** (c) How can the cost function be written in linear form? **(1 point)** (d) What are the dimensions of the Jacobian? Show the structure of the Jacobian in block form. **(2 points)**

(a)

$$\sum_{i=1}^{2} \sum_{j=1}^{n} \| x_{ij} - P_i X_j \|_2^2$$

where n is the number of points.

(b) We assume our image measurements x to have zero mean Gaussian noise.

$$Pr(x \mid X, P) = \prod_i \prod_j \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2} \frac{d(x_{ij}, P_i X_j)^2}{\sigma^2}}$$

Taking the log-likelihood,

$$\log Pr(x \mid X, P) = \sum_i \sum_j \left(\frac{-1}{2\sigma^2} \right) d(x_{ij}, P_i X_j)^2 + K$$

$\Rightarrow$ The maximum likelihood estimate of X, P leads to minimizing $\sum_i \sum_j d(x_{ij}, P_i X_j)$

(c) The non-linearity is induced by the perspective projection mapping P.

We can avoid it by using an alternative formulation,

$$\sum_{i=1}^{2}\sum_{j=1}^{n} \| x_{ij} - x'_{ij} \|_2^2$$

subject to $x'^{T}_{2j} [t]_x R x'_{ij} = 0 \quad \forall j = 1, \dots, N$

& $x'^{T}_{2j} e_3 = 1, \; x'^{T}_{1j} e_3 = 1$

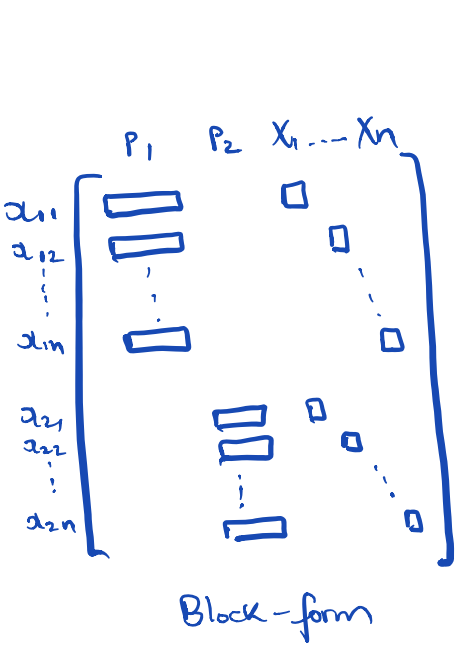
(d) Assuming we're optimizing over the 3x4 camera matrices and all the 3D points,

Dimensions of J : $2mn \times (24 + 3n)$

$r_{ij} \doteq [x_{ij} - P_i X_j]$

$\theta \doteq [P_1, P_2, X_1, X_2, \dots, X_n]_{(24+3n)}$

$$J = \frac{\partial r}{\partial \theta} = \begin{bmatrix} \frac{\partial r_{11}}{\partial P_1} & \frac{\partial r_{11}}{\partial P_2} & \frac{\partial r_{11}}{\partial X_1} & \cdots & \frac{\partial r_{11}}{\partial X_n} \\ \frac{\partial r_{12}}{\partial P_1} & \frac{\partial r_{12}}{\partial P_2} & \frac{\partial r_{12}}{\partial X_1} & \cdots & \frac{\partial r_{12}}{\partial X_n} \\ \vdots & \vdots & \vdots & & \vdots \\ \frac{\partial r_{1n}}{\partial P_1} & \frac{\partial r_{1n}}{\partial P_2} & \frac{\partial r_{1n}}{\partial X_1} & \cdots & \frac{\partial r_{1n}}{\partial X_n} \\ \frac{\partial r_{21}}{\partial P_1} & \frac{\partial r_{21}}{\partial P_2} & \frac{\partial r_{21}}{\partial X_1} & \cdots & \frac{\partial r_{21}}{\partial X_n} \\ \vdots & \vdots & \vdots & & \vdots \\ \frac{\partial r_{2n}}{\partial P_1} & \frac{\partial r_{2n}}{\partial P_2} & \frac{\partial r_{2n}}{\partial X_1} & \cdots & \frac{\partial r_{2n}}{\partial X_n} \end{bmatrix} \doteq$$



Block-form

Q3) Suppose a 2D robot exists in space and it measures its distance from a set of n known landmarks located at $\mathrm{l}_1, \mathrm{l}_2, ..., \mathrm{l}_\mathrm{n}$ using a range sensor. Each distance measurement r_i is given as $r_i = \|\mathrm{x} - \mathrm{l}_\mathrm{i}\| + \eta_i$, where x is the robot location and η_i is some unknown noise. Given these noisy measurements, describe in detail how you'd estimate the location of the robot. Derive any intermediate quantities needed. **(5 points)**

Measurement model : $r_i = \underbrace{\|x - l_i\|}_{f(x)} + \eta_i$

Given : $r_1, r_2, \ldots, r_n$ measurements

To find : 2D position x

We can formulate this as a minimization problem,

$$\arg\min_x \sum_{i=1}^{n} (r_i - f(x))^2$$

The objective is non-linear in nature, so we employ a solver like Gauss-newton where we start with an initial estimate and iteratively refine it under the assumption that the objective is locally linear.

$$J_i = \left[\frac{\partial f_i}{\partial x}\right]_{1\times 2} = \frac{\partial}{\partial x}\left[\sqrt{(x-l_i^x)^2+(y-l_i^y)^2}\right]$$

$$= \left[\frac{\partial}{\partial x}\sqrt{(x-l_i^x)^2+(y-l_i^y)^2} \quad \frac{\partial}{\partial y}\sqrt{(x-l_i^x)^2+(y-l_i^y)^2}\right]$$

$$= \begin{bmatrix} \frac{(x - l_i^x)}{\sqrt{(x-l_i^x)^2 + (y-l_i^y)^2}} \\ \frac{(y - l_i^y)}{\sqrt{(x-l_i^x)^2 + (y-l_i^y)^2}} \end{bmatrix}^T$$

Algorithm:

- initialize x
- Compute $[J]_{N\times 2}$, r where $r = \begin{bmatrix} r_1 - f(x) \\ r_2 - f(x) \\ \vdots \\ r_n - f(x) \end{bmatrix}_{N\times 1}$
- while $J^T r > \epsilon$
  - Compute approx. Hessian $H = J^T J$
  - Compute update $\Delta = (H)^{-1} J^T r$
  - Apply update $x \leftarrow x + \Delta$
  - Compute J, r
- return x

Note: This is NOT a filtering problem. It was nowhere mentioned that the robot is moving.

Q4) A sensor platform, consisting of a stereo camera and an inertial measurement unit (IMU), moves freely through a 3D scene. We identify the pose of the platform as the pose of the IMU. The rotation and translation of the camera relative to the IMU is known and is given by $\mathrm{R}_\mathrm{c}^\mathrm{I}$ and $\mathrm{t}_\mathrm{c}^\mathrm{I}$ respectively. We wish to estimate the pose of the platform with respect to some fixed world frame, using the sensor measurements as the platform moves around via an extended Kalman filter.

The scene contains known point landmarks on the floor which are observed by the stereo camera, as $\mathrm{y_k} = [u_{left}, v_{left}, u_{right}, v_{right}]^T$, for every landmark at every time step k. The 3D locations of these landmarks are known in the world frame. The angular velocity and translational velocity of the platform, ω_k, $\mathrm{v_k}$, relative to the IMU, are obtained from the IMU measurements for every time step k.

(a) What is the state vector and its dimension? **(1 point)** (b) Derive the translational and rotational motion models for the platform. (*Hint:* $\Psi_k = \omega_\mathrm{k} dt$, where Ψ_k is the angle-axis representation for the rotation.) **(2 points)** (c) Derive a suitable observation model for the stereo camera. **(2 points)** (d) What are the dimensions of the observation noise covariance? **(1 point)** (e) What are the dimensions of the Kalman gain for a single landmark? **(2 points)** (f) Derive the Jacobian for the observation model. **(2 points)**

(a) Our state is the pose of the platform in the world frame

$$x = \left(p_I^w,\ R_I^w\right)^T$$

Dimensions : 12×1 (for euler angles or axis-angle parameterization it would be 6×1)

(b) Motion model:

translational model: $p_{I,k}^w = p_{I,k-1}^w + R_{I,k-1}^w \left(v_{k-1} dt + \eta_v\right)$

$$\Psi_k = \omega_{k-1} dt + \eta_w$$

$$\Rightarrow R_{I,k}^{I,k-1} = I_{3\times 3} + \frac{\Psi_k}{\|\Psi_k\|} \sin(\|\Psi_k\|) + \frac{\Psi_k^2}{\|\Psi_k\|^2}\left(1 - \cos(\|\Psi_k\|)\right)$$

[Rodrigues' formula]

rotational model: $\Rightarrow R_{I,k}^w = R_{I,k-1}^w \, R_{I,k}^{I,k-1}$

(c) Observation of j^{th} landmark:

$$y_{j,k} = \left(u_{j,l}, v_{j,l}, u_{j,r}, v_{j,r}\right)_k^T$$

Observation model:

Consider the known landmark in world coordinates, X_j^w

Its coordinates in the IMU frame at $t=k$

$$X_{j,k}^I = (R_{I,k}^w)^T \left(X_j^w - p_{I,k}^w \right)$$

Its coordinates in the camera frame at $t=k$

$$X_{j,k}^c = R_I^c \left[(R_{I,k}^w)^T \left(X_j^w - p_{I,k}^w \right) \right] + t_I^c$$

Its image coordinates (observation),

$$\begin{pmatrix} u_{j,left} \\ v_{j,right} \end{pmatrix} = \pi \left\{ R_I^c \left[(R_{I,k}^w)^T \left(X_j^w - p_{I,k}^w \right) \right] + t_I^c \right\}$$

The total observation for the stereo camera,

$$y_j = \begin{pmatrix} u_{j,left} \\ v_{j,left} \\ u_{j,right} \\ v_{j,right} \end{pmatrix} = \begin{pmatrix} \pi\, X_{j,k}^c \\ \\ \pi \left(X_{j,k}^c - b \right) \end{pmatrix} + \eta$$

where $\pi : \mathbb{R}^3 \to \mathbb{R}^2$

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} \mapsto \begin{pmatrix} f_x x \\ f_y y \\ 1 \end{pmatrix} \frac{1}{z} + \begin{pmatrix} c_x \\ c_y \\ 0 \end{pmatrix}$$

(d) Dimensions of observation: 4×1

$\Rightarrow$ Dimensions of observation covariance Q_k: 4×4

(e) $$K = \check{P}_k G_k^T \left(G_k \check{P}_k G_k^T + Q_k \right)^{-1}$$

Dimensions of $\check{P}_k$: 12×12

Dimensions of G_k: 4×12

$\Rightarrow$ Dimensions of K: 12×4

(f) Observation model Jacobian, G:

$$G_k = \frac{\partial y_j}{\partial x_k}$$

$\Rightarrow$ $$G = \begin{pmatrix} \frac{\partial y}{\partial p_{I,k}^w} & \frac{\partial y}{\partial R_{I,k}^w} \end{pmatrix}_{4 \times 12}$$

$$\frac{\partial y}{\partial P_k} = \frac{\partial y}{\partial X_{j,k}^c} \times \frac{\partial X_{j,k}^c}{\partial p_{I,k}^w}$$

$$\frac{\partial y}{\partial R_k} = \frac{\partial y}{\partial X_{jm}^c} \times \frac{\partial X_{j,k}^c}{\partial R_{I,k}^w}$$

$$\frac{\partial y}{\partial X^c_{j,k}} = \frac{\partial}{\partial X^c_{j,k}} \begin{pmatrix} f_x X/z + c_x \\ f_y Y/z + c_y \\ f_x (X-b)/z + c_x \\ f_y Y/z + c_y \end{pmatrix} = \begin{pmatrix} f_x/z & 0 & -f_x X/z^2 \\ 0 & f_y/z & -f_y Y/z^2 \\ f_x/z & 0 & -f_x (X-b)/z^2 \\ 0 & f_y/z & -f_y Y/z^2 \end{pmatrix}_{4\times 3}$$

$$\frac{\partial X^c_{j,k}}{\partial p^w_{I,k}} = \frac{\partial}{\partial p^w_{I,k}} \left(R^c_I \left[(R^w_{I,k})^T (X^w_j - p^w_{I,k}) \right] + t^c_I \right)$$

$$= \frac{\partial}{\partial p^w_{I,k}} \left(-R^c_I (R^w_{I,k})^T p_{I,k} \right) \quad \text{(ignoring other terms)}$$

$$= -R^c_I (R^w_{I,k})^T$$

(optional):

$$\frac{\partial X^c_{j,k}}{\partial R^w_{I,k}} = \frac{\partial}{\partial R^w_{I,k}} \left(R^c_I \left[(R^w_{I,k})^T (X^w_j - p^w_{I,k}) \right] + t^c_I \right)$$

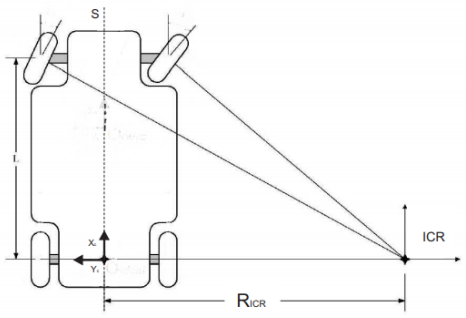
$$= \frac{\partial}{\partial R^w_{I,k}} \left(R^c_I (R^w_{I,k})^T X^w_j \right)$$

$$= \begin{pmatrix} r_{11}x & r_{12}x & r_{13}x & r_{11}y & r_{12}y & r_{13}y & r_{11}z & r_{12}z & r_{13}z \\ r_{21}x & r_{22}x & r_{23}x & r_{21}y & r_{22}y & r_{23}y & r_{21}z & r_{22}z & r_{23}z \\ r_{31}x & r_{32}x & r_{33}x & r_{31}y & r_{32}y & r_{33}y & r_{31}z & r_{32}z & r_{33}z \end{pmatrix}$$

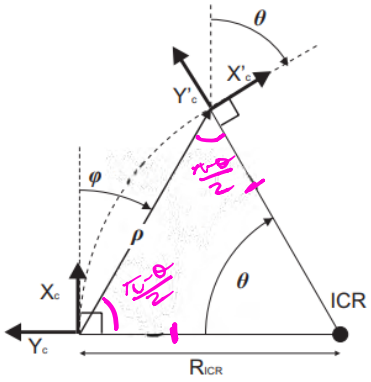
where $r_{ij} \doteq [R^c_I]_{ij}$

$$X^w_j \doteq \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

Q5) Consider a non-holonomic car that is equipped with a camera. The camera axes are denoted by X_c and Y_c, and is placed such that X_c is perpendicular to the rear-wheel axis as shown in figure (a). Due to the motion constraints of the car, the camera's motion can be perceived to be a circular motion, as shown in figure (b). X_c' and Y_c' are the camera axes in motion, whose origin with respect to the first frame, in polar coordinates, is at ρ, ϕ and is rotated by θ.



(a)



(b)

(a) Compute the essential matrix E for this configuration. **(3 points)** (b) How many parameters does it have? How many corresponding image points are required for estimating this E? It is not eight. **(2 points)**

(a) We know $E = [t]_x R$

Here, $t = \begin{pmatrix} f\cos\phi \\ -f\sin\phi \\ 0 \end{pmatrix}$ [Converting polar coordinates to Cartesian]

$$[t]_x = \begin{pmatrix} 0 & 0 & -f\sin\phi \\ 0 & 0 & -f\cos\phi \\ f\sin\theta & f\cos\phi & 0 \end{pmatrix}$$

$$R = \begin{pmatrix} \cos(-\theta) & -\sin(-\theta) & 0 \\ \sin(-\theta) & \cos(-\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$ [rotation of $-\theta$ about z-axis]

$$= \begin{pmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$\Rightarrow \quad E = [t]_\times R$$

$$= \begin{pmatrix} 0 & 0 & -f\sin\phi \\ 0 & 0 & -f\cos\phi \\ f\sin\phi & f\cos\phi & 0 \end{pmatrix} \begin{pmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 0 & -f\sin\phi \\ 0 & 0 & -f\cos\phi \\ f\sin\phi\cos\theta - f\cos\phi\sin\theta & f\sin\phi\sin\theta + f\cos\phi\cos\theta & 0 \end{pmatrix}$$

$$= f \begin{pmatrix} 0 & 0 & -\sin\phi \\ 0 & 0 & -\cos\phi \\ \sin(\phi-\theta) & \cos(\phi-\theta) & 0 \end{pmatrix}$$

(b) E consists of <u>three</u> parameters : f, ϕ, θ

We know one pair of correspondences (x, x') gives us one equation.

Since the scale factor is unknown, f can be arbitrarily set to 1 and we can solve for θ, ϕ alone for which two points are sufficient.

optional : From the figure we see that

$$\phi + \frac{\pi - \theta}{2} = \pi/2$$

$$\Rightarrow \quad \phi = \theta/2$$

Hence it is sufficient to estimate ϕ alone for which only <u>one</u> point is required.

[More details : Scaramuzza et al. 2009]

Q6) (a) Consider a holonomic disc-shaped robot with diameter, say d. The robot is initially at start location and is required to reach the goal location while avoiding obstacles. In order to avoid collision, the robot must maintain at least l distance (where $l = d/2$) from its center to the obstacle surface. It considers to have reached the goal if its center coincides with the goal location. Write an RRT (unidirectional, and not goal biased) algorithm for the robot to find a path from start to goal. **(3 points)** (b) State the likely problems faced by basic RRT and goal biased RRT in the given environment. Also suggest improvements to be done to the algorithm to resolve these problems. **(2 points)**

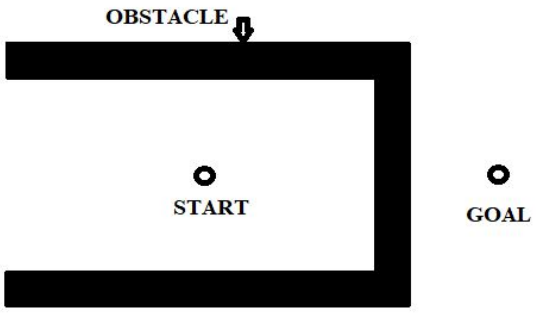


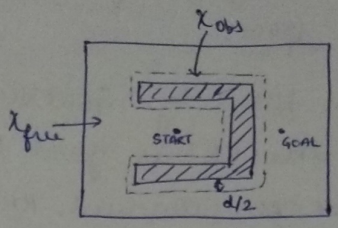
Figure 2: Robot environment

P.T.O

Q6

a) Basic RRT (x_s, x_G, x_{free}, x_{obs}):

1) Create a tree 'T' with a node 'x_s'.
2) For i = 0 to maxNoOfIters do
3) $x_R \leftarrow$ RANDOM_STATE (x_{free})
4) $x_N \leftarrow$ NEAREST_NEIGHBOR(x_R, T)
5) x_{New}, $u_{New} \leftarrow$ NEXT_STATE (x_R, x_N)
6) If COLLISION_FREE (x_N, x_{New}) then
7) T $\leftarrow$ T $\cup$ {(x_{New}, u_{New})}
8) If $x_{New} == x_G$ then
9) return ("REACHED", T)
10) End if.
11) End if
12) End for
13) return ("ITERATIONS OVER", T)



RANDOM_STATE(x_{free}): Randomly samples a point from x_{free} space.

NEAREST_NEIGHBOR(x_R, T): Fends nearest neighboring node for x_R in the tree T

NEXT_STATE(x_R, x_N): Generates a state x_{New} and controls u_{New} such that x_{New} lies Δ distance away from x_N along the direction $x_N \rightarrow x_R$. u_{New} are the controls required to steer from x_N to x_{New}.

COLLISION_FREE(x_N, x_{New}): Tests if line segment connecting x_N and x_{New} is contained in x_{free}.

b) Problem faced by basic RRT in the given environment –
Slow convergence of the algorithm due to random sampling of the space.

Problem faced by Goal biased RRT in the given environment –
Since the RRT is goal biased, therefore it will get trapped in the environment. RRT will sample points near GOAL which will result in paths going from START towards the obstacle boundary.

Solution to both problems – Bidirectional RRT

1) Since at a time, two trees from START and GOAL respectively are getting built, therefore the convergence time reduces.

2) Even if T_{START} gets trapped, T_{GOAL} will be building towards GOAL. In the bidirectional RRT algo, T_{START} and T_{GOAL} are swapped in order to keep both the trees growing equally.

Q7) Consider a drone that starts from a location (x_0, y_0, z_0) which strives to reach (x_g, y_g, z_g). (a) Formulate a quadratic goal reaching cost function that computes a set of controls to reach the goal. **(1 points)** (b) What are the constraints you may want to add to the above cost function. Do not worry about the obstacle avoidance constraint. Write down all other constraints that you feel are essential. Explain these constraints. **(1 points)** (c) Consider an obstacle of radius R, centered at (x_{ob}, y_{ob}, z_{ob}). Formulate an obstacle avoidance constraint that enables optimizing over a sequence of controls to eventually reach the goal. **(1 points)** (d) Linearize the obstacle avoidance constraint. **(2 points)**

Q7

a) Let robot position is given by (x_n, y_n, z_n) at a time t and goal position is (x_G, y_G, z_G)

$\therefore$ Goal reaching criteria would be satisfied when

$(x_n, y_n, z_n) = (x_G, y_G, z_G)$

$\Rightarrow$ We want to

$$\text{minimize } ((x_n - x_G)^2 + (y_n - y_G)^2 + (z_n - z_G)^2) \quad \text{— ①}$$

$\because$ Drone is holonomic robot.

$\therefore$ we can say,

$$\left.\begin{aligned} x_n &= x_0 + \sum_{i=0}^{n-1} \dot{x}_i \, \delta t \\ y_n &= y_0 + \sum_{i=0}^{n-1} \dot{y}_i \, \delta t \\ z_n &= z_0 + \sum_{i=0}^{n-1} \dot{z}_i \, \delta t \end{aligned}\right\} \text{— ②}$$

Eq ② can be substituted in ① and further simplified to,

$$\min \left((x_0 - x_G) + \sum_i \dot{x}_i \, \delta t\right)^2 + \left((y_0 - y_G) + \sum_i \dot{y}_i \, \delta t\right)^2 + \left((z_0 - z_G) + \sum_i \dot{z}_i \, \delta t\right)^2$$

~~On expanding and constructing the quadratic form, we get~~ which

Here (x_0, y_0, z_0) and (x_G, y_G, z_G) are constants (start and Goal position respectively).

$\Rightarrow$ The objective is to find set of $(\dot{x}_i, \dot{y}_i, \dot{z}_i)$ i.e., velocity components along X, Y and Z respectively such that $(x_n, y_n, z_n) \longrightarrow (x_G, y_G, z_G)$.

b) Various constraints that can be added to goal reaching cost function-

1) Velocity constraint at some way points.

2) Range of drone movement – height constraint, no fly areas etc.

c) Let obstacle position $= (x_{ob}, y_{ob}, z_{ob})$

So the obstacle could be avoided if the following condition is true,

$$(x_n - x_{ob})^2 + (y_n - y_{ob})^2 + (z_n - z_{ob})^2 > R^2 \quad - (3)$$

Substitute (2) in (3),

$$(x_0 - x_{ob}) + \sum_{i=0}^{n-1} \dot{x}_i \delta t)^2 + (y_0 - y_{ob}) + \sum_{i=0}^{n-1} \dot{y}_i (\delta t)^2 + (z_0 - z_{ob}) + \sum_{i=0}^{n-1} \dot{z}_i \delta t)^2 > R^2 \quad - (4)$$

d) To linearize the obstacle avoidance constraint, we can use first order Taylor series expansion.

Multivariate

First order Taylor series expansion is given as follows,

$$f(\dot{x}_0 + \dots + \dot{x}_{n-1}) = f(\dot{x}_0^* + \dots + \dot{x}_{n-1}^*) + \left.\frac{\delta f}{\delta \dot{x}_0}\right|_{(\dot{x}_0^*, \dot{x}_1^*, \dots \dot{x}_n^*)} (\dot{x}_0 - \dot{x}_0^*) + \dots + \left.\frac{\delta f}{\delta \dot{x}_{n-1}}\right|_{(\dot{x}_0^*, \dots, \dot{x}_{n-1}^*)} (\dot{x}_{n-1} - \dot{x}_{n-1}^*)$$

$$f(\dot{x}_0 + \dots + \dot{x}_{n-1}) = f(\dot{x}_0^* + \dots + \dot{x}_{n-1}^*) + \underset{(\dot{x}_0^*, \dots \dot{x}_{n-1}^*)}{\left[\frac{\delta f}{\delta \dot{x}_0} \dots \frac{\delta f}{\delta \dot{x}_{n-1}}\right]} \begin{bmatrix} \dot{x}_0 - \dot{x}_0^* \\ \vdots \\ \dot{x}_{n-1} - \dot{x}_{n-1}^* \end{bmatrix} \quad - (5)$$

Now, $f(\dot{x}_0^* + \dots + \dot{x}_{n-1}^*) = \left((x_0 - x_{ob}) + \sum_{i=0}^{n-1} \dot{x}_i^* \delta t\right)^2$ [from (4)]

$$= (x_0 - x_{ob})^2 + 2(x_0 - x_{ob})\left(\sum_{i=0}^{n-1} \dot{x}_i^* \delta t\right) + \left(\sum_{i=0}^{n-1} \dot{x}_i^* \delta t\right)^2$$

$$= c_1 + 2(x_0 - x_{ob})\Delta t [1, \dots, 1] \begin{bmatrix} \dot{x}_0^* \\ \vdots \\ \dot{x}_{n-1}^* \end{bmatrix} + [\dot{x}_0^*, \dots \dot{x}_{n-1}^*] \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} [1 \dots 1] \Delta t^2 \begin{bmatrix} \dot{x}_0^* \\ \vdots \\ \dot{x}_{n-1}^* \end{bmatrix}$$

$$\Rightarrow f(\dot{x}_0^* + \dots + \dot{x}_{n-1}^*) = c_1 + q_1^T X^* + X^{*T} A X^* \quad - (6)$$

where

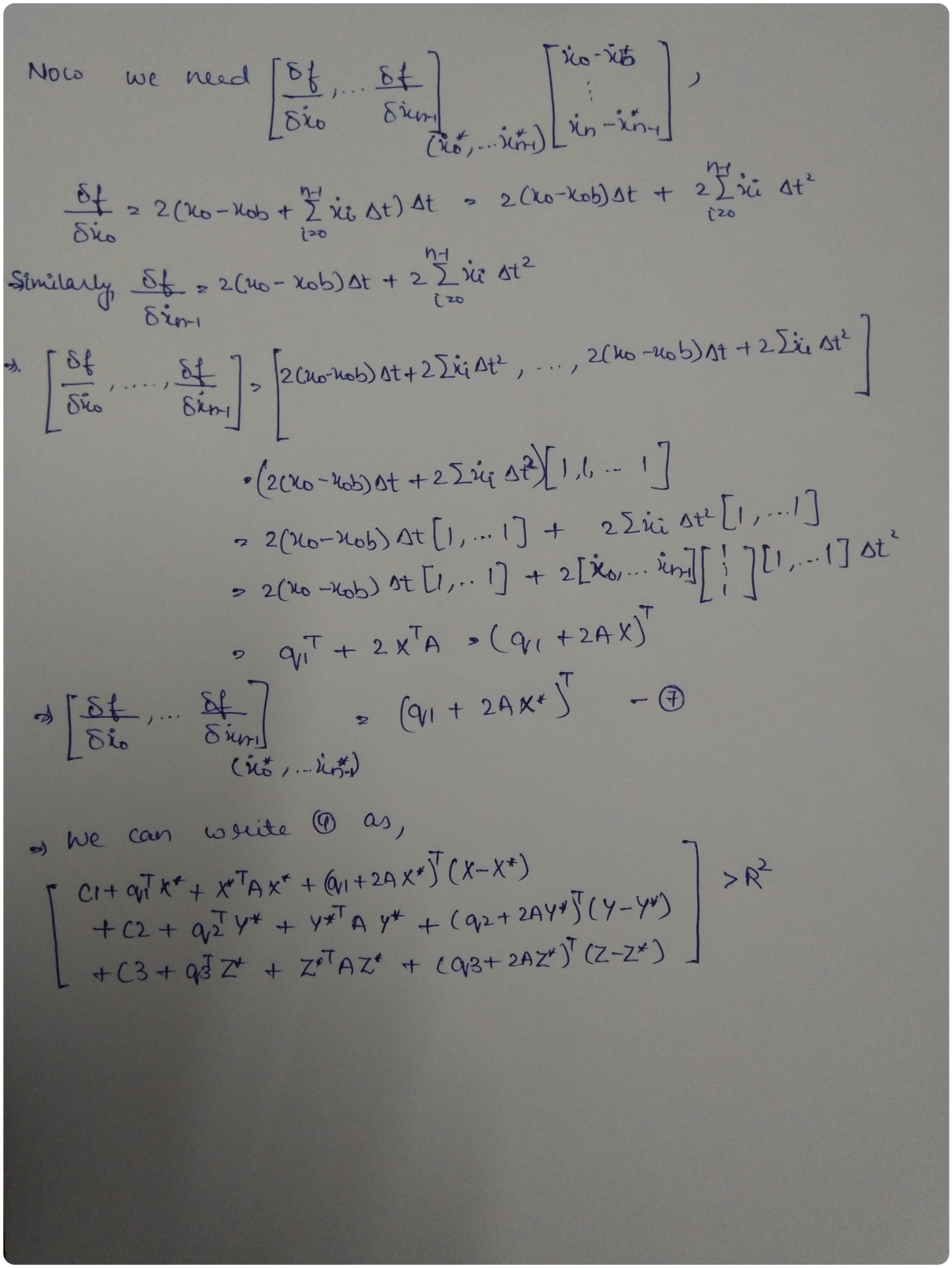
$$c_1 = (x_0 - x_{ob})^2$$

$$q_1 = 2(x_0 - x_{ob})\Delta t \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}$$

$$A = \begin{bmatrix} 1 & \dots & 1 \\ \vdots & & \vdots \\ 1 & \dots & 1 \end{bmatrix} \Delta t^2$$

$$X^* = \begin{bmatrix} \dot{x}_0^* \\ \vdots \\ \dot{x}_{n-1}^* \end{bmatrix}$$

Extra space


Now we need $\left[\frac{\delta f}{\delta \dot{x}_0}, \ldots \frac{\delta f}{\delta \dot{x}_{n-1}}\right]_{(\dot{x}_0^*, \ldots \dot{x}_{n-1}^*)} \begin{bmatrix} \dot{x}_0 - \dot{x}_0^* \\ \vdots \\ \dot{x}_n - \dot{x}_{n-1}^* \end{bmatrix}$,
$\frac{\delta f}{\delta \dot{x}_0} = 2(x_0 - x_{ob} + \sum_{i=0}^{n-1} \dot{x}_i \Delta t)\Delta t = 2(x_0 - x_{ob})\Delta t + 2\sum_{i=0}^{n-1} \dot{x}_i \Delta t^2$
Similarly, $\frac{\delta f}{\delta \dot{x}_{n-1}} = 2(x_0 - x_{ob})\Delta t + 2\sum_{i=0}^{n-1} \dot{x}_i \Delta t^2$
$\Rightarrow \left[\frac{\delta f}{\delta \dot{x}_0}, \ldots, \frac{\delta f}{\delta \dot{x}_{n-1}}\right] = \left[2(x_0 - x_{ob})\Delta t + 2\sum \dot{x}_i \Delta t^2, \ldots, 2(x_0 - x_{ob})\Delta t + 2\sum \dot{x}_i \Delta t^2\right]$
$= (2(x_0 - x_{ob})\Delta t + 2\sum \dot{x}_i \Delta t^2)[1, 1, \ldots 1]$
$= 2(x_0 - x_{ob})\Delta t [1, \ldots 1] + 2\sum \dot{x}_i \Delta t^2 [1, \ldots 1]$
$= 2(x_0 - x_{ob})\Delta t [1, \ldots 1] + 2[\dot{x}_0, \ldots \dot{x}_{n-1}] \begin{bmatrix} ! \\ ! \end{bmatrix} [1, \ldots 1] \Delta t^2$
$= q_1^T + 2X^T A = (q_1 + 2AX)^T$
$\Rightarrow \left[\frac{\delta f}{\delta \dot{x}_0}, \ldots \frac{\delta f}{\delta \dot{x}_{n-1}}\right]_{(\dot{x}_0^*, \ldots \dot{x}_{n-1}^*)} = (q_1 + 2AX^*)^T \quad - (7)$
$\Rightarrow$ we can write (7) as,
$\begin{bmatrix} C_1 + q_1^T X^* + X^{*T} A X^* + (q_1 + 2AX^*)^T (X - X^*) \\ + C_2 + q_2^T Y^* + Y^{*T} A Y^* + (q_2 + 2AY^*)^T (Y - Y^*) \\ + C_3 + q_3^T Z^* + Z^{*T} A Z^* + (q_3 + 2AZ^*)^T (Z - Z^*) \end{bmatrix} > R^2$

Extra space

Extra space