GNC Homework 1

Table of Contents

Bailey Waterman, Keshuai Xu

1a

We set up the axes that y is due north and the x-y plane is tangent to the earth surface. Turning 90 degrees in x-y plane will result in **almost** East heading. We ignore the curvature of the earth since the distance between Worcester and Boston is relatively small.

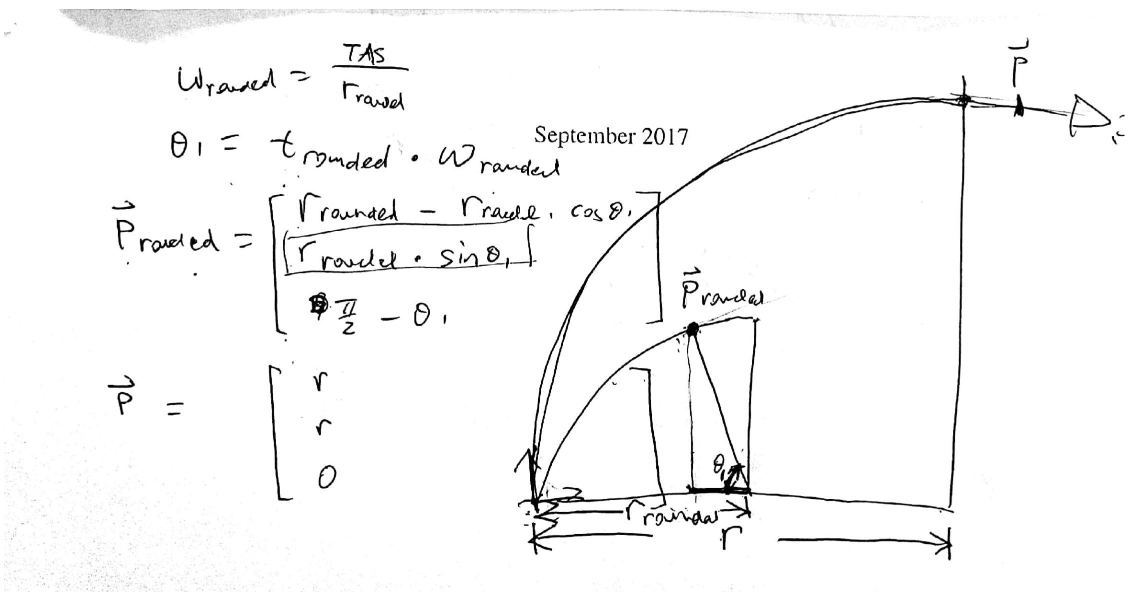
```
TAS = 70; %m/s
g = 9.8039; %gravitational constant at Boston (m/s^2)
bank = 20*pi/180; %radians
syms r;
% Angle of bank formula
% http://www.luizmonteiro.com/Article_Bank_Angle_for_Std_Rate_01.aspx
turn_radius = vpasolve(bank-atan(((TAS^2)/r)/g),r,1000); % m
time = vpa(((90*pi/180)*turn_radius)/TAS, 6) % s
```

```
time =

30.8144
```

1b

ignoring the curvature of the earth



September 2017

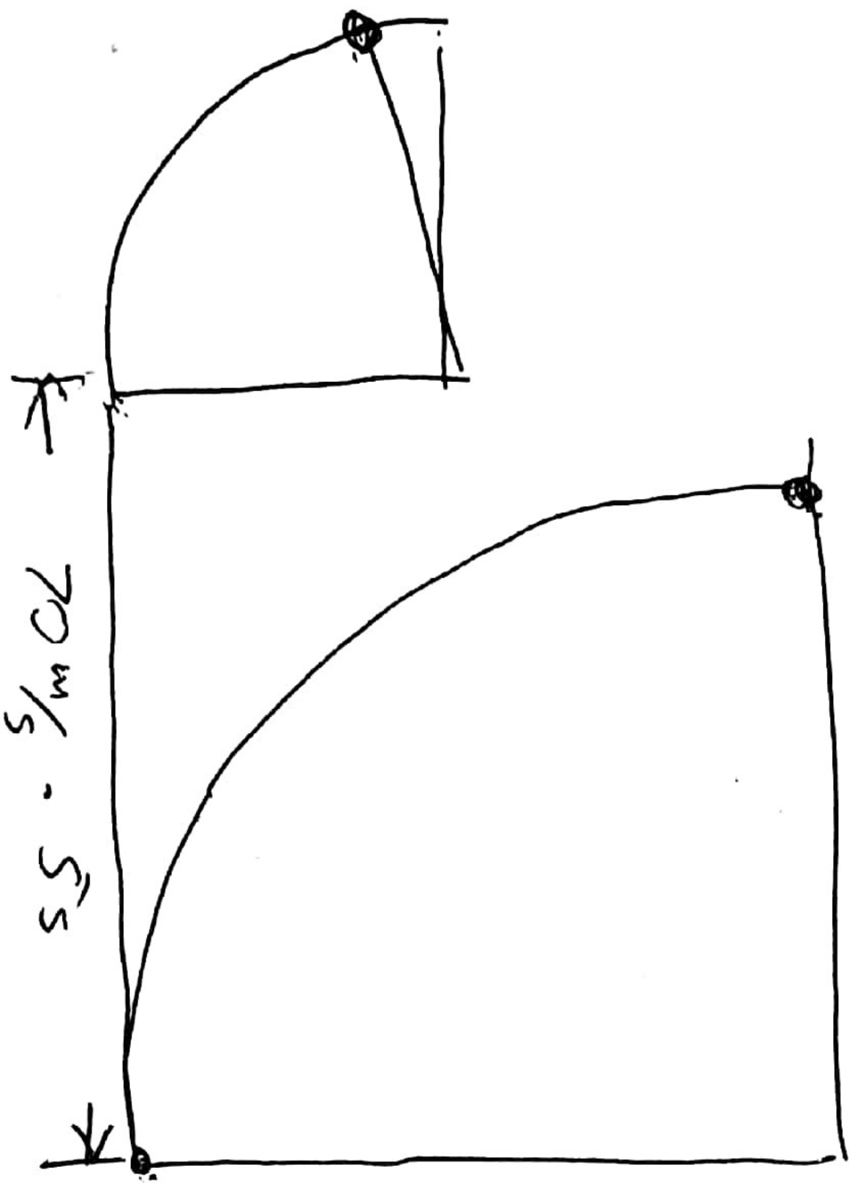
```
g_rounded = 10; %(m/s^2)
turn_radius_rounded = vpasolve(bank-atan(((TAS^2)/r)/
g_rounded),r,1000); % m
time_rounded = round(((90*pi/180)*turn_radius_rounded)/TAS); % s

omega_rounded = TAS/turn_radius_rounded; % rad/s
theta_rounded = time_rounded*omega_rounded; % rad
P_rounded = [turn_radius_rounded - turn_radius_rounded *
 cos(theta_rounded); % x in m
    turn_radius_rounded * sin(theta_rounded); % y in m
    pi/2 - theta_rounded]; % heading in rad wrt x axis
P = [turn_radius; turn_radius; 0]; % [m; m; rad]
error = P_rounded - P; % error in [x, y, heading]. unit in [m; m; rad]
vpa(error, 6)
```

```
ans =

  -41.6345
  -27.0086
 0.0109239
```

1c



```
P_rounded = [turn_radius_rounded - turn_radius_rounded *
 cos(theta_rounded); % x in m
    turn_radius_rounded * sin(theta_rounded) + TAS*5; % y in m
    pi/2 - theta_rounded]; % heading in rad wrt x axis
P = [turn_radius; turn_radius; 0]; % [m; m; rad]
error = P_rounded - P; % error in [x, y, heading]. unit in [m; m; rad]
vpa(error, 6)
```

```
ans =

  -41.6345
   322.991
 0.0109239
```

2a

$$\|P_r - P_i\| + \varepsilon_c = \rho_i \qquad (2.1)$$

$$\|P_r - P_i\| = \sqrt{(P_x - P_{xi})^2 + (P_y - P_{yi})^2 + (P_z - P_{zi})^2} \qquad (2.2)$$

$$\left.\begin{aligned}
\|P_r - P_1\| + \varepsilon_c - \rho_1 &= 0\\
\|P_r - P_2\| + \varepsilon_c - \rho_2 &= 0\\
\|P_r - P_3\| + \varepsilon_c - \rho_3 &= 0\\
\|P_r - P_4\| + \varepsilon_c - \rho_4 &= 0
\end{aligned}\right\} (2.3)$$

$$\left.\begin{aligned}
f_1(P_r, \varepsilon_c) &= \|P_r - P_1\| + \varepsilon_c - \rho_1 = 0\\
f_2(P_r, \varepsilon_c) &= \|P_r - P_2\| + \varepsilon_c - \rho_2 = 0\\
f_3(P_r, \varepsilon_c) &= \|P_r - P_3\| + \varepsilon_c - \rho_3 = 0\\
f_4(P_r, \varepsilon_c) &= \|P_r - P_4\| + \varepsilon_c - \rho_4 = 0
\end{aligned}\right\} (2.4)$$

$$\frac{\partial f_1}{\partial \varepsilon_c} = \frac{\partial}{\partial \varepsilon_c}\left(\|P_r - P_1\| + \varepsilon_c - \rho_1\right) = 1$$

$$\frac{\partial f_2}{\partial \varepsilon_c} = \frac{\partial}{\partial \varepsilon_c}\left(\|P_r - P_2\| + \varepsilon_c - \rho_2\right) = 1$$

$$\frac{\partial f_3}{\partial \varepsilon_c} = \frac{\partial}{\partial \varepsilon_c}\left(\|P_r - P_3\| + \varepsilon_c - \rho_3\right) = 1$$

$$\frac{\partial f_4}{\partial \varepsilon_c} = \frac{\partial}{\partial \varepsilon_c}\left(\|P_r - P_4\| + \varepsilon_c - \rho_4\right) = 1$$

$$\left.\begin{aligned}
f_1(P_r^0, \varepsilon_c^0) + \begin{bmatrix} \frac{(P_{xr}^0 - P_{x1})}{\|P_r^0 - P_1\|} & \frac{(P_{yr}^0 - P_{y1})}{\|P_r^0 - P_1\|} & \frac{(P_{zr}^0 - P_{z1})}{\|P_r^0 - P_1\|} & 1 \end{bmatrix} dP &= 0\\
f_2(P_r, \varepsilon_c^0) + \begin{bmatrix} \frac{(P_{xr}^0 - P_{x2})}{\|P_r^0 - P_2\|} & \frac{(P_{yr}^0 - P_{y2})}{\|P_r^0 - P_2\|} & \frac{(P_{zr}^0 - P_{z2})}{\|P_r^0 - P_2\|} & 1 \end{bmatrix} dP &= 0\\
f_3(P_r, \varepsilon_c) + \begin{bmatrix} \frac{(P_{xr}^0 - P_{x3})}{\|P_r^0 - P_3\|} & \frac{(P_{yr}^0 - P_{y3})}{\|P_r^0 - P_3\|} & \frac{(P_{zr}^0 - P_{z3})}{\|P_r^0 - P_3\|} & 1 \end{bmatrix} dP &= 0\\
f_4(P_r, \varepsilon_c) + \begin{bmatrix} \frac{(P_{xr}^0 - P_{x4})}{\|P_r^0 - P_4\|} & \frac{(P_{yr}^0 - P_{y4})}{\|P_r^0 - P_4\|} & \frac{(P_{zr}^0 - P_{z4})}{\|P_r^0 - P_4\|} & 1 \end{bmatrix} dP &= 0
\end{aligned}\right\} (2.5-$$

* ε_c is included in dP

$$\begin{bmatrix}
\frac{(P_{xr}^0 - P_{x1})}{\|P_r^0 - P_1\|} & \frac{(P_{yr}^0 - P_{y1})}{\|P_r^0 - P_1\|} & \frac{(P_{zr}^0 - P_{z1})}{\|P_r^0 - P_1\|} & 1\\
\frac{(P_{xr}^0 - P_{x2})}{\|P_r^0 - P_2\|} & \frac{(P_{yr}^0 - P_{y2})}{\|P_r^0 - P_2\|} & \frac{(P_{zr}^0 - P_{z2})}{\|P_r^0 - P_2\|} & 1\\
\frac{(P_{xr}^0 - P_{x3})}{\|P_r^0 - P_3\|} & \frac{(P_{yr}^0 - P_{y3})}{\|P_r^0 - P_3\|} & \frac{(P_{zr}^0 - P_{z3})}{\|P_r^0 - P_3\|} & 1\\
\frac{(P_{xr}^0 - P_{x4})}{\|P_r^0 - P_4\|} & \frac{(P_{yr}^0 - P_{y4})}{\|P_r^0 - P_3\|} & \frac{(P_{zr}^0 - P_{z4})}{\|P_r^0 - P_4\|} & 1
\end{bmatrix} dP = \begin{bmatrix}
\rho_1 - \varepsilon_c - \|P_r^0 - P_1\|\\
\rho_2 - \varepsilon_c - \|P_r^0 - P_2\|\\
\rho_3 - \varepsilon_c - \|P_r^0 - P_3\|\\
\rho_4 - \varepsilon_c - \|P_r^0 - P_4\|
\end{bmatrix}$$

• final equ

Iterative code:

1) Choose initial guess $\underline{p}_r^0$, ε_{c_0}
2) Choose ε_1, ε_2, and $n_{iter\text{-}max}$
3) set $n := 0$
4) while $(n < n_{iter\text{-}max})$ do
5) Set $\mathbb{1}_i = [(\underline{p}_r^N - \underline{p}_i)/\|\underline{p}_r^N - \underline{p}_i\|, 1]$ for $i = 1, 2, 3, 4$
6) solve the linear equation

$$\begin{bmatrix} \rho_1 - \varepsilon_c - \|\underline{p}_r^N - \underline{p}_1\| \\ \rho_2 - \varepsilon_c - \|\underline{p}_r^N - \underline{p}_2\| \\ \rho_3 - \varepsilon_c - \|\underline{p}_r^N - \underline{p}_3\| \\ \rho_4 - \varepsilon_c - \|\underline{p}_r^N - \underline{p}_4\| \end{bmatrix} = \begin{bmatrix} \mathbb{1}_1^T \\ \mathbb{1}_2^T \\ \mathbb{1}_3^T \\ \mathbb{1}_4^T \end{bmatrix} d\underline{p}$$

to obtain $d\underline{p}$

7) set $\underline{p}_r^{N+1} := \underline{p}_r^N + d\underline{p}(1:3, :)$
8) set $\varepsilon_c^{N+1} = \varepsilon_c^N + d\underline{p}(4, :)$
9) set $n := n+1$
10) if $\|d\underline{p}\| < \varepsilon_1$ or $\sqrt{\sum_{i=1}^{4} (\|\underline{p}_r^N - \underline{p}_i\| + \varepsilon_c - r_i)^2} < \varepsilon_2$ then
11) break
12) Return $\underline{p} = \underline{p}_r^N$
13) Return $\varepsilon_c = \varepsilon_c^N$

2b

```
function [receiver_position, epsilon_c] =
 trilat_clockbias(satellite_positions, measured_ranges, initial_guess,
 e1, e2, max_iter)
%TRILAT_CLOCKBIAS Trilateration for radio ranging

big_fat_one = @(p_r_guess, p_sat) (p_r_guess - p_sat) / norm(p_r_guess
 - p_sat);
receiver_position = initial_guess;
epsilon_c = 0;
```

```
for iter = 1:max_iter
    A = cell2mat(cellfun(@(p_sat) [big_fat_one(receiver_position,
 p_sat); 1]', num2cell(satellite_positions, 1), 'UniformOutput',
 false)');
    b = measured_ranges - ones(size(measured_ranges,1),1)*epsilon_c
 - cellfun(@norm, num2cell(bsxfun(@minus, receiver_position,
 satellite_positions), 1))';
    d_p = A \ b;
    receiver_position = receiver_position + d_p(1:end-1, :);
    epsilon_c = epsilon_c + d_p(end, :);
    if norm(d_p) < e1 || norm(b) < e2
        return
    end
end
warning('reached maximum iteration')
b
end


sat_positions = ...
    [90 -70 95;
    -90 -5 60;
    -75 -20 85;
    60 80 -70;
    80 90 100;
    35 -45 55;
    -80 65 20]'; % km

rho = ...
    [139.533;
    118.995;
    120.099;
    126.191;
    120.315;
    79.1232;
    110.215]; % km

[receiver_position, epsilon_c] = trilat_clockbias(sat_positions, rho,
 zeros(3,1), 1e-6, 1e-6, 1e3) % all in km

c = 299792458; %m/s

clock_bias = epsilon_c * 1e3 /c % sec


receiver_position =

   17.0000
   23.0000
   29.0000


epsilon_c =
```

```
    4.1298


clock_bias =

   1.3775e-05
```

Published with MATLAB® R2016a