Nicholas Charron
20440122

ME 640 - Autonomous Mobile Robotics

Final Exam

April 10, 2018

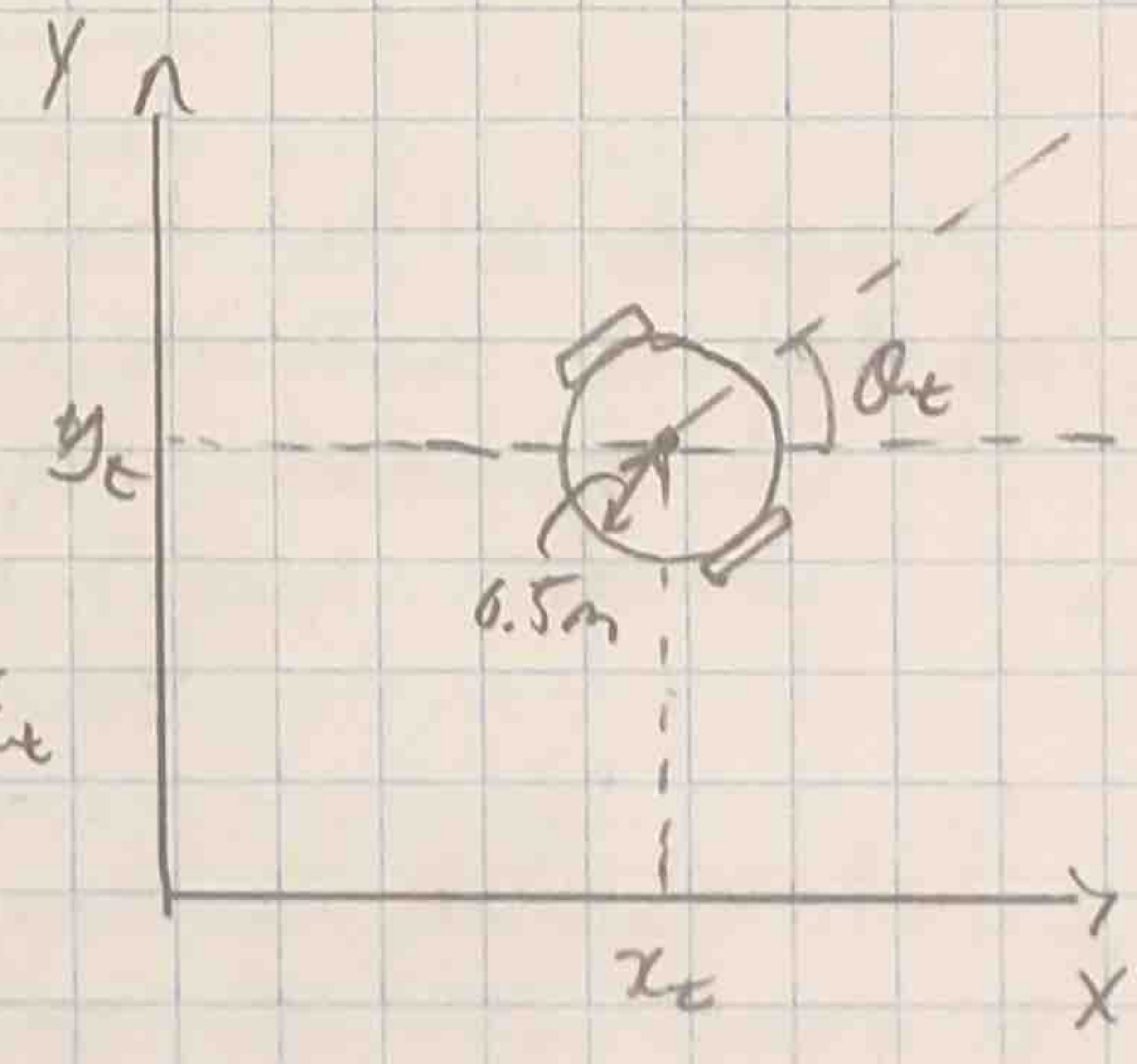
Question 1

a) States

$$X_t = \begin{bmatrix} x_t \\ y_t \\ \theta_t \\ \vdots \\ m_x^i \\ m_y^i \\ \vdots \end{bmatrix} = \quad X_{1,t}$$

Inputs

$$U_t = \begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} U_{1,t} \\ U_{2,t} \end{bmatrix}$$



Motion Model

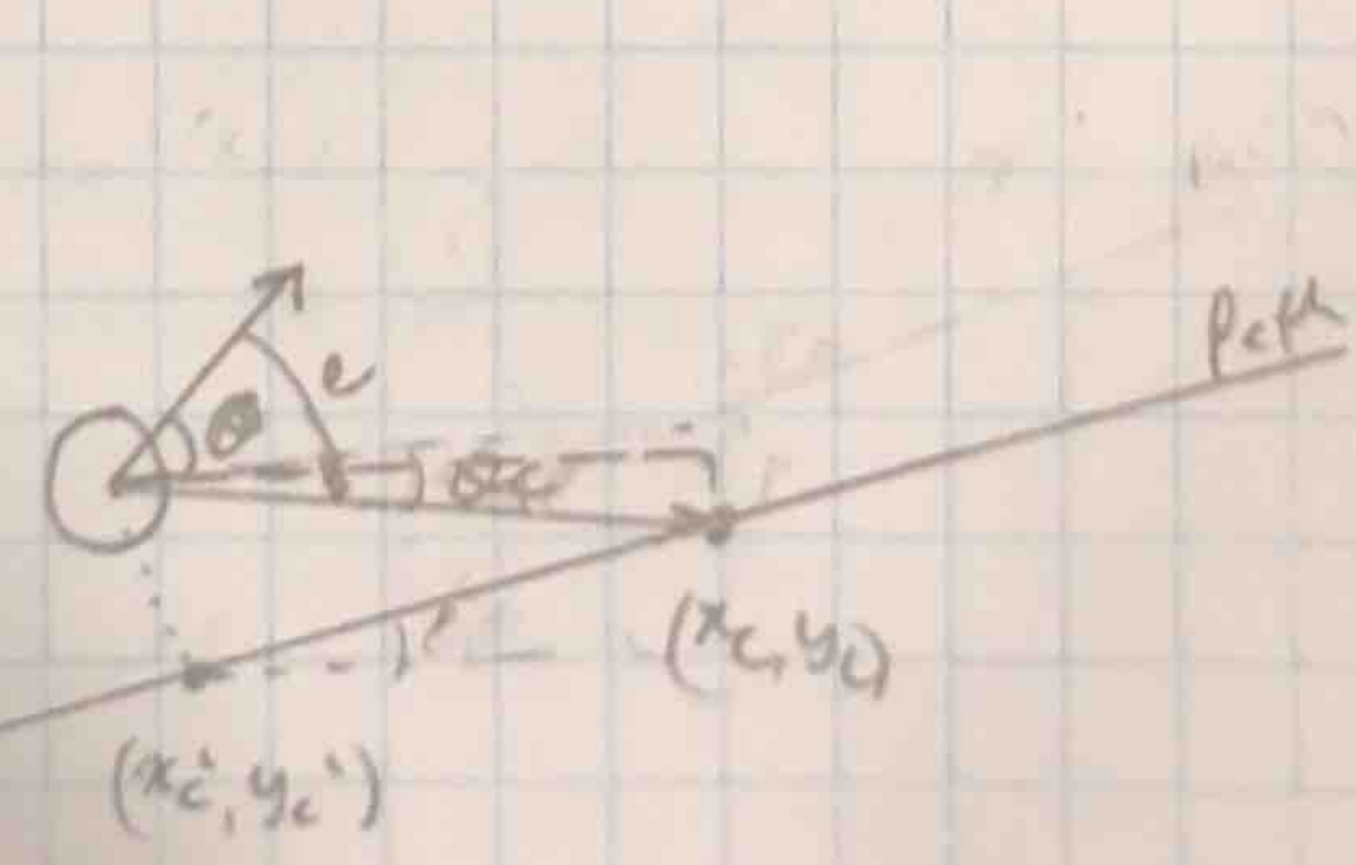
$$\begin{bmatrix} X_{1,t} \\ X_{2,t} \\ X_{3,t} \end{bmatrix} = \begin{bmatrix} X_{1,t-1} + U_{1,t} \cos(X_{3,t-1})\,dt \\ X_{2,t-1} + U_{1,t} \sin(X_{3,t-1})\,dt \\ X_{3,t-1} + U_{2,t}\,dt \end{bmatrix} + \varepsilon_t$$

Controller Design: set v to a constant

Using a carrot controller + PD control

$$e = \theta - \theta_c$$

$$= \theta - \operatorname{atan}\left[\frac{y_c - y}{x_c - x}\right]$$



$\therefore$ Let $\omega = K_p e + K_d (e - e_{last})$ ①

Question 1 - Continued

a) The robot motion will be updated with the motion model on the last page, by inputting the velocity ($v = u_{1,t}$) and rotation rate ($\omega = u_{2,t}$) inputs.

For simplicity, the velocity input will be constant throughout. This is realistic for many retail robots such as floor cleaners as it makes sense for the user to be able to assign a constant velocity that they are comfortable with based on their end application.

The rotation rate will be inputted by the controller shown in eq. 11 to ensure the robot follows the given path.

Assuming the robot has some sort of a contact or range sensor that tells it whether it has hit a dynamic obstacle, the robot will spin clockwise by 90° and continue on by giving it a fixed rotation rate and time period.

For static obstacles, avoidance will be incorporated in the planning process. The map features can be dialated to account for the size of the robot plus a buffer for any deviations from the path by the carrot controller

The path will be tracked by searching for the closest path point given the ekf pose estimate and the path vertices. (similar to assignment 2)

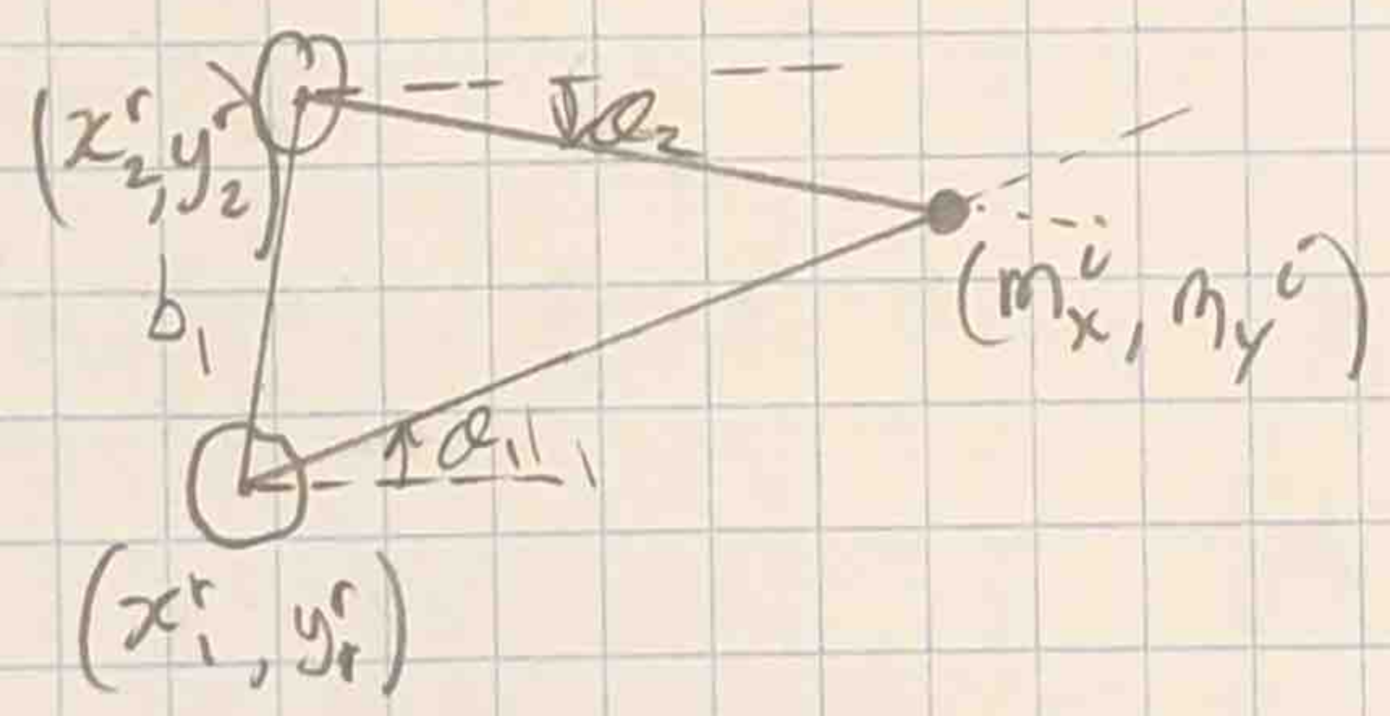
Question 1 - Continued

b) Measurement Model

$$\begin{bmatrix} y_{1,t}^i \\ y_{2,t}^i \end{bmatrix} = h^i(X_t) = \begin{bmatrix} \sqrt{X_{1,t}^2 + X_{2,t}^2} \\ \tan^{-1}\left[\frac{m_y^i - X_{2,t}}{m_x^i - X_{1,t}}\right] - X_{3,t} \end{bmatrix} + \delta_t$$

Initialization of the map features will be done by waiting until we have two measurements of the same feature, then triangulating to find the initial position

i.e.:



In reality, this triangulation will only work well with a large baseline, i.e. the robot needs to move a lot between each measurement. The best solution would be to only initialize once the covariance has dropped below an acceptable threshold

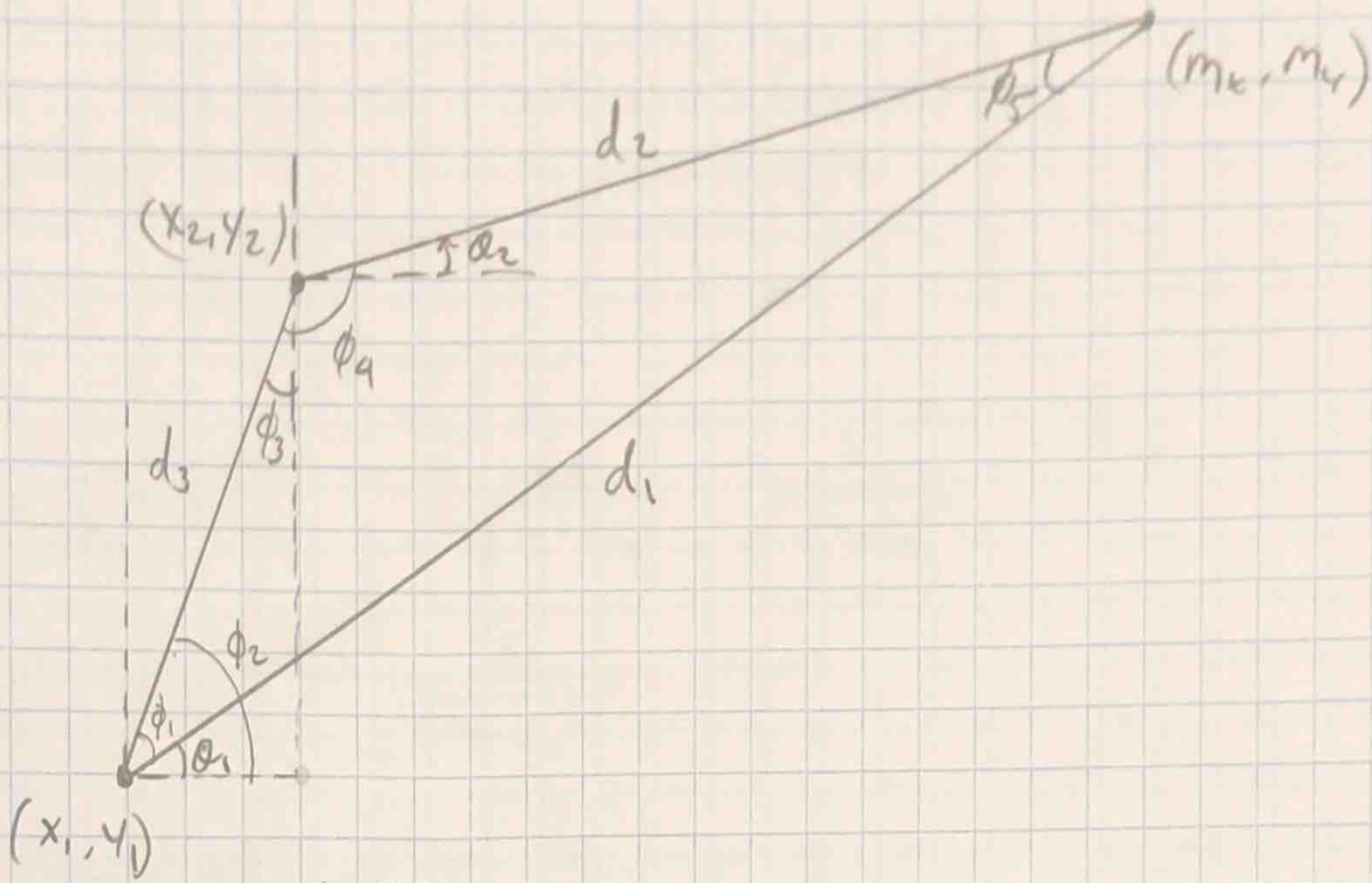
$$m_x^i = x_1^r + \frac{\sin\phi}{\sin(\theta_1-\theta_2)}\sqrt{(x_2^r-x_1^r)^2+(y_2^r-y_1^r)^2}\cos\theta_1$$

$$m_y^i = y_1^r + \frac{\sin\phi}{\sin(\theta_1-\theta_2)}\sqrt{(x_2^r-x_1^r)^2+(y_2^r-y_1^r)^2}\sin\theta_1$$

$$\phi = 180 + \theta_2 - \tan^{-1}\left[\frac{y_2^r-y_1^r}{x_2^r-x_1^r}\right]$$

[Question 1 continued]

b)



$\swarrow$ atan2(y,x)

$\phi_2 = \tan^{-1}\left[\frac{y_2 - y_1}{x_2 - x_1}\right]$, $\phi_1 = \phi_2 - \theta_1$, $\phi_3 = 90 - \phi_2$

$$\frac{d_1}{\sin\phi_4} = \frac{d_3}{\sin\phi_5}$$

$$\therefore d_1 = \frac{\sin\phi_4}{\sin\phi_5}\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

$$m_x = x_1 + d_1 \cos\theta_1$$

$$m_y = y_1 + d_1 \sin\theta_1$$

$$\phi_4 = 90 + \phi_3 + \theta_2$$
$$= 90 + 90 - \phi_2 + \theta_2$$
$$\phi_4 = 180 - \tan^{-1}[\cdots] + \theta_2$$
$$\phi_5 = 180 - \phi_4 - \phi_1$$
$$= \tan^{-1}[\cdots] - \theta_2 - (\phi_2 - \theta_1)$$
$$= -\theta_2 + \theta_1$$

$$\phi_4 = (90 - \phi_1) + 90 + \theta_2 = 90 - \left(\tan^{-1}\left(\frac{y_2 - y_1}{x_2 - x_1}\right) - \theta_1\right) + 90 + \theta_2 = 180 + \theta_1 + \theta_2 - \tan^{-1}\left(\frac{y_2 - y_1}{x_2 - x_1}\right)$$

$\boxed{\text{Question 1 - Continued}}$

b) <u>Linearization</u>

$$G_t = \begin{bmatrix} 1 & 0 & -u_{1,t} \sin x_{3,t-1}\, dt \\ 0 & 1 & u_{1,t} \sin x_{3,t-1}\, dt \\ 0 & 0 & 1 \end{bmatrix}$$

$$\frac{d}{dx_1}\left[x_1^2 + x_2^2\right]^{1/2} = \frac{1}{2} \frac{2x_1}{\sqrt{x_1^2+x_2^2}} = \frac{x_1}{\sqrt{x_1^2+x_2^2}}$$

$$\frac{d}{dx_2}\left[x_1^2 + x_2^2\right]^{1/2} = \frac{x_2}{\sqrt{x_1^2+x_2^2}}$$

$$\frac{d}{dx_1}\left[\tan^{-1}\left[\frac{m_y - x_2}{m_x - x_1}\right] - x_3\right] = \frac{dy}{r^2}$$

$$H_t^i = \begin{bmatrix} \frac{x_{1,t}}{\sqrt{x_{1,t}^2 + x_{2,t}^2}} & \frac{x_{2,t}}{\sqrt{x_{1,t}^2 + x_{2,t}^2}} & 0 & \cdots & & \longrightarrow & 0 \\ \frac{dy_t^i}{r^{i2}} & \frac{-dx_t^i}{r^{i2}} & -1 & \cdots & \frac{-dy_t^i}{r^2} & \frac{dx_t^i}{r^2} & 0 \cdots 0 \end{bmatrix}$$

where $r_t^i = \sqrt{(dx_t^i)^2 + (dy_t^i)^2}$

$$dx_t^i = m_x^i - x_{1,t}$$

$$dy_t^i = m_y^i - x_{2,t}$$

[Question 1 - Continued]

c) EKF

The EKF has 3 steps:

1. Motion Update (Prediction step)

$$\bar{\mu}_t = g(\mu_{t-1}, u_t)$$

$$\bar{\Sigma}_t = G_t \Sigma_{t-1} G_t^T + R_t$$

where G was defined before

2. Measurement Update 1 (beacon)

$$h = \sqrt{x_{1,t}^2 + x_{2,t}^2} + \delta_t$$

$$H_t = \left[\frac{x_{1,t}}{\sqrt{x_{1,t}^2 + x_{2,t}^2}} \quad \frac{x_{2,t}}{\sqrt{x_{1,t}^2 + x_{2,t}^2}} \quad 0 \; \cdots \; 0 \right]$$

$$K_t = \bar{\Sigma}_t H_t^T (H_t \bar{\Sigma}_t H_t^T + Q_t)^{-1}$$

$$\mu_t = \bar{\mu}_t + K_t (y_t - h(\bar{\mu}_t))$$

$$\Sigma_t = (I - K_t H_t) \bar{\Sigma}_t$$

3. Measurement Update for all features in view

$$h^i = \tan^{-1}\left[\frac{m_y^i - x_{2,t}}{m_x^i - x_{1,t}}\right] - x_{3,t} + \delta_t$$

$$H_t^i = \left[\frac{dy_t}{r^{i2}} \quad \frac{-dx_t^i}{r^{i2}} \quad -1 \quad \ldots, \frac{-dy_t^i}{r^2} \quad \frac{dx_t^i}{r^2} \quad 0 \cdots 0 \right]$$

$$K_t^i = \bar{\Sigma}_t (H_t^i)^T (H_t^i \bar{\Sigma}_t (H_t^i)^T + Q_t)^{-1}$$

$$\mu_t = \bar{\mu}_t + K_t^i (y_t - h(\bar{\mu}_t))$$

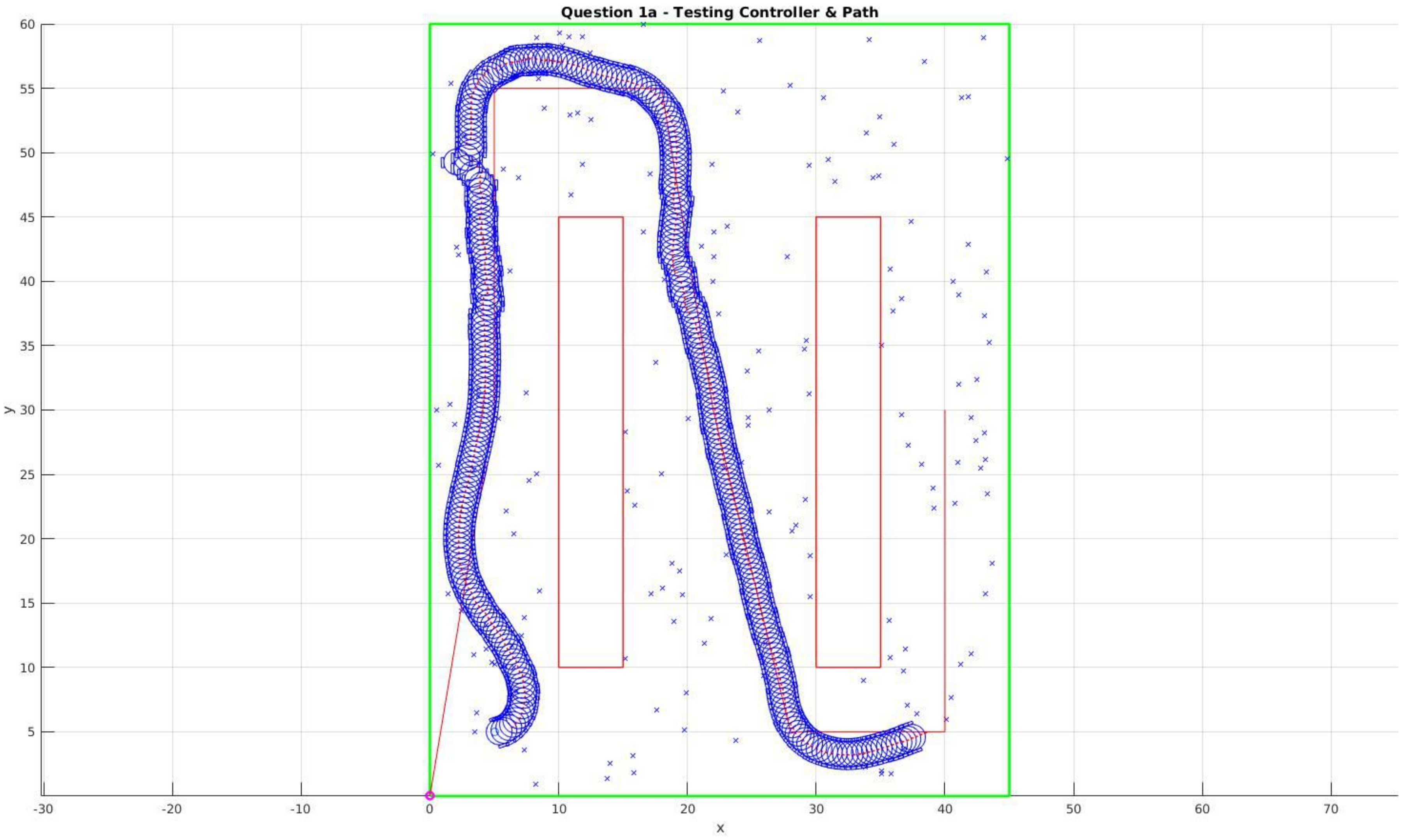
$$\Sigma_t = (I - K_t H_t^i) \bar{\Sigma}_t$$

[Question 1 - Continued]

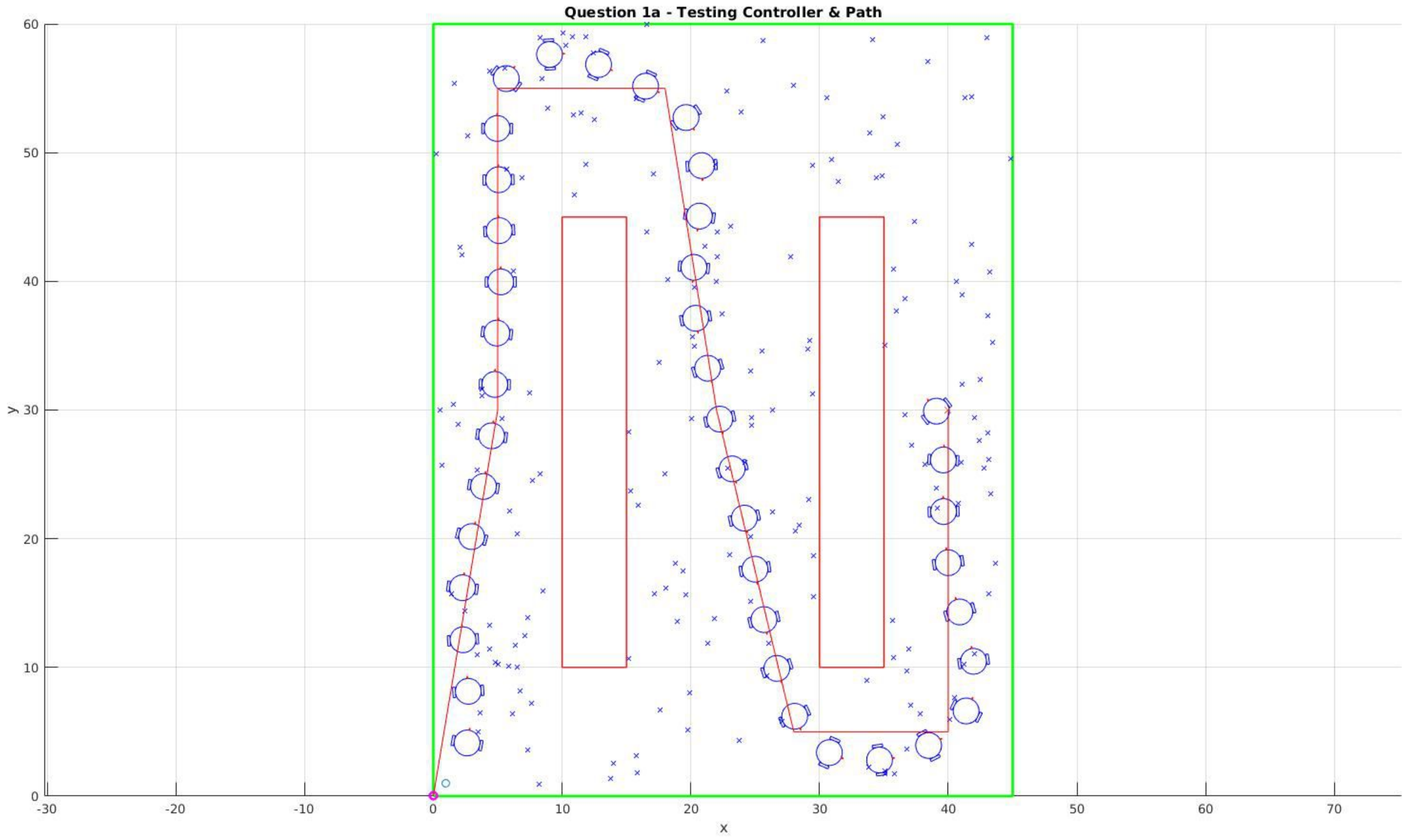
d) No time to implement.

If there were correspondence errors, I would implement RANSAC as an outlier rejection method.

A feature measurement would be used for SLAM if the estimated pose from that measurement corresponds with the other measurements at that timestep. Good practice for SLAM would be to select a given number of strongest measurements. The "strength" measure would be how well that measurement corresponds with a sample of measurements.



Question 1a - Testing Controller & Path
y
x


Question 1a - Testing Controller & Path
y
x
60
50
40
30
20
10
0
-30
-20
-10
0
10
20
30
40
50
60
70

Nicholas Charron
20440122

Question 2 - Bicycle State Estimation

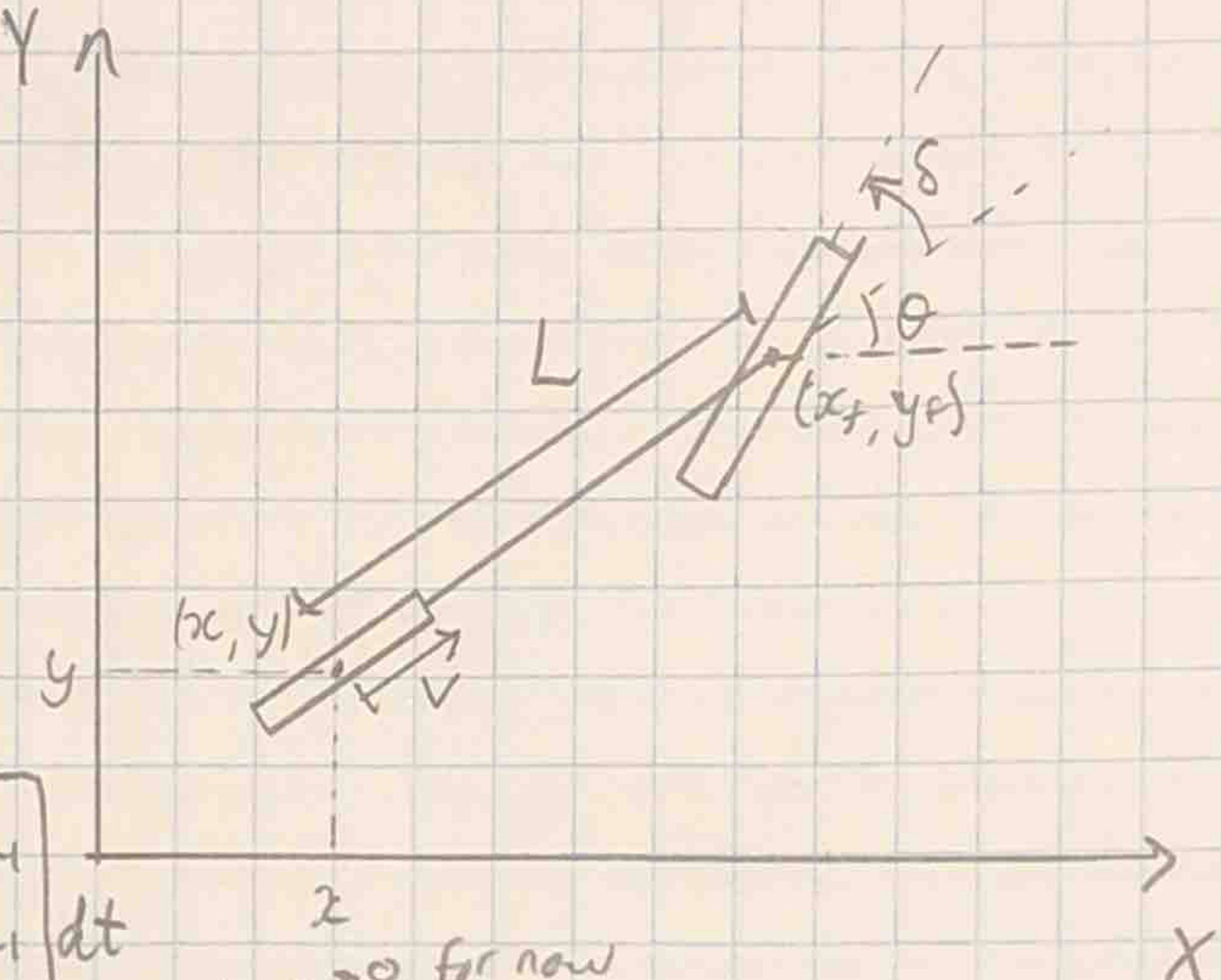
a) $f = 2\text{Hz}$, $dt = 0.5s$, $T_t = 100s$

$L = 1m$

$v = 8 m/s$

$\delta_t = 0.1\sin\left(\frac{t}{10}\right) + 0.05\cos\left(\frac{t}{12}\right) + \frac{t}{16}$

Plot every 10 timesteps
(every 5 seconds)



Motion Model:

$$\begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix} = \begin{bmatrix} X_{1,t} \\ X_{2,t} \\ X_{3,t} \end{bmatrix} = \begin{bmatrix} X_{1,t-1} \\ X_{2,t-1} \\ X_{3,t-1} \end{bmatrix} + u_{1,t} \begin{bmatrix} \cos X_{3,t-1} \\ \sin X_{3,t-1} \\ \frac{\tan u_{2,t}}{L} \end{bmatrix} dt + \cancel{\varepsilon_t}^{0} \text{ for now}$$

See Figure 2a-1

Question 2 - Continued

b)

$$X_t = \begin{bmatrix} X_{1,t} \\ X_{2,t} \\ X_{3,t} \\ X_{4,t} \\ X_{5,t} \end{bmatrix} = \begin{bmatrix} x_t \\ y_t \\ \theta_t \\ v_t \\ \delta_t \end{bmatrix} \qquad u = [\;] \quad \text{: no inputs}$$

Motion model

$$X_t = \begin{bmatrix} x_{1,t-1} \\ x_{2,t-1} \\ x_{3,t-1} \\ x_{4,t-1} \\ x_{5,t-1} \end{bmatrix} + \begin{bmatrix} x_{4,t-1}\cos(x_{3,t-1})dt \\ x_{4,t-1}\sin(x_{3,t-1})dt \\ x_{4,t-1}\tan(x_{5,t-1})dt/L \\ 0 \\ 0 \end{bmatrix} + \varepsilon_{m,t}$$

Where $\varepsilon_{m,t} \sim N(0, \Sigma_m)$

$$\Sigma_m = \begin{bmatrix} 0.001 & 0 & 0 & 0 & 0 \\ 0 & 0.001 & 0 & 0 & 0 \\ 0 & 0 & 0.0005 & 0 & 0 \\ 0 & 0 & 0 & 0.01 & 0 \\ 0 & 0 & 0 & 0 & 0.004 \end{bmatrix}$$

Constraints: $-25° \le x_{5,t} \le 25°$

Question 2 - Continued

c) GPS Measurement Model

over-complicated, see next page

$Y_1 = x = X_1$
$Y_2 = y = X_2$
$Y_3 = v_x = \dot{X}_1$
$Y_4 = v_y = \dot{X}_2$

instead of adding more states to our model, let's convert $\dot{X}_1$ & $\dot{X}_2$ to a heading measurement, θ, and velocity measurement, v.

$$X_{4,t} = v_t = \sqrt{v_{x,t}^2 + v_{y,t}^2} = \sqrt{Y_{3,t}^2 + Y_{4,t}^2}$$

$$X_{3,t} = \theta_t = \tan^{-1}\left[\frac{v_{y,t}}{v_{x,t}}\right] = \tan^{-1}\left[\frac{Y_{4,t}}{Y_{3,t}}\right]$$

this is the inv. measurement model, let's convert it

$X_4^2 = Y_3^2 + Y_4^2$ -①, $Y_3 \tan X_3 = Y_4$ -② (dropping t subscripts)

② into ① : $X_4^2 = Y_3^2 + Y_3^2 \tan^2 X_3 \therefore Y_3 = \dfrac{X_4}{\sqrt{1 + \tan^2 X_3}}$

Putting it all together:

$$Y_{g,t} = \begin{bmatrix} Y_{1,t} \\ Y_{2,t} \\ Y_{3,t} \\ Y_{4,t} \end{bmatrix} = \begin{bmatrix} X_{1,t} \\ X_{2,t} \\ X_{4,t} \cdot \left[1 + \tan^2(X_{3,t})\right]^{-1/2} \\ \tan(X_{3,t}) \cdot X_{4,t}\left[1 + \tan^2(X_{3,t})\right]^{-1/2} \end{bmatrix} + \varepsilon_g,$$

where $\varepsilon_g \sim N(0, \Sigma_g)$

$$\Sigma_g = \begin{bmatrix} 0.5^2 & 0 & 0 & 0 \\ 0 & 0.5^2 & 0 & 0 \\ 0 & 0 & 1^2 & 0 \\ 0 & 0 & 0 & 1^2 \end{bmatrix}$$

Steering Measurement Model

$Y_{s,t} = X_{5,t} + \varepsilon_{s,t}$, where $\varepsilon_{s,t} \sim N(0, \Sigma_s)$

$$\Sigma_s = 4\,deg^2 \times \frac{(1 \cdot \pi/180)^2\, rad^2}{(1)\, deg^2}$$

$$= 0.0012\, rad^2$$

c) Measurement Model for GPS & steering

$y_1 = x$
$y_2 = y$
$y_3 = \dot{x}$
$y_4 = \dot{y}$
$y_5 = v$

$y_1 = x_{1,t}$
$y_2 = x_{2,t}$
$y_3 = x_{4,t} \cos x_{3,t}$
$y_4 = x_{4,t} \sin x_{3,t}$
$y_5 = x_{5,t}$

$$\therefore \quad y_t = \begin{bmatrix} x_{1,t} \\ x_{2,t} \\ x_{4,t} \cos(x_{3,t}) \\ x_4 \sin(x_{3,t}) \\ x_{5,t} \end{bmatrix} + \varepsilon_t$$

where $\varepsilon_t \sim N(0, \Sigma_t)$, $\Sigma_t = \begin{bmatrix} 0.5^2 & 0 & 0 & 0 & 0 \\ 0 & 0.5^2 & 0 & 0 & 0 \\ 0 & 0 & 1^2 & 0 & 0 \\ 0 & 0 & 0 & 1^2 & 0 \\ 0 & 0 & 0 & 0 & \left(\frac{\pi}{90}\right)^2 \end{bmatrix}$

Question 2 - Continued

d) $f = 1\ Hz$: freq. of measurements
$T = 1s$: time between each measurement
$C = 2.1m$: circumference of wheel

$P_{miss} = 1/100 = 0.01$

Let n = number of pulses outputted by microcontroller in $1T = 1s$

Assume no slip.

$$v = \frac{distance}{time} = \frac{n \cdot C}{T} = 2.1n \quad , \therefore \quad n = \frac{v}{2.1} = \frac{1}{2.1} \cdot x_4$$

Measurement Model

$$Y_{\omega,t} = n_t = \frac{1}{2.1} x_{4,t} - \varepsilon_\omega$$

or $$Y_{\omega,t} = \frac{T}{C} x_{4,t} - \varepsilon_\omega$$

where ε_ω is drawn from a poisson distribution

$\varepsilon_\omega \sim P(\mu)$

μ: mean number of events per time period, T

$\mu = 0.01 \frac{miss}{rev} \cdot \frac{k\ rev}{s} \times 1s \quad , k = \frac{vT}{c}$

$\mu = \frac{0.01}{2.1} v$

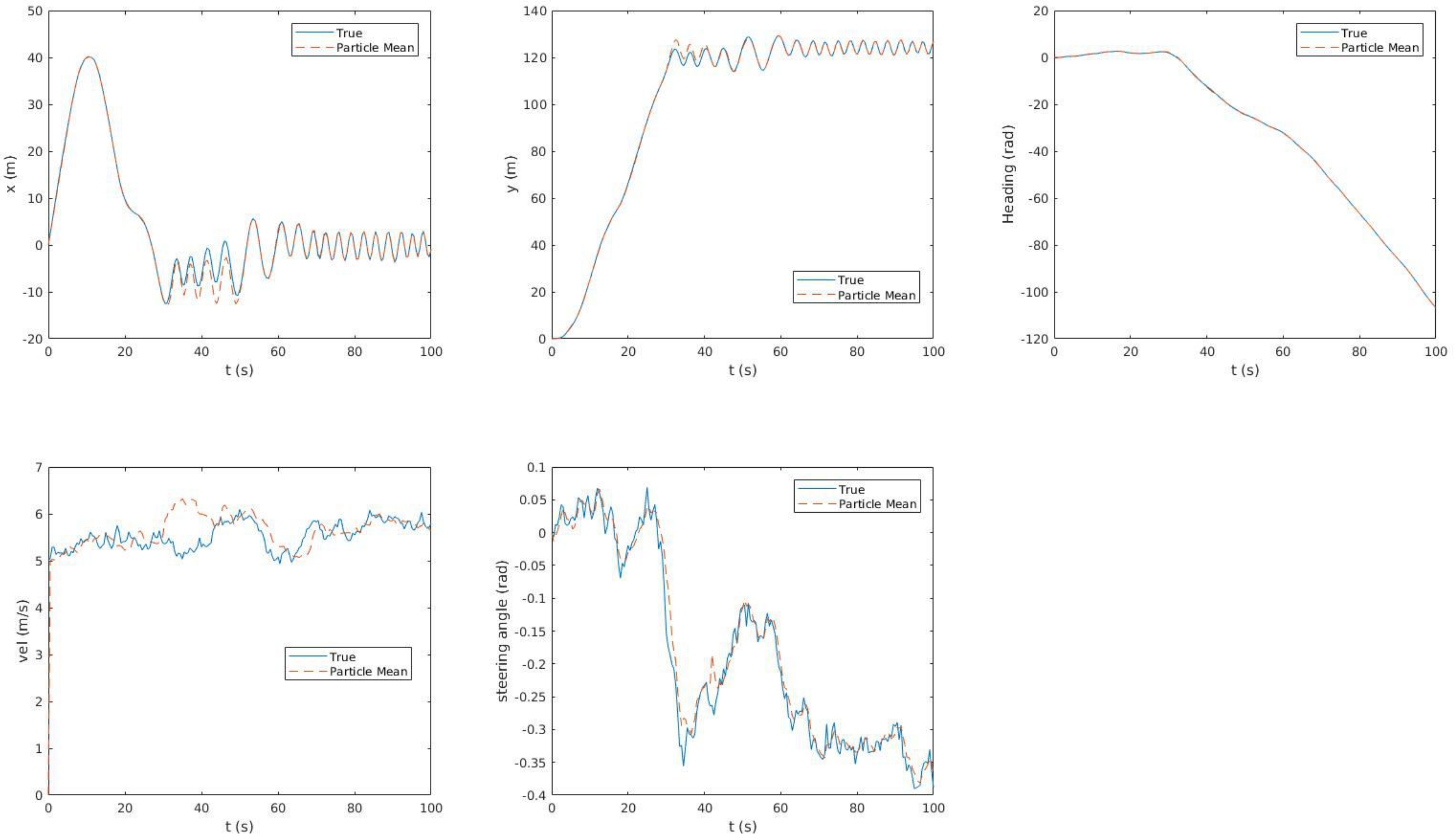
→ come back

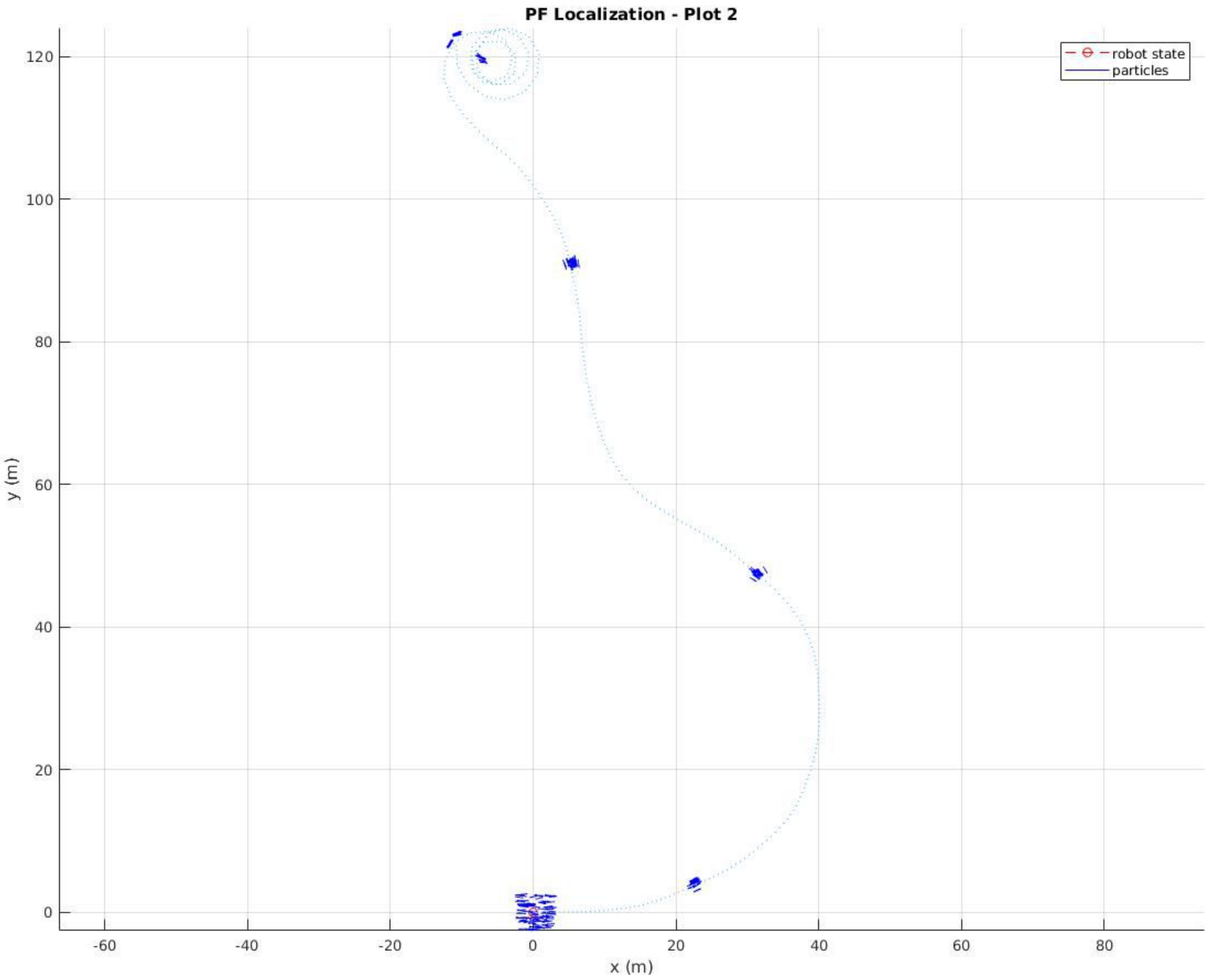
[Question 2 - Continued]

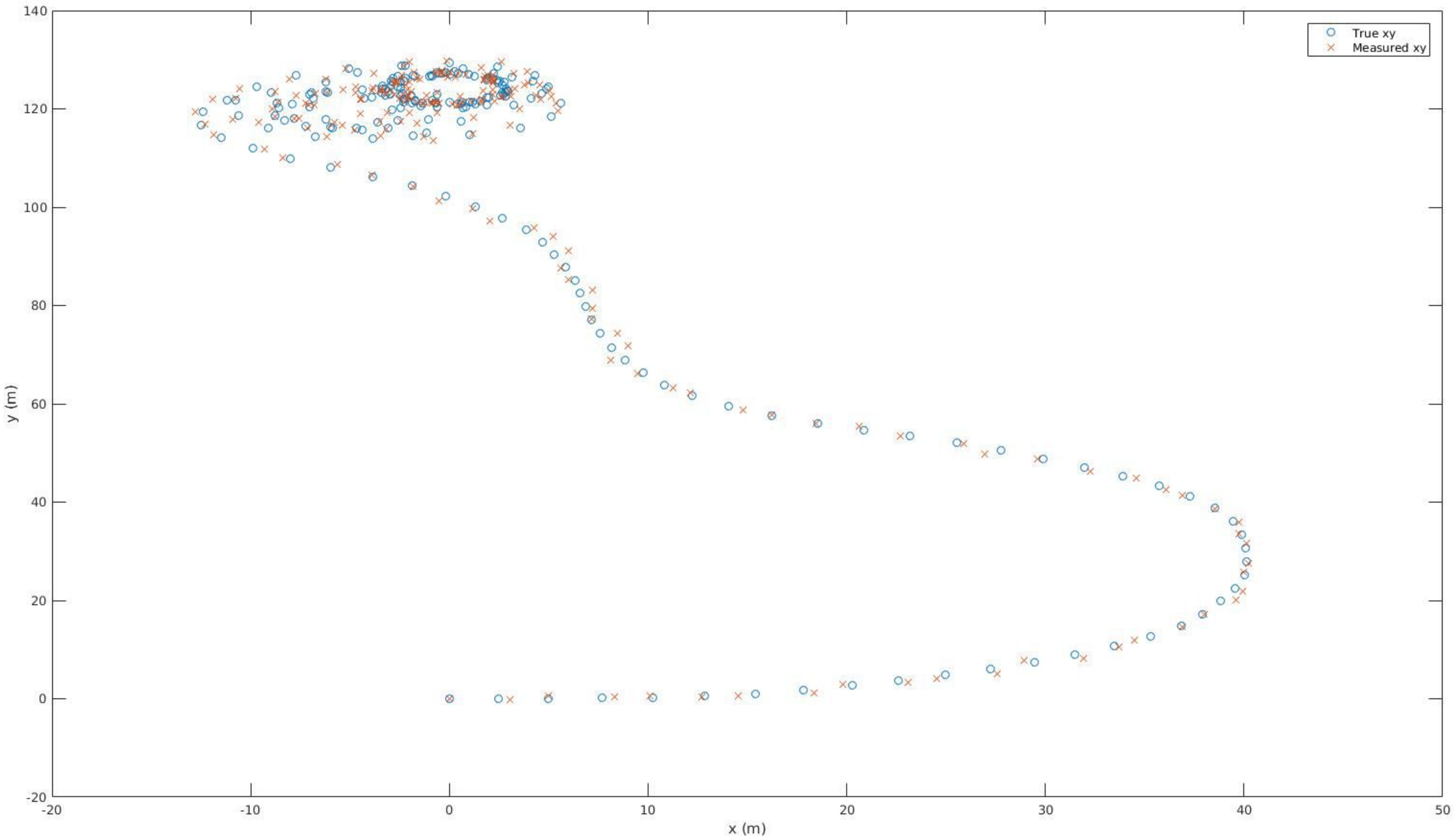
e) See Fig 2e-1 to 2e-3

I selected 100 particles which turned out to be an appropriate number to get decent accuracy without particle deprivation and still running at a rapid rate.

Particle deprevation was not an issue for me, but if it were, I would increase my noise in the propagation step.


x (m)
y (m)
Heading (rad)
vel (m/s)
steering angle (rad)
t (s)
True
Particle Mean


PF Localization - Plot 2
robot state
particles
y (m)
x (m)
120
100
80
60
40
20
0
-60
-40
-20
0
20
40
60
80



True xy
Measured xy
x (m)
y (m)
-20
-10
0
10
20
30
40
50
140
120
100
80
60
40
20
0
-20

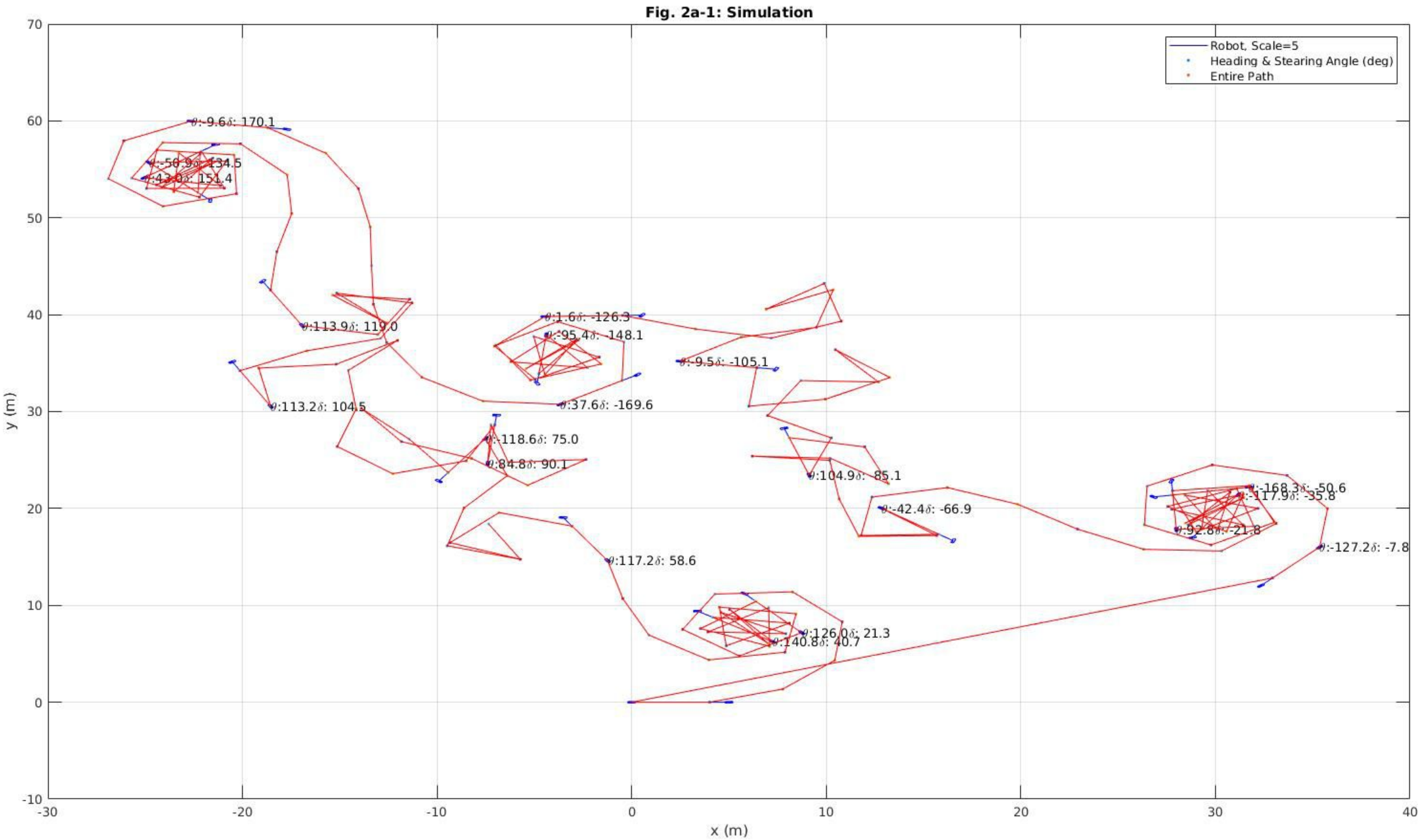


Fig. 2a-1: Simulation
Robot, Scale=5
Heading & Stearing Angle (deg)
Entire Path
x (m)
y (m)
θ:-9.6δ: 170.1
θ:-56.9δ: 134.5
θ:-43.0δ: 151.4
θ:113.9δ: 119.0
θ:1.6δ: -126.3
θ:-95.4δ: -148.1
θ:-9.5δ: -105.1
θ:113.2δ: 104.5
θ:37.6δ: -169.6
θ:-118.6δ: 75.0
θ:84.8δ: 90.1
θ:104.9δ: -85.1
θ:-168.3δ: -50.6
θ:-117.9δ: -35.8
θ:-42.4δ: -66.9
θ:92.8δ: -21.8
θ:-127.2δ: -7.8
θ:117.2δ: 58.6
θ:126.0δ: 21.3
θ:140.8δ: 40.7

Question 3 - Drone Racing

a) States:

$$X_t = \begin{bmatrix} X_{1,t} \\ X_{2,t} \\ X_{3,t} \\ X_{4,t} \end{bmatrix} = \begin{bmatrix} x \\ \dot{x} \\ y \\ \dot{y} \end{bmatrix}$$

Inputs:

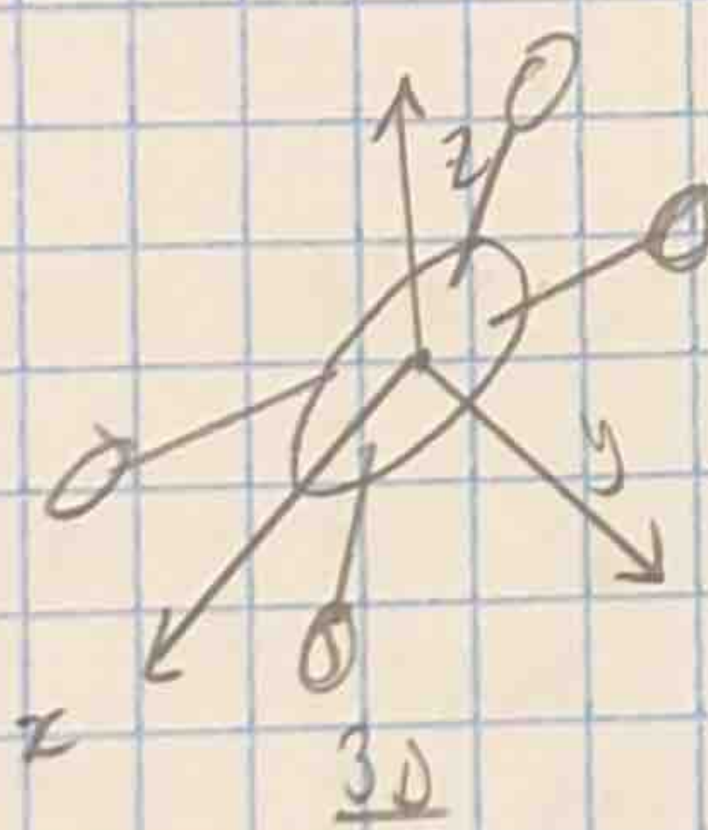
$$u_t = \begin{bmatrix} u_{1,t} \\ u_{2,t} \end{bmatrix} = \begin{bmatrix} a_x \\ a_y \end{bmatrix}$$

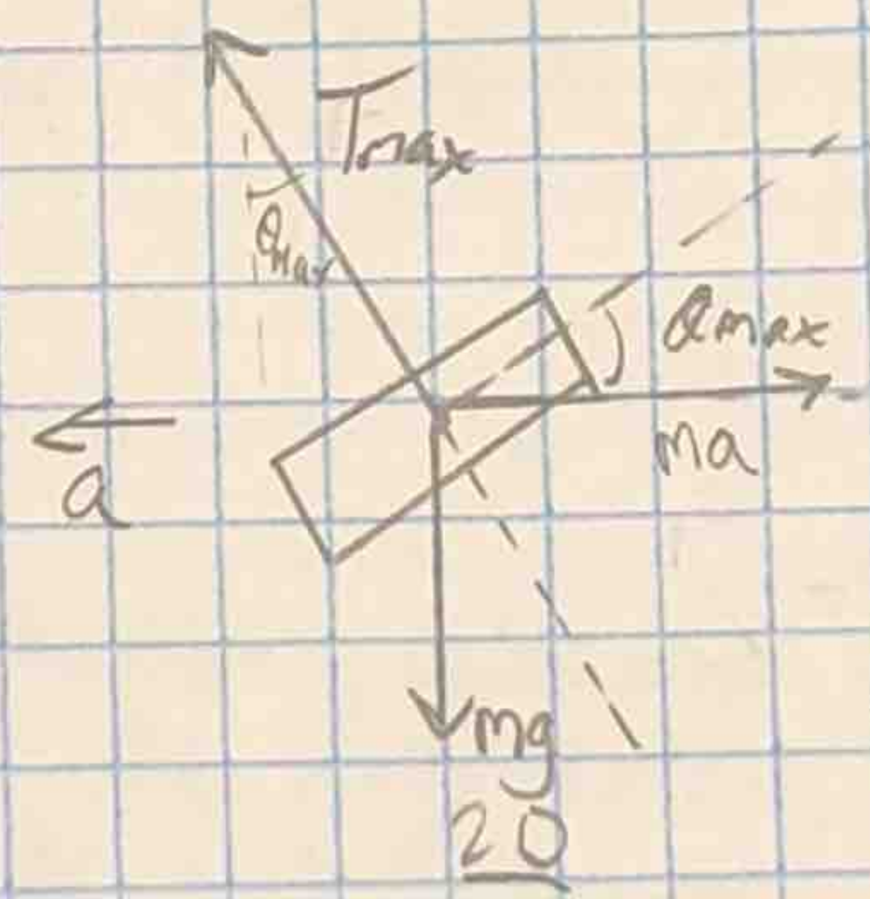
Σ_t

$m = 1kg$, $T_{max} = 2mg$ ⇐ Assuming this means 2 x mass x acc of gravity

$T_{max} = 2 \times 1 \times 9.81 = 19.6 N$

FBD





Assume $v = 0$
∴ No drag

$\overset{+}{\rightarrow} \Sigma F_{x'} = -ma + T_{max} \sin\theta_{max} = 0$

$$a_{max} = \frac{T_{max} \sin\theta_{max}}{m} \quad -①$$

$+\uparrow \Sigma F_{y'} = 0 = -mg + T_{max} \cos\theta_{max}$

$$\therefore \theta_{max} = \cos^{-1}\left(\frac{mg}{T_{max}}\right) = \cos^{-1}\left(\frac{1}{2}\right) = 60°$$

$$a_{max} = \frac{2 \times 9.81 \sin 60}{1} = 17 m/s^2$$

∴ The max bank angle is 60°, and the max lateral acceleration is $17 m/s^2$

[Question 3 - Continued]

a) Motion Model

$$\begin{bmatrix} x_{1,t} \\ x_{2,t} \\ x_{3,t} \\ x_{4,t} \end{bmatrix} = \begin{bmatrix} 1 & dt & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & dt \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{1,t-1} \\ x_{2,t-1} \\ x_{3,t-1} \\ x_{4,t-1} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ dt & 0 \\ 0 & 0 \\ 0 & dt \end{bmatrix} \begin{bmatrix} u_{1,t} \\ u_{2,t} \end{bmatrix} + \varepsilon_t$$

Where $-17 m/s^2 \leq u_{1,t} \leq 17 m/s^2$

$-17 m/s^2 \leq u_{2,t} \leq 17 m/s^2$

Maximum Velocity

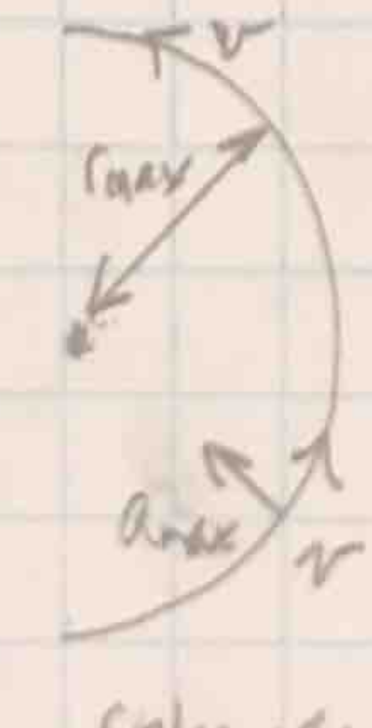
We know that there is a maximum velocity that the UAV can travel based on the drag forces, but since we have no information on the UAV size/shape, assume a maximum velocity, v_{max}

To simplify, assume that we can accelerate at a_{max} in any direction, until reaching v_{max} at which case our velocity peaks (despite the acceleration inputs)

Let $v_{max} = 10 m/s$

the max. turn radius we can make is:

$$r_{max} = \frac{v^2}{a_{max}}$$



(plan view)

Question 3 - Continued

*Note: Tried with PRM and kept getting stuck inside 'M'

b) The proposed motion planner will have 2 steps:

1) Generate optimal path without considering motion constraints

2) Use trajectory rollout with a given set of desired inputs and selecting the inputs that result in minimizing the distance to the path & distance to the goal

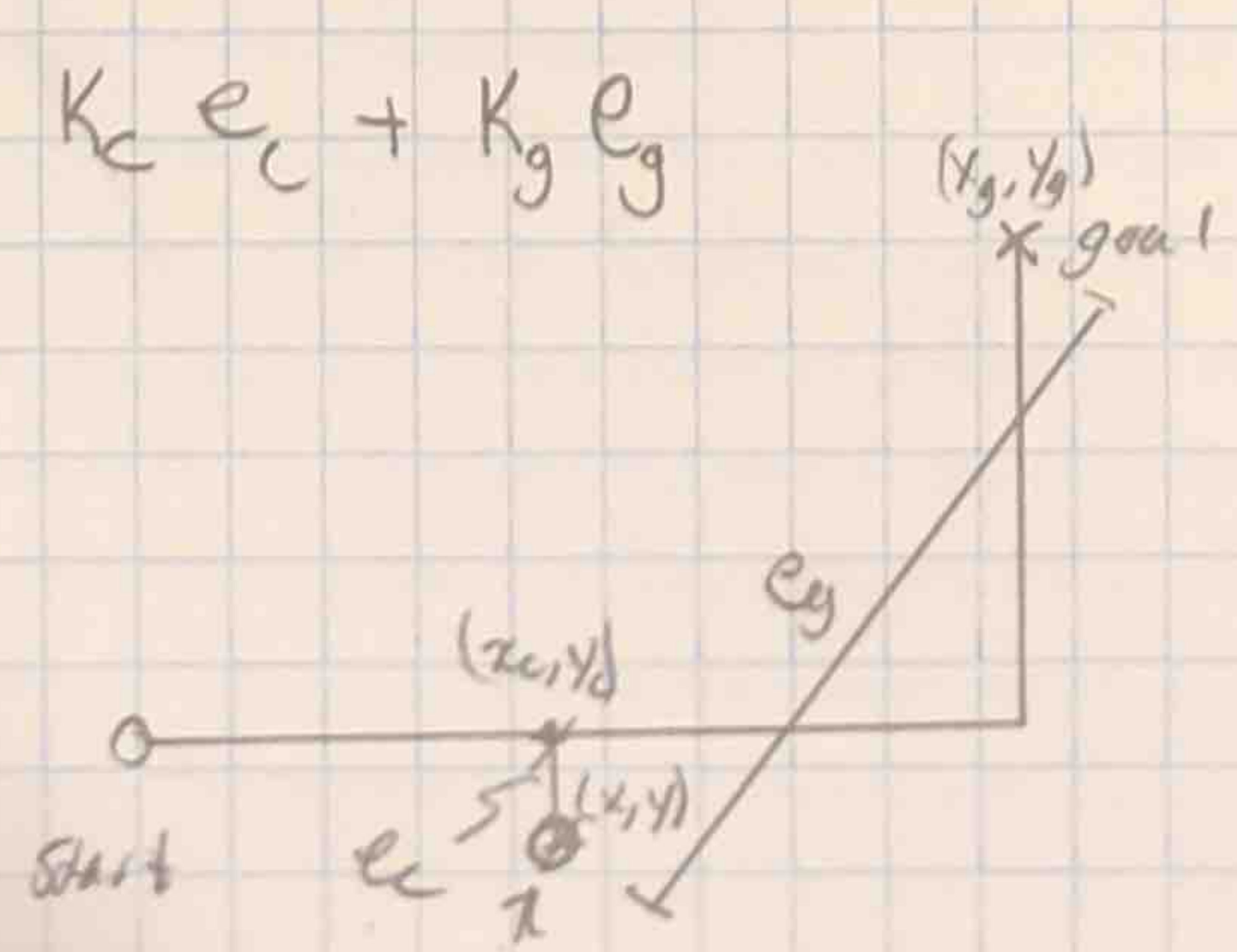
Step 1 has been completed using the wavefront algorithm with a dialated version of the map to add distance from obstacles. See Fig. 3b-1

Step 2, we will use the motion model, V_{max} & a_{max} constraints. Since this is for racing drones, we want max speed, therefore the following will be the inputs.

$$[u_1, u_2] = \begin{bmatrix} -17 & 0 & 17 & 0 & -8.5 & +8.5 \\ 0 & -17 & 0 & 17 & 8.5 & -8.5 \end{bmatrix}^T m/s^2$$

*Assume the max accelerations we can give in both directions at a time are $a_{max}/2 = 8.5 m/s^2$

Let Cost, $C = K_c e_c + K_g e_g$



$e_g = \sqrt{(x_g - x)^2 + (y_g - y)^2}$ → Probably a good LQR Problem?

$e_c = \sqrt{(x_c - x)^2 + (y_c - y)^2}$

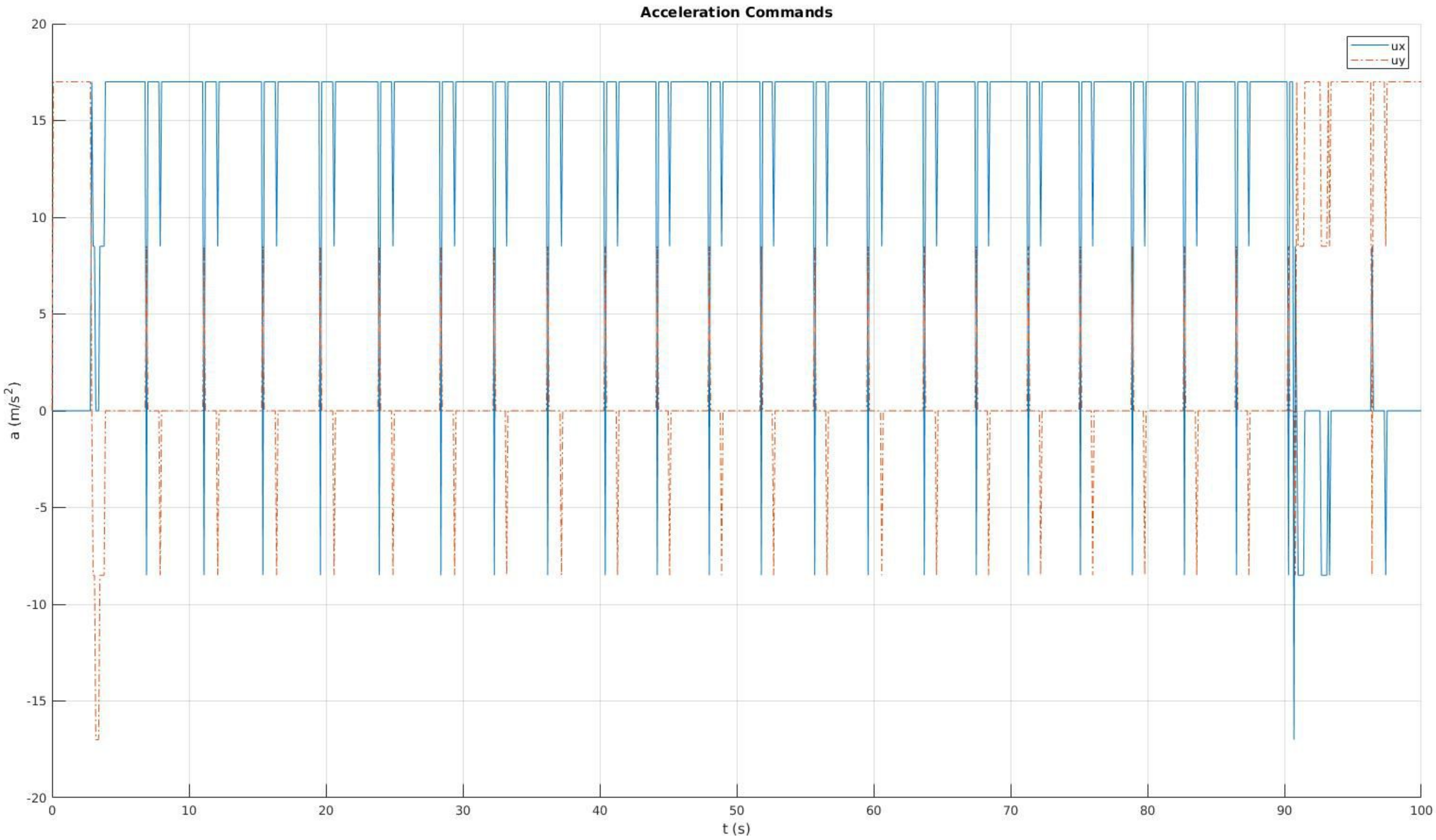
[Question 3 - Continued]

c) See Fig. 3c-1 to Fig 3c-4

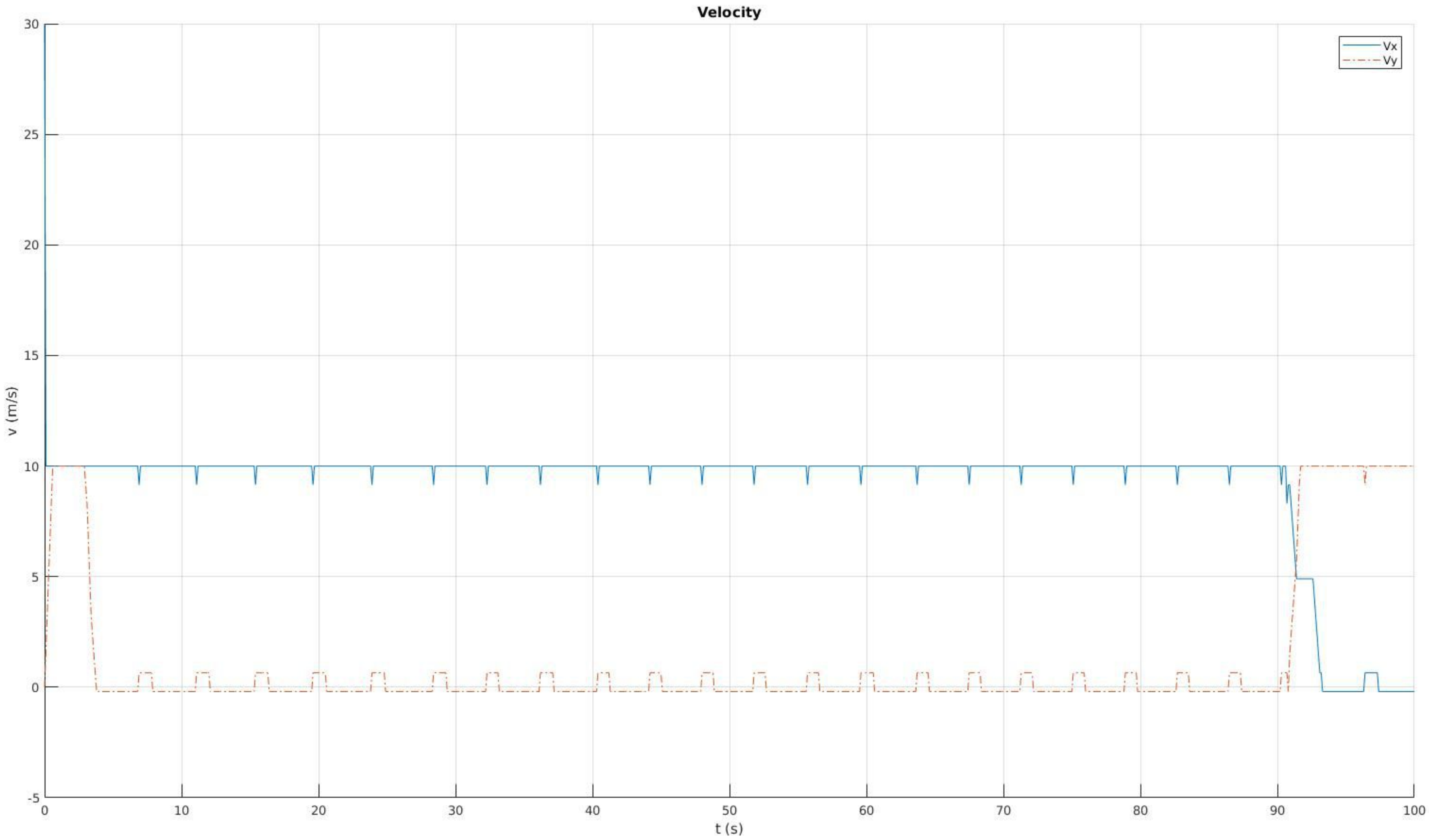
Fig 3c-1 shows the path from the wavefront algorithm. In Fig. 3c-2 I tried to implement the trajectory rollout but had some bugs that I couldn't fix in time. (Maybe in the motion model). As you can see from Fig. 3c-3 & Fig. 3c-4, the velocity & acceleration constraints were met.

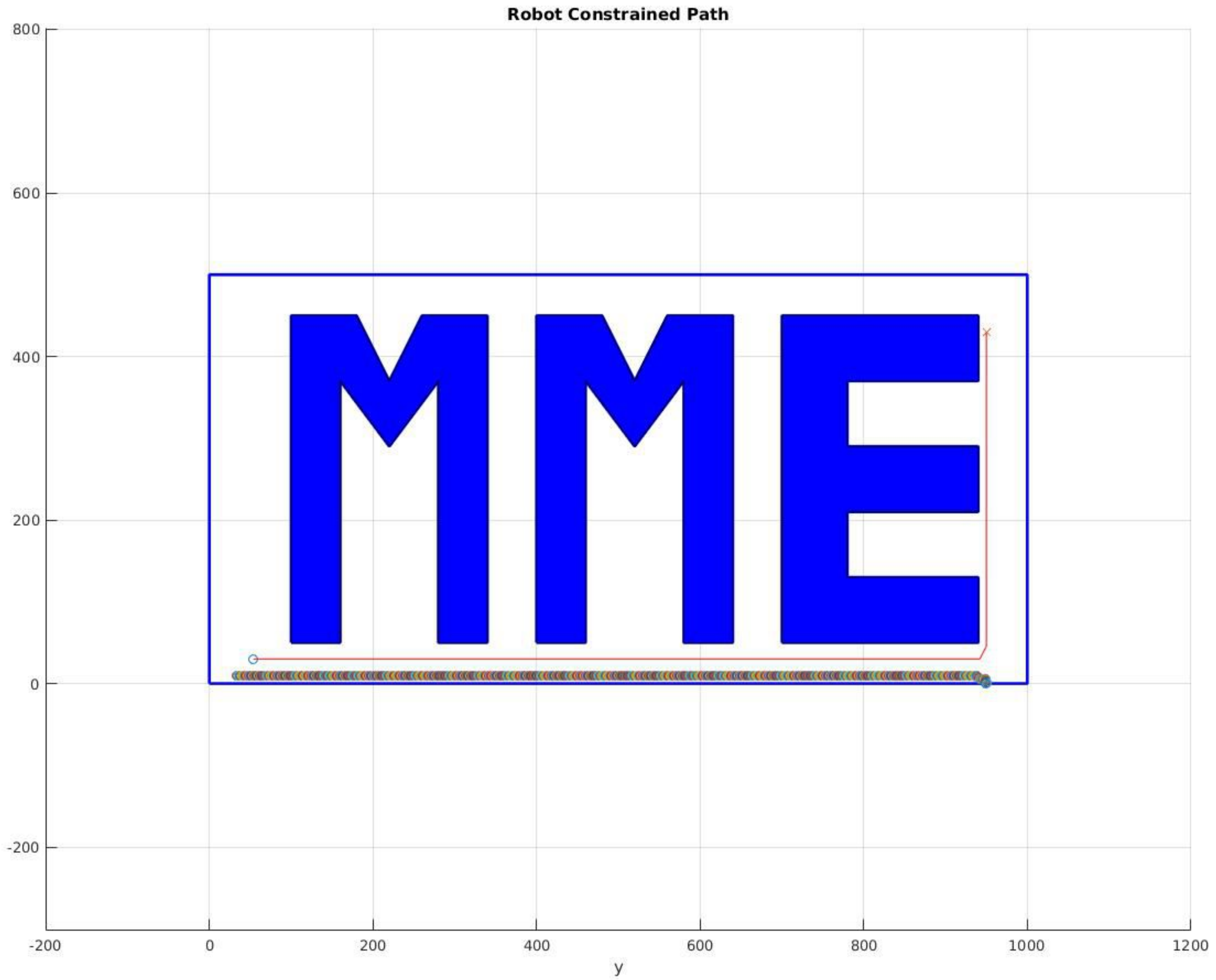
d) My planner was designed with an optimal route in mind. The wavefront path is an optimal path and the trajectory rollout was designed to stay as close as possible to the path by minimizing a cost function.

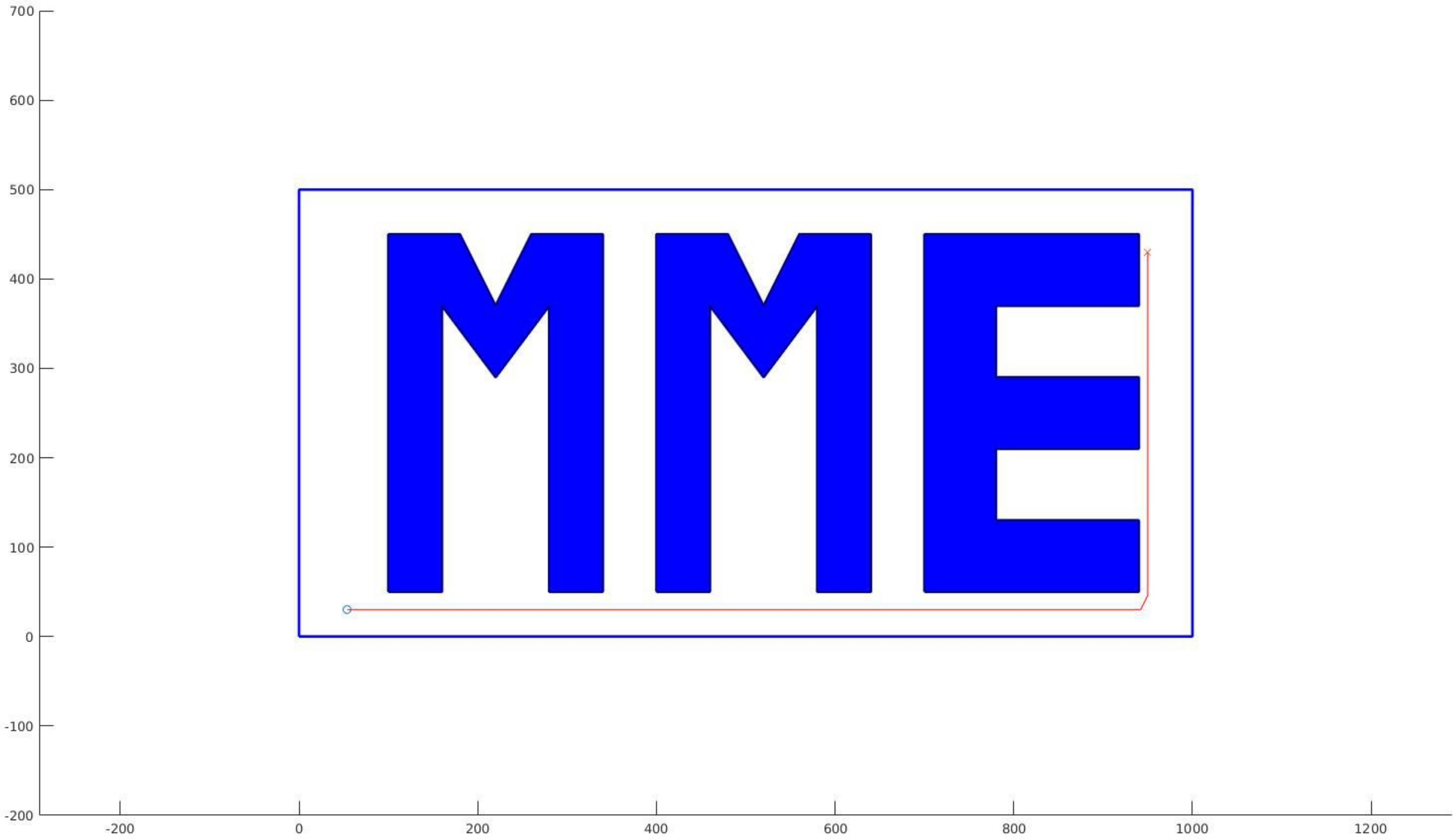
e) Implemented in C



Acceleration Commands
ux
uy
a (m/s²)
t (s)
20
15
10
5
0
-5
-10
-15
-20
0
10
20
30
40
50
60
70
80
90
100


Velocity
Vx
Vy
v (m/s)
t (s)
30
25
20
15
10
5
0
-5
0
10
20
30
40
50
60
70
80
90
100


Robot Constrained Path
800
600
400
200
0
-200
-200
0
200
400
600
800
1000
1200
y



700
600
500
400
300
200
100
0
-100
-200
-200
0
200
400
600
800
1000
1200