UNIVERSITY OF MICHIGAN

AERO 584, Homework 7

Huckleberry Febbo
December 6, 2017

PROBLEM 5.1

Following the derivation for Pursuit Guidance from page 121-123 in the book Eqn. (5.19) can be extended to higher order derivatives as follows. First for this scheme, we first set $\theta = \beta$, next the established pattern for finite derivatives is studied:
For k = 1:

$$\gamma \leq 2 = \frac{1+1}{1}$$

For k = 2:

$$\gamma \leq \frac{3}{2} = \frac{2+1}{2}$$

Thus, for any k:

$$\gamma \leq \frac{k+1}{k}$$

PROBLEM 5.2

GIVEN:

$V_T = 1000 \frac{ft}{sec}$

$V_M = 3000 \frac{ft}{sec}$

$Y_T = 20{,}000$ ft ↳ CONSTANT

$\theta_T = 0$

$\beta(0) = 90°$

FIND:

(1) TRAJECTORIES OF TARGET AND MISSILE

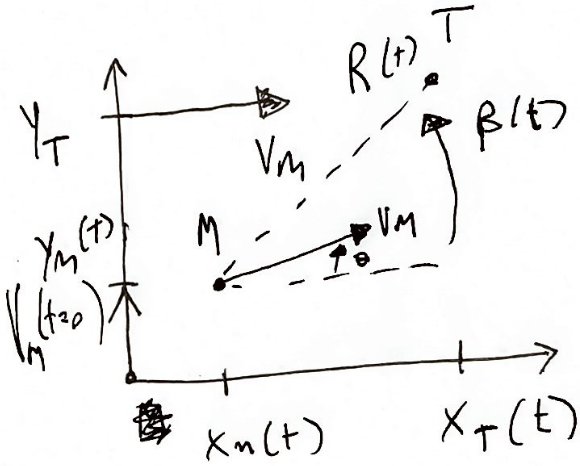
(2) TIME OF IMPACT

SOLUTION:

TYPO IN BOOK!

$$\dot{R} = V_T \cos\beta - V_M$$

$$\dot{\beta} = -\frac{V_T \sin\beta}{R}$$



THEN WE HAVE,

$$X_T(t) = \int_0^t V_T \, dt = V_T t$$

$$= 1000 \frac{ft}{sec} \, t$$

$Y_T = 20{,}000$ ft

FOR THE MISSILE, RELATIVE TO TARGET IN GLOBAL FRAME:

$$X_M(t) = X_T(t) - R(t)\cos(\beta(t))$$

$$Y_M(t) = Y_T - R(t)\sin(\beta(t))$$

// NEED TO FIND $R(t)$ & $\beta(t)$

↳ REST OF PROBLEM DONE ON COMPUTER!

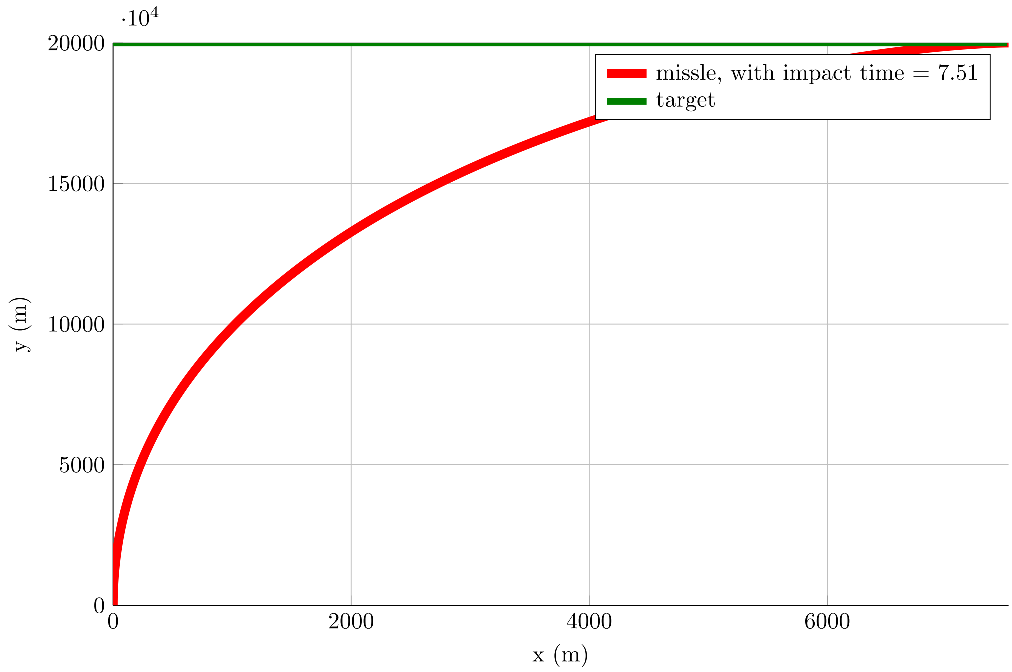


Figure 0.1

PROBLEM 5.3

NOTE: $\theta = \beta$ for pursuit guidance

As seen in Fig. 0.2 the trajectory is rate limited where it stops, which is at $t = 6.79$. If integration where to continue it would go behind the target and miss it. But, it cannot continue because the differential equations become unstable as $t \to t_f$ because $\dot{\beta} \to \infty$ as can be seen in Fig. 0.3. Which is is approximately[1] the time where we will go to our acceleration limit.

To find the miss distance, a new set of differential equations where solved (see code) where a state was added $\ddot{\beta}$ which was set to 40×32.2 [2] and then $\dot{\beta}$ was set to itself (now that it has been included as a state). Then integration was picked up where Fig. 0.2 left off. These trajectory results are shown in Fig. 0.4 where the missile missed the target given rate limit shown in Fig. 0.6. The miss distance is calculated as the minimum of R, because that is the closest that the missile got to the target, the result is shown in Fig. 0.5

[1] approximately because the differential equation given for $\ddot{\beta}$ is only good for $\beta << 1$ which is not the case
[2] the maximum turning acceleration

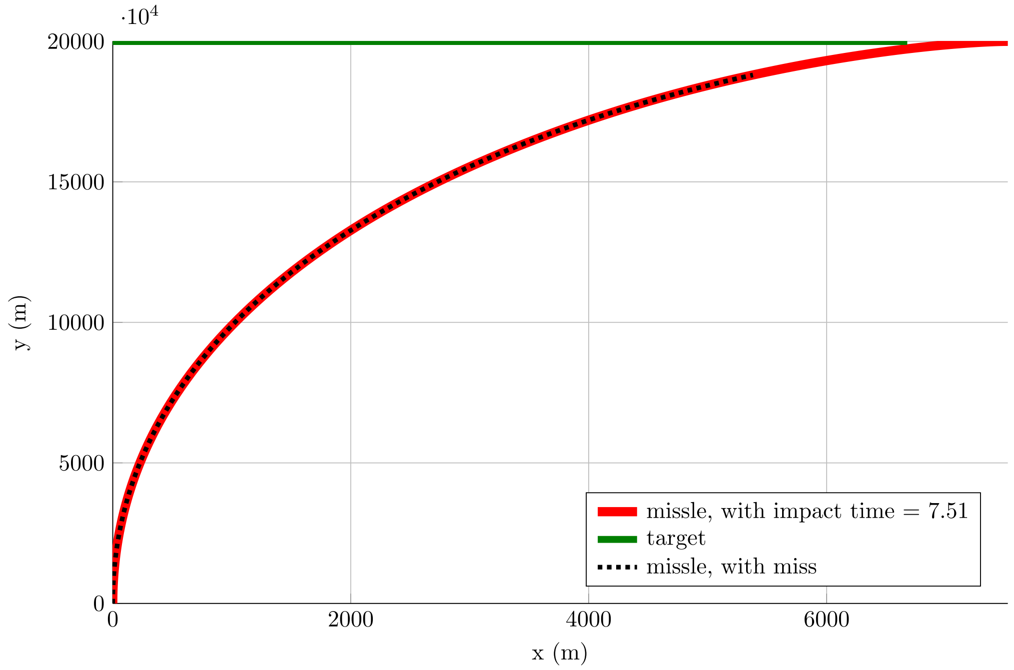


Figure 0.2

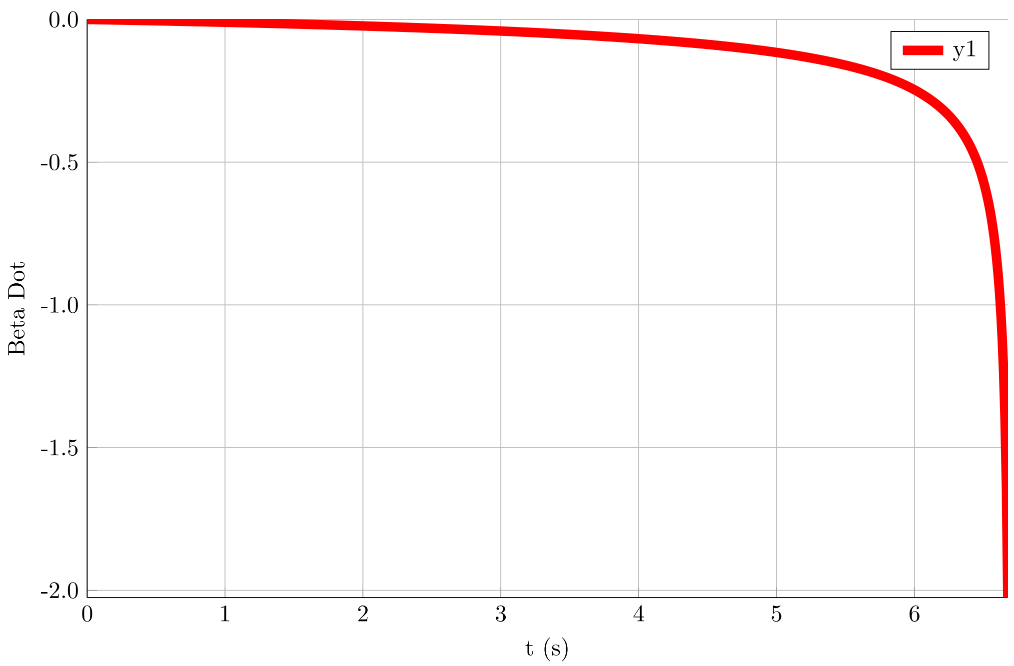


Figure 0.3

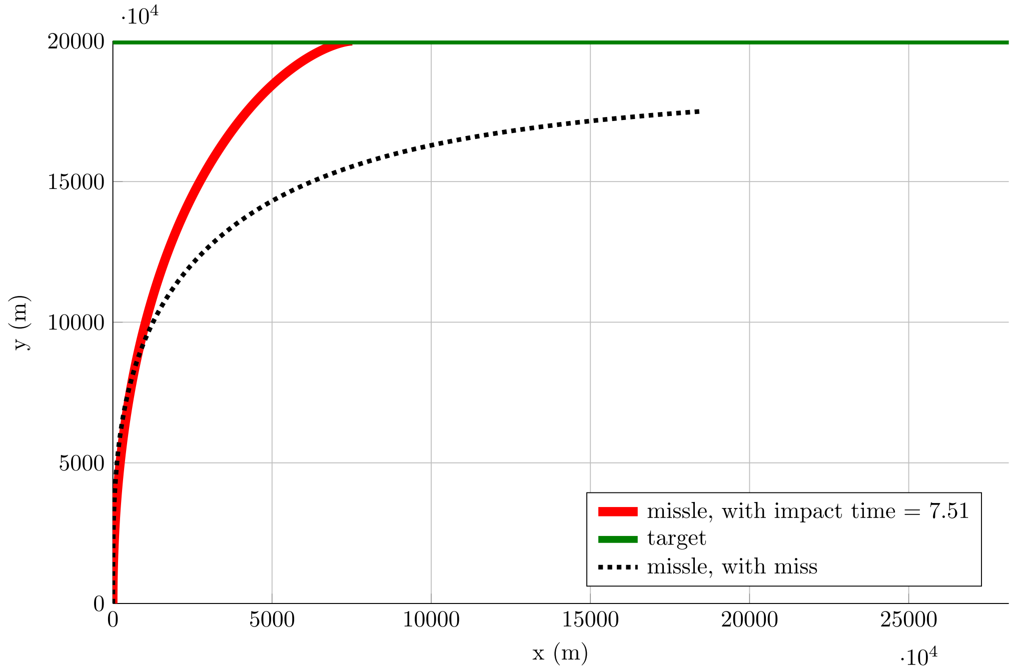


Figure 0.4

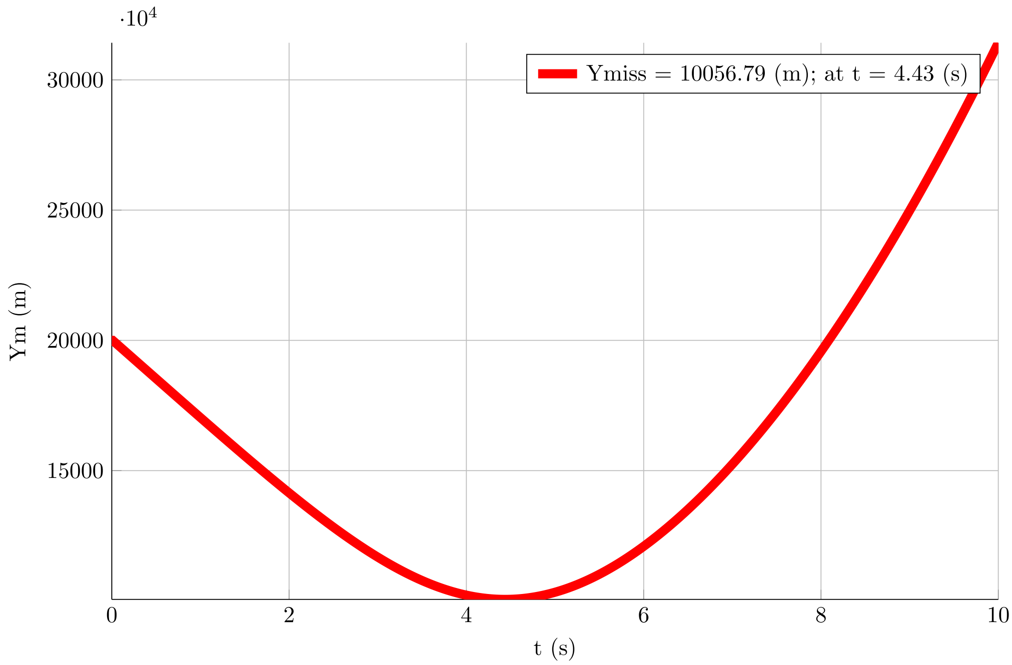


Figure 0.5

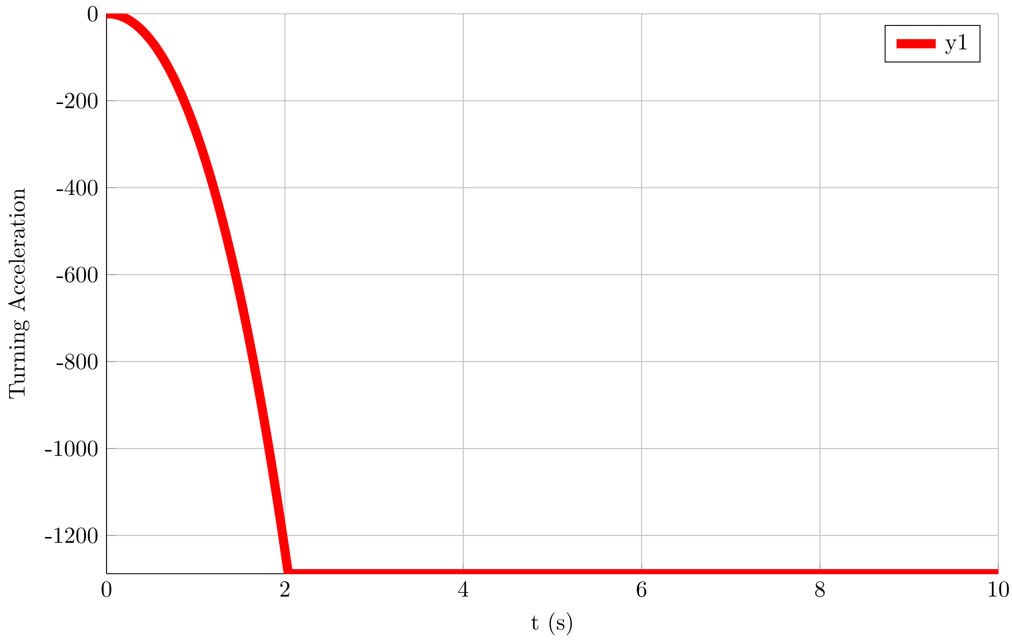


Figure 0.6: Missile hits acceleration limit at about 2 seconds. This does not match what was predicted in Fig. 0.3, but different integration schemes where used. In this investigation a simple one was used, see code

PROBLEM 5.4

For fixed lead guidance we have

$$\frac{V_m}{sin(\beta_0 - \theta_m)} = \frac{V_t}{sin(\beta_0 - \theta_t)}$$

where $\beta_0 = \frac{\pi}{2}$, $V_m = 1000\frac{ft}{s}$, $V_t = 1000\frac{ft}{s}$, and $R(0) = 2000ft$

For the missile to collide, it needs to be headed with the angle,

$$\theta_m = acos(\frac{1}{3}) = 1.23rad$$

The time to impact is shown in Fig. 0.7 as $tf = 7.08s$

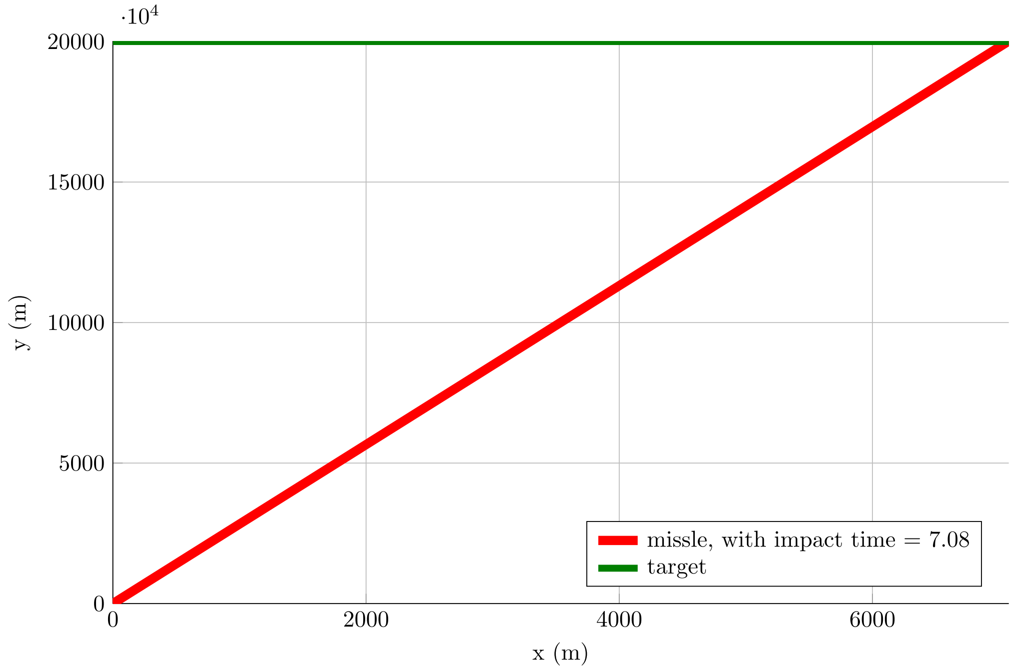
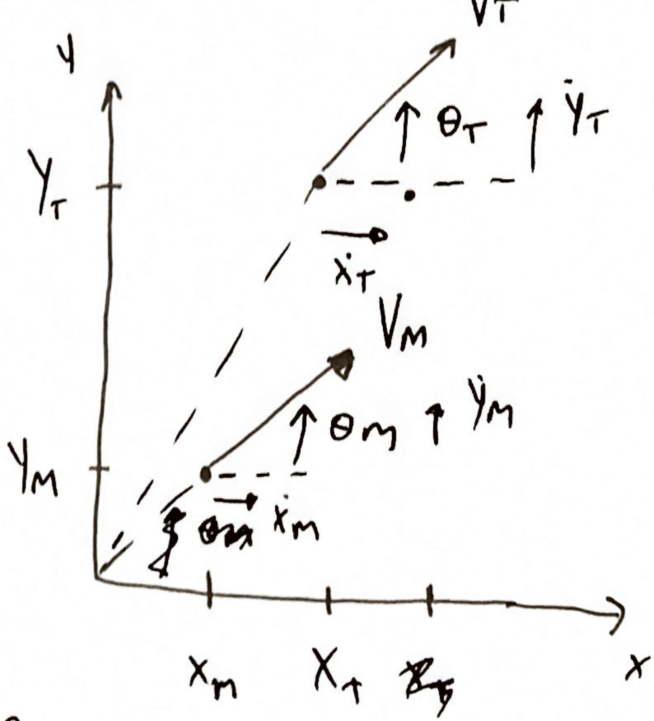


Figure 0.7

Problem # 5.7

Given:

$$X = X_T - X_M$$
$$Y = Y_T - Y_M$$



Find:

Differential Equations for x & y

Solution:

$$\dot{X}_T = V_T \cos(\theta_T) \qquad \dot{Y}_T = V_T \sin(\theta_T)$$

$$\dot{X}_M = V_M \cos(\theta_M) \qquad \dot{Y}_M = V_M \sin(\theta_M)$$

So, the solution is simply

$$\dot{x} = V_T \cos(\theta_T) - V_M \cos(\theta_M)$$
$$\dot{y} = V_T \sin(\theta_T) - V_M \sin(\theta_M)$$

PROBLEM # 5.8

GIVEN

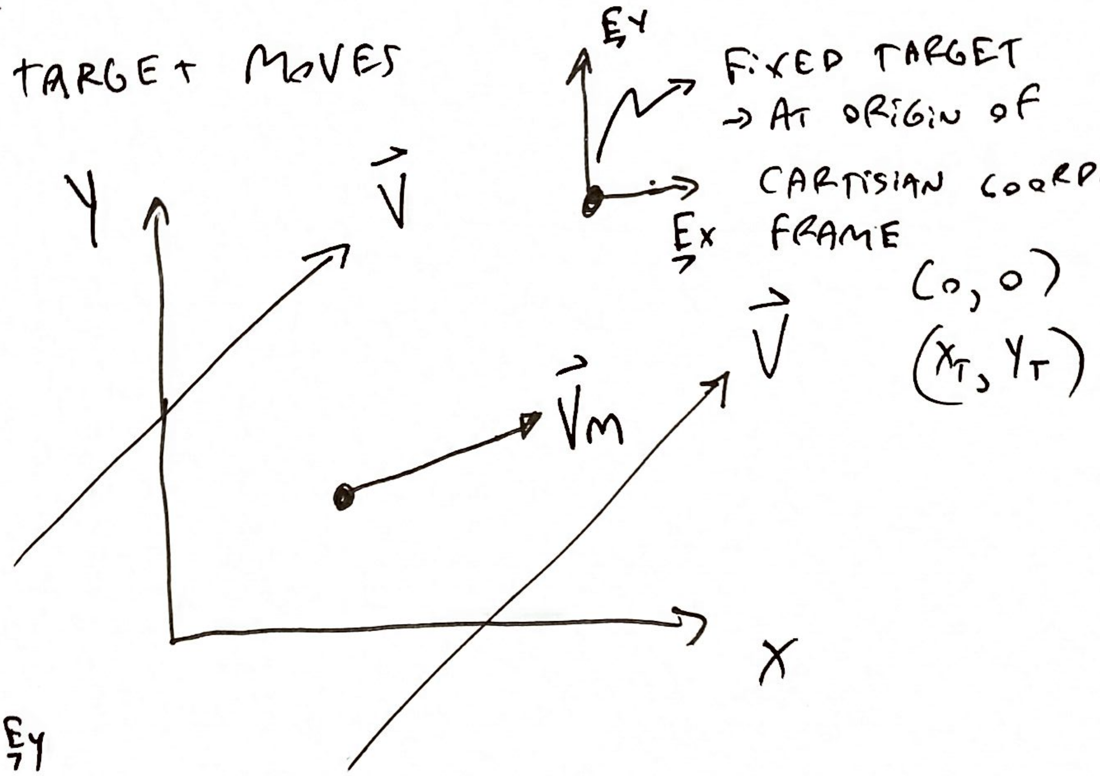
SAME PROBLEM AS 5.7 EXCEPT NOW THERE IS A CONSTANT VELOCITY FIELD & TARGET IS FIXED

FIND

SHOW THAT THE EFFECT OF CURRENTS IS THE SAME AS IF THE TARGET MOVES

FIXED TARGET → AT ORIGIN OF CARTISIAN COORDINAT FRAME

Solution:



$\vec{V} = V_x \underset{\sim}{E}_x + V_y \underset{\sim}{E}_y$

$\vec{V}_M = \dot{X}_M \underset{\sim}{E}_x + \dot{Y}_M \underset{\sim}{E}_y$

AS IN 5.7, WE SUBTRACT AS: $\vec{V}_M - \vec{V}$ AND look AT EACH COMPONENT:

$\dot{X} = \dot{X}_M - V_x = V_M \cos(\theta_M) - V_x$

$\dot{Y} = \dot{Y}_M - V_y = V_M \sin(\theta_M) - V_y$

PROBLEM # 5.9

GIVEN

$\lambda = 3$

$V_T = 1000 \dfrac{ft}{sec}$

$\beta(0) = 90°$

$V_M = 3000 \dfrac{ft}{sec}$

$R(0) = 2000\ ft$

$\theta(t) = 0.01\pi \sin(2\pi t)$

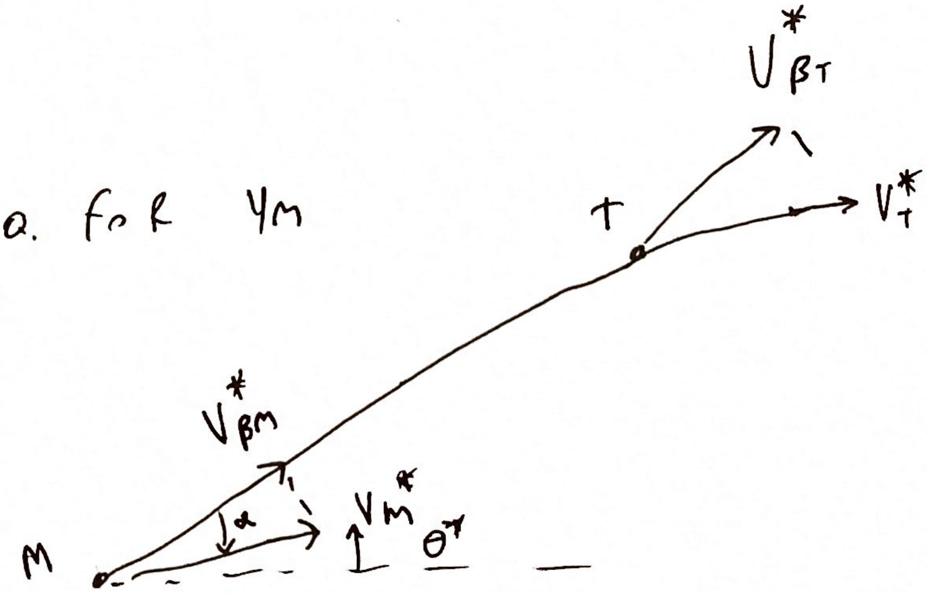
$\theta(0) = 0.4\pi\ RAD$

$$Y(s) = \frac{1}{1+0.5s}$$

FIND:

LINEARIZED EQ. FOR Y_M

SOLUTION:



Λ

$$V^*_{\beta M} = V^*_M \cos(\alpha)$$

$$D(s)\ \dot{Y}_M = \frac{V^*_{\beta M}\lambda}{\left|V^*_{\beta T} - V^*_{\beta M}\right|} N(s) \left(\frac{Y_T - Y_M}{t^*_f - t}\right)$$

WHERE,

$$t_f^* = \frac{R_0}{\left| V_{\beta_T}^* - V_{\beta_M}^* \right|}$$

CODE FOR 5.2 AND 5.3

```
using Plots
pgfplots()
using LaTeXStrings
PGFPlots.pushPGFPlotsPreamble("\\usepackage{amssymb}")
using OrdinaryDiffEq
using DiffEqBase
#using ParameterizedFunctions
using DiffEqCallbacks

TT = linspace(0,10,1000)
const Vt = 1000;
const Vm = 3000;

# differential equations
f = (t,x,dx) -> begin

    # diff eqs.
    dx[1] = Vt*cos(x[2])-Vm;        # 1. R
    dx[2] = -Vt*sin(x[2])/x[1];     # 2. B
end

x0 = [20000;pi/2]
tspan = (TT[1],TT[end])
prob = ODEProblem(f,x0,tspan)
sol = DiffEqBase.solve(prob,Tsit5())

# extract results
x1 = [sol(t)[1] for t in TT]
x2 = [sol(t)[2] for t in TT]

Xm = zeros(length(TT),1); Ym = zeros(length(TT),1);
Xt = zeros(length(TT),1); Yt = 20000*ones(length(TT),1);
tf=0;num=0;
for i in 1:length(TT)
    Xt[i] = Vt*TT[i]
    Xm[i] = Xt[i] - x1[i]*cos(x2[i])
    Ym[i] = 20000 - x1[i]*sin(x2[i])
    if Ym[i] > 20000
        tf = TT[i]
        num = i
        break
```

```
    end
end

# misc variables
l1 = (4,:red,:solid)
l2 = (3,:green,:solid)
l3 = (2,:black,:dot)

s1 = string("missle, with impact time = ",round(tf,2))
s2 = "target"

# position
p1 = plot(Xm[1:num],Ym[1:num],line=l1,label=s1)
plot!(Xt[1:num],Yt[1:num],line=l2,label=s2)
yaxis!("y (m)")
xaxis!("x (m)")
savefig(string("figs/p2",".",:svg));

# prob 5.3, amy_max = 40*32.2
# since theta = beta for pursuit guidance
c = 670
c2 = c-1
c3=1000
const g = Vm/Vt
const tf_g = TT[c]
const b0 = pi/2
const TM = 40*32.2
# differential equations
f = (t,x,dx) -> begin

    # diff eqs.
    dx[1] = Vt*cos(x[2])-Vm;        # 1. R
    dx[2] = -Vt*sin(x[2])/x[1];     # 2. B - > dx[3] does not work here because B is not
    dx[3] = b0*(2-g)/((g-1)^2*(tf_g)^2)*((tf_g-t)/(tf_g))^((3-2*g)/(g-1))
# unstable because Bdot goes to infinity as t goes to tf
end

x0 = [20000;pi/2;0]
tspan = (TT[1],TT[c2])
prob = ODEProblem(f,x0,tspan)
sol = DiffEqBase.solve(prob,Tsit5())

# extract results
ff=indmin(abs.(sol.t[end] - TT))-1
```

```
x1 = [sol(t)[1] for t in TT[1:ff]]
x2 = [sol(t)[2] for t in TT[1:ff]]
x3 = [sol(t)[3] for t in TT[1:ff]]

Xml = zeros(length(TT),1); Yml = zeros(length(TT),1);
Xt = zeros(length(TT),1);
tf2 = 0;num2=0;
for i in 1:ff
    Xt[i] = Vt*TT[i]
    Xml[i] = Xt[i] - x1[i]*cos(x2[i])
    Yml[i] = 20000 - x1[i]*sin(x2[i])
    tf2 = TT[i]
    num2 = i
    if Yml[i] > 20000
        break
    end
end

s1 = string("missle, with impact time = ",round(tf,2))
s2 = "target"
s3 = string("missle, with miss")

# position
p1 = plot(Xm[1:num],Ym[1:num],line=l1,label=s1)
plot!(Xt[1:num],Yt[1:num],line=l2,label=s2)
plot!(Xml[1:num2],Yml[1:num2],line=l3,label=s3,legend=:bottomright)
yaxis!("y (m)")
xaxis!("x (m)")
savefig(string("figs/p3",".",:svg));

# turning rate
p1 = plot(TT[1:ff],x3[1:ff],line=l1)
yaxis!("Beta Dot")
xaxis!("t (s)")
savefig(string("figs/p3b",".",:svg));


####################
## finding Ym
Xml = zeros(length(TT),1); Yml = zeros(length(TT),1);
Xt = zeros(length(TT),1);
x1 = zeros(length(TT),1);
x2 = zeros(length(TT),1);
```

```
x3 = zeros(length(TT),1);
x4 = zeros(length(TT),1);

# inital conditions
Yt[1] = 20000
x1[1] = 20000
x2[1] = pi/2
for i in 1:length(TT)-1
    dt = TT[i+1]-TT[i]
    Xt[i+1] = Xt[i] + Vt*TT[i]*dt
    Xml[i+1] = Xml[i] + Vm*cos(x2[i])*dt
    Yml[i+1] = Yml[i] + Vm*sin(x2[i])*dt
    x1[i+1] = sqrt((Yt[i]-Yml[i])^2 + (Xt[i]-Xml[i])^2)
    x2[i+1] = atan2(Yt[i]-Yml[i],Xt[i]-Xml[i]) # beta
    x3[i+1] = x3[i] + (x2[i+1] - x2[i])/dt # beta dot
    x4[i+1] = x4[i] + (x3[i+1] - x3[i])/dt # beta double dot
    if x4[i+1] > TM
        x4[i+1] = TM
    elseif x4[i+1] < -TM
        x4[i+1] = -TM
    end
end

s1 = string("missle, with impact time = ",round(tf,2))
s2 = "target"
s3 = string("missle, with miss")

# position
num2 = 1000
p1 = plot(Xm[1:num],Ym[1:num],line=l1,label=s1)
plot!(Xt[1:num],Yt[1:num],line=l2,label=s2)
plot!(Xml[1:num2],Yml[1:num2],line=l3,label=s3,legend=:bottomright)
yaxis!("y (m)")
xaxis!("x (m)")
savefig(string("figs/p3c",".",:svg));

# miss distance
Ym=x1[indmin(abs.(x1))]
tm=TT[indmin(abs.(x1))]
p1 = plot(TT,x1,line=l1,label=string("Ymiss = ",round(Ym,2)," (m); at t = ",round(tm,2),
yaxis!("Ym (m)")
xaxis!("t (s)")
savefig(string("figs/p3d",".",:svg));
```

```
# miss distance
p1 = plot(TT,x4,line=l1)
yaxis!("Turning Acceleration")
xaxis!("t (s)")
savefig(string("figs/p3e",".",:svg));
```

CODE

CODE