 doi:10.20944/preprints202208.0546.v1

Feasible Control Strategies and Feasible System Response for the Navigation of an Autonomous Ship Robot

MASIALA MAVUNGU
[1]University of Johannesburg, Department of Mechanical Engineering Science, South Africa
Corresponding author: First A. Author: MASIALA MAVUNGU.
"This work was supported by the University of Johannesburg"

Abstract

This paper develops a methodology to control the navigation of an autonomous ship robot from an initial state to a final. To solve the problem the following approach is used: The ship robot system is modelled as a control system of six ordinary differential equations involving six state variables and three control variables. After having computed the system Hamiltonian, the feasible controls for optimality are derived from the normal equations of optimality as functions of the state and costate variables. Such controls are substituted into the state and the costate equations and give a combined control-free state-costate system of ordinary differential equations which is solved numerically by using a developed Matlab computer program. Associated computational simulations are designed and provided to show the reliability of the approach.

Key words: Autonomous Ship Robot, Autonomous System, Autonomous Vehicle, Control System, Differential Equations, Intelligent Systems, Mobile Systems, Optimal Control, Nonholonomic Constraint.

1-Introduction

Nowadays, some under water activities like fuel explorations are dangerous for humans to perform since there exist underwater areas which are harmful. For those activities, underwater vehicle robot are diligently and relevantly needed.

The transportation of huge and heavy quantities of goods from a country to another by airplanes are very costly and cumbersome so that one needs to rely on affordable and easy ways. To minimize costs and maximize utility for the customers, the transportation of huge quantities of goods from a country to another is suitably performed by means of ships. Sometimes ships cannot meet all the assigned requirements and expectations. Sometimes they are involved in accidents due to humans limited controls to detect, prevent and avoid adroitly accidents. Sometimes accidents are due to lack of control systems which can detect defaults existing in the system and then predict bad events.

To overcome all the issues, autonomous ship robots are emerging as relevant and efficient ways for transportation of goods.

An autonomous ship is a ship that moves autonomously. Unlike the autonomous cars, the autonomous ship is not subject to nonholonomic wheels constraints.

As a dynamical system, it's subject to equilibrium considerations. The motion of an autonomous ship

 doi:10.20944/preprints202208.0546.v1

from an initial state to a final state needs to be planned so that for a given initial condition the trajectory and all the other motion's characteristics can be predicted in a rigorous manner for secure and reliable decision making. Or for a prescribed safe trajectory, a feedback control can be designed for the ship to follow asymptotically the prescribed trajectory. Even when the ship is not following properly the prescribed trajectory, some feedback control function strategies can take it from outside to follow asymptotically the prescribed trajectory. Modelling and control of a ship are relevant tasks to perform.
To study the dynamics of his motion, one needs to judiciously model it so that for a given initial state all the characteristics of his motion can be rigorously and precisely determined. There exist many techniques for modelling and controlling the motion of a ship. One of the way consists of developing a system of ordinary differential equations to describe the dynamics of the ship.
There exist many works carried out on autonomouis ships. The reader is referred to [1]-[21].
This paper is organized in the following manner: Section 2 is about developing mathematical models which include the kinematic model of the ship and the objective functional cost to minimize.

Section 3 is about computing the system Hamiltonian, deriving the normal equations for optimality and the feasible controls. Section 4 is about stating the Pontryagin's Minimum Principle. Section 5 is about designing computational simulations.

2-Mathematical Models and Problem Formulation

The kinematic model of an autonomous ship robot is given in [22] as follows:

$$\frac{dx}{dt}(t) = u(t)\cos\big(\theta(t)\big) - v(t)\sin(\theta(t)) \qquad (1)$$

$$\frac{dy}{dt}(t) = u(t)\sin\big(\theta(t)\big) + v(t)\cos(\theta(t)) \qquad (2)$$

$$\frac{d\theta}{dt}(t) = \omega(t) \qquad (3)$$

$(x(t), y(t))$ is the coordinates of the ship in the XY horizontal plane at time t, $u(t)$ is the lateral velocity of the ship at time t, $v(t)$ is the transversal velocity of the ship at time t, $\theta(t)$ is the heading angle of the ship at time t, $\omega(t)$ is the angular velocity at time t. Let's extend the kinematic model by incorporating some useful elements.

Define $u_1(t) = u(t)$, $u_2(t) = v(t)$ and $u_3(t) = \omega(t)$ and define $\omega_1(t)$, $\omega_2(t)$ and $\omega_3(t)$ as reference commands regulating respectively the behaviour of $u_1(t)$, $u_2(t)$ and $u_3(t)$. Define the reference commands as functions satisfying the following closed-loop system of ordinary differential equations:

$$\frac{du_1}{dt}(t) = -c_1u_1(t)+c_1\omega_1(t) \qquad (4)$$

$$\frac{du_2}{dt}(t) = -c_2u_2(t)+c_2\omega_2(t) \qquad (5)$$

$$\frac{du_3}{dt}(t) = -c_3u_3(t)+c_3\omega_3(t) \qquad (6)$$

The objective functional to minimize is as follows:

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 31 August 2022 doi:10.20944/preprints202208.0546.v1

$$J(\omega_1,\omega_2,\omega_3)=\int_{t_0}^{t_f}({\omega_1}^2+{\omega_2}^2+{\omega_3}^2)dt \tag{7}$$

The whole kinematic system is as follows:

$$\frac{dx}{dt}(t)=u_1(t)\cos\big(\theta(t)\big)-u_2(t)\sin(\theta(t)) \tag{8}$$

$$\frac{dy}{dt}(t)=u_1(t)\sin\big(\theta(t)\big)+u_2(t)\cos(\theta(t)) \tag{9}$$

$$\frac{d\theta}{dt}(t)=u_3(t) \tag{10}$$

$$\frac{du_1}{dt}(t)=-c_1u_1(t)+c_1\omega_1(t) \tag{11}$$

$$\frac{du_2}{dt}(t)=-c_2u_2(t)+c_2\omega_2(t) \tag{12}$$

$$\frac{du_3}{dt}(t)=-c_3u_3(t)+c_3\omega_3(t) \tag{13}$$

The problem is to compute optimal controls $({\omega_1}^*,{\omega_2}^*,{\omega_3}^*)$ to drive the autonomous ship robot represented by the system (8)-(13) from an initial state to a final state such that the total energy (7) is minimized.

3- Hamiltonian, Normal Equations and Feasible Controls

The system Hamiltonian as $H=H(t,x,y,\theta,u_1,u_2,u_3,\omega_1,\omega_2,\omega_3,\alpha_1,\alpha_2,\alpha_3,\alpha_4,\alpha_5,\alpha_6)$ is given by

$$H=h(\omega_1,\omega_2,\omega_3)+\sum_{k=1}^{6}\alpha_k(t)f_k(t,x,y,\theta,u_1,u_2,u_3,\omega_1,\omega_2,\omega_3) \tag{14}$$ where

$$h(\omega_1,\omega_2,\omega_3)={\omega_1}^2+{\omega_2}^2+{\omega_3}^2 \tag{15}$$

$$f_1(t,x,y,\theta,u_1,u_2,u_3,\omega_1,\omega_2,\omega_3)=u_1(t)\cos\big(\theta(t)\big)-u_2(t)\sin(\theta(t)) \tag{16}$$

$$f_2(t,x,y,\theta,u_1,u_2,u_3,\omega_1,\omega_2,\omega_3)=u_1(t)\sin\big(\theta(t)\big)+u_2(t)\cos(\theta(t)) \tag{17}$$

$$f_3(t,x,y,\theta,u_1,u_2,u_3,\omega_1,\omega_2,\omega_3)=u_3(t) \tag{18}$$

$$f_4(t,x,y,\theta,u_1,u_2,u_3,\omega_1,\omega_2,\omega_3)=-c_1u_1(t)+c_1\omega_1(t) \tag{19}$$

$$f_5(t,x,y,\theta,u_1,u_2,u_3,\omega_1,\omega_2,\omega_3)=-c_2u_2(t)+c_2\omega_2(t) \tag{20}$$

$$f_6(t,x,y,\theta,u_1,u_2,u_3,\omega_1,\omega_2,\omega_3)=-c_3u_3(t)+c_3\omega_3(t) \tag{21}$$

$\boldsymbol{\alpha}(t)=[\alpha_1(t),\alpha_2(t),\alpha_3(t),\alpha_4(t),\alpha_5(t),\alpha_6(t)]$ is the vector of costate variables,

The feasible controls are solutions to normal equations which are as follows:

$$\frac{\partial H}{\partial\omega_1}=2\omega_1+c_1\alpha_4=0 \tag{22}$$

$$\frac{\partial H}{\partial\omega_2}=2\omega_2+c_2\alpha_5=0 \tag{23}$$

$$\frac{\partial H}{\partial\omega_3}=2\omega_3+c_3\alpha_6=0 \tag{24}$$

The associated feasible controls are as follows:

$$\omega_1{}^* = -0.5c_1\alpha_4{}^* \quad (25)$$

$$\omega_2{}^* = -0.5c_2\alpha_5{}^* \quad (26)$$

$$\omega_3{}^* = -0.5c_3\alpha_6{}^* \quad (27)$$

4- Pontryagin's Minimum Principle

If $\omega_1{}^*, \omega_2{}^*$ and $\omega_3{}^*$ are the optimal control and $x^*, y^*, \theta^*,\ u_1{}^*, u_2{}^*$ and $u_3{}^*$ are the associated optimal states, then there exist a costate vector
$\boldsymbol{\alpha}^*(t) = [\alpha_1{}^*(t), \alpha_2{}^*(t), \alpha_3{}^*(t), \alpha_4{}^*(t), \alpha_5{}^*(t), \alpha_6{}^*(t)] \in\ \mathbb{R}^5$ such that

$$\frac{dx^*}{dt}(t) = u_1{}^*(t)\cos\big(\theta^*(t)\big) - u_2{}^*(t)\sin(\theta^*(t)) \quad (28)$$

$$\frac{dy^*}{dt}(t) = u_1{}^*(t)\sin\big(\ \theta^*(t)\big) + u_2{}^*(t)\cos(\ \theta^*(t)) \quad (29)$$

$$\frac{d\theta^*}{dt}(t) = u_3{}^*(t) \quad (30)$$

$$\frac{du_1{}^*}{dt}(t) = -c_1u_1{}^*(t){+}c_1\omega_1{}^*(t) \quad (31)$$

$$\frac{du_2{}^*}{dt}(t) = -c_2u_2{}^*(t){+}c_2\omega_2{}^*(t) \quad (32)$$

$$\frac{du_3{}^*}{dt}(t) = -c_3u_3{}^*(t){+}c_3\omega_3{}^*(t) \quad (33)$$

$$\frac{d\alpha_1{}^*}{dt}(t) = 0 \quad (34)$$

$$\frac{d\alpha_2{}^*}{dt}(t) = 0 \quad (35)$$

$$\frac{d\alpha_3{}^*}{dt}(t) = (\alpha_1{}^*(t)\sin\big(\theta^*(t)\big) - \alpha_2{}^*(t)\cos\big(\theta^*(t)\big))u_1{}^* - (\alpha_1{}^*(t)\cos\big(\theta^*(t)\big) - \alpha_2{}^*(t)\sin\big(\theta^*(t)\big))u_2{}^* \quad (36)$$

$$\frac{d\alpha_4{}^*}{dt}(t) = -(\alpha_1{}^*(t)\cos\big(\theta^*(t)\big) + \alpha_2{}^*(t)\sin\big(\theta^*(t)\big) - c_1\alpha_4{}^*(t)) \quad (36)$$

$$\frac{d\alpha_5{}^*}{dt}(t) = -(\alpha_2{}^*(t)\cos\big(\theta^*(t)\big) - \alpha_1{}^*(t)\sin\big(\theta^*(t)\big) - c_2\alpha_5{}^*(t)) \quad (37)$$

$$\frac{d\alpha_6{}^*}{dt}(t) = -(\alpha_3{}^*(t) - c_3\alpha_6{}^*(t)) \quad (38)$$

$J(\omega_1{}^*, \omega_2{}^*, \omega_3{}^*) \le J(\omega_1, \omega_2, \omega_3)$ for each $(\omega_1, \omega_2, \omega_3) \in D \subseteq \mathbb{R}^3$ (39)

Where $\omega_1{}^*$, $\omega_2{}^*$ and $\omega_3{}^*$ are as above defined.

5- Vectorization of the problem and Computational simulations

Define $\boldsymbol{Y}^*(t) = [x^*, y^*, \theta^*, u_1{}^*, u_2{}^*, u_3{}^*]$, $\boldsymbol{A}(t) = [f_1(t), f_2(t), f_3(t), f_4(t), f_5(t), f_6(t)]$ and $\boldsymbol{z}^*(t) = [\boldsymbol{Y}^*(t), \boldsymbol{\alpha}^*(t)]$, then the Pontryagin's Minimum Principle can be rewritten as:

 doi:10.20944/preprints202208.0546.v1

$$\frac{dz_1{}^*}{dt}(t) = z_4{}^*(t)\cos\big(z_3{}^*(t)\big) - z_5{}^*(t)\sin(z_3{}^*(t)) \tag{40}$$

$$\frac{dz_2{}^*}{dt}(t) = z_4{}^*(t)\sin\big(\,z_3{}^*(t)\big) + z_5{}^*(t)\cos(\,z_3{}^*(t)) \tag{41}$$

$$\frac{dz_3{}^*}{dt}(t) = z_6{}^*(t) \tag{42}$$

$$\frac{dz_4{}^*}{dt}(t) = -c_1 z_4{}^*(t) + c_1\omega_1{}^*(t) \tag{43}$$

$$\frac{dz_5{}^*}{dt}(t) = -c_2 z_5{}^*(t) + c_2\omega_2{}^*(t) \tag{44}$$

$$\frac{dz_6{}^*}{dt}(t) = -c_3 z_6{}^*(t) + c_3\omega_3{}^*(t) \tag{45}$$

$$\frac{dz_7{}^*}{dt}(t) = 0 \tag{46}$$

$$\frac{dz_8{}^*}{dt}(t) = 0 \tag{47}$$

$$\frac{dz_9{}^*}{dt}(t) = (z_7{}^*(t)\sin\big(z_3{}^*(t)\big) - z_8{}^*(t)\cos\big(z_3{}^*(t)\big))z_4{}^* - (z_7{}^*(t)\cos\big(z_3{}^*(t)\big) - z_8{}^*(t)\sin\big(z_3{}^*(t)\big))z_5{}^* \tag{48}$$

$$\frac{dz_{10}{}^*}{dt}(t) = -(z_7{}^*(t)\cos\big(z_3{}^*(t)\big) + z_8{}^*(t)\sin\big(z_3{}^*(t)\big) - c_1 z_{10}{}^*(t)) \tag{49}$$

$$\frac{dz_{11}{}^*}{dt}(t) = -(z_8{}^*(t)\cos\big(z_3{}^*(t)\big) - z_7{}^*(t)\sin\big(z_3{}^*(t)\big) - c_2 z_{11}{}^*(t)) \tag{50}$$

$$\frac{dz_{12}{}^*}{dt}(t) = -(z_9{}^*(t) - c_3 z_{12}{}^*(t)) \tag{51}$$

Where $\omega_1{}^* = -0.5c_1 z_{10}{}^*$, $\omega_2{}^* = -0.5c_2 z_{11}{}^*$, $\omega_3{}^* = -0.5c_3 z_{12}{}^*$ (52)

Which is generally

$$\frac{d\boldsymbol{Z}^*}{dt}(t) = \boldsymbol{F}(t,\boldsymbol{Z}^*) =,\ \boldsymbol{Z}(0) = \boldsymbol{Z_0} \in \mathbb{R}^{\mathbf{12}} \tag{53}$$

The system of ordinary differential equations (40)-(52) obtained by combining the state and the costate systems of ordinary differential equations. The vector function $f(t,\boldsymbol{Z}^*)$ in (53) is continuous because each of its components is continuous. The partial derivative of each component of $f(t,\boldsymbol{Z}^*)$ with respect to each component of $\boldsymbol{Z}^*$ is continuous. Thus, the initial value problem (53) admit unique solution. The components of function $f(t,\boldsymbol{Z}^*)$ are the right hand sides of the above system (40)-(51). To solve the problem, computer programs are developed in Matlab. The first program denoted dzdt=ship2(t,z) codes the combined nonlinear state and costate system of ordinary differential equations (40)-(51) together with expressions (52). The second program denoted

[t,z] = runge(fs,t0,tf,N,z0), codes a fourth-order Runge-Kutta numerical method. The third program denoted main_ship2, as the main program, contain script codes to invoke runge(fs,t0,tf,N,z0) to solve the system of ordinary differential equations coded by ship2(t,z).

Notice that each statement preceded by % has no effect on the computer program. It is just a comment for the reader who is not familiar with Matlab codes to know what the current statement code is doing.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 31 August 2022 doi:10.20944/preprints202208.0546.v1

The following set of codes written in Matlab concerns the nonlinear system of differential equations into a function called dzdt= ship2 (t,z)The coded using Matlab function as follows:

```
function dzdt= ship2(t,z)

dzdt = zeros(12,1);

c1=1; c2=1; c3=1;

dzdt(1) = z(4)*cos(z(3)) - z(5)*sin(z(3));

dzdt(2) = z(4)*cos(z(3)) + z(5)*cos(z(3));

dzdt(3) = z(6);

dzdt(4) = -c1*z(4) + c1*(-0.5*c1*z(10));

dzdt(5) = -c2*z(5) + c2*(-0.5*c2*z(11));

dzdt(6) = -c3*z(6) + c3*(-0.5*c3*z(12));

dzdt(7) = 0;

dzdt(8) = 0;

dzdt(9) = (z(7)*z(4) + z(8)*z(5))*sin(z(3)) + (z(7)*z(5) -
z(8)*z(4))*cos(z(3));

dzdt(10) = -(z(7)*cos(z(3)) + z(8)*sin(z(3)) - c1*z(10));

dzdt(11) = -(-z(7)*sin(z(3)) + z(8)*cos(z(3)) - c2*z(11));

dzdt(12) = -(z(9) - c3*z(12));
```

Such a system is solved by calling the following Matlab function

```
function [t,z] = runge(fs,t0,tf,N,z0)

%The related m-file is runge.m

h=(tf-t0)./(N-1);%The step size

s=t0:h:tf;%subdivision of the time interval into discrete subintervals

t=s';% column vector t

%z0 is the initial vector solution

%Let start solving the state system

z = zeros(N,length(z0));

z(1,:) = z0.'; %each mesh point of the solution is a row vector
```

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 31 August 2022 doi:10.20944/preprints202208.0546.v1

```
%state is the name of the function defining the state system of
equations,

for n = 2:N

k1 = feval(fs,t(n-1),z(n-1,:));

k2 = feval(fs,t(n-1)+(h/2),z(n-1,:)+(h/2)*k1');

k3 = feval(fs,t(n-1)+(h/2),z(n-1,:)+(h/2)*k2');

k4 = feval(fs,t(n-1)+h,z(n-1,:)+h*k3');

z(n,:) = z(n-1,:)+(h/6)*(k1'+2*k2'+2*k3'+k4');

end
```

The above Matlab function `function [t,z] = runge(fs,t0,tf,N,z0)` is invoked through the main function (written in Matlab) which constitutes the main function of the program coded as follows:

```
function main_ship2

clear all

clc

format short

c1=1; c2=1; c3=1;

t0=0;% t0 is the initial time of the motion;
tf=5;% tf is the final time of the motion;
N=501; %N is the number of discrete point;
p=12;  %Number of state variables (6) and costate variables (6).
h=(tf-t0)/(N-1);% Step size.
t=t0:h:tf;%Vector of discrete times
% z0=[zeros(6,1);ones(6,1)]; A trial initial condition.

% z0=[0;0;pi/4;0;0;0;ones(6,1)]; A trial initial condition.

z=zeros(N,p); % Initialization of z

[t,z]=runge('ship2',t0,tf,N,z0);
dx = z(:,4).*cos(z(:,3)) - z(:,5).*sin(z(:,3));%Velocity along x axis
dy = z(:,4).*sin(z(:,3)) + z(:,5).*cos(z(:,3));%Velocity along y axis
dTheta = z(:,6); % heading angular velocity
d2x=(-c1*z(:,4)-0.5*c1*c1*z(:,10)-z(:,5).*z(:,6)).*cos(z(:,3)) +
(c2*z(:,5)+0.5*c2*c2*z(:,11)-z(:,4).*z(:,6)).*sin(z(:,3));
d2y=(-c1*z(:,4)-0.5*c1*c1*z(:,10)-z(:,5).*z(:,6)).*sin(z(:,3)) + (-
c2*z(:,5)-0.5*c2*c2*z(:,11)+z(:,4).*z(:,6)).*cos(z(:,3));
speed=(dx.^2+dy.^2).^0.5;
Acceleration=(d2x.^2 + d2y.^2).^0.5;
disp('To plot the feasible trajectory of the ship robot, press a key');
pause;
plot(z(:,1),z(:,2),'r'); xlabel('x '); ylabel('y ');
```

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 31 August 2022 doi:10.20944/preprints202208.0546.v1

```
print C:\Users\Guest\Documents\4july2022\Robot_Ship_Trajectory.png
clc
disp('To plot the feasible speed of the ship robot, press a key');
pause;
plot(t,speed,'r'); xlabel('t'); ylabel('speed ');
print C:\Users\Guest\Documents\4july2022\Robot_Ship_Speed.png
clc
plot(t,Acceleration,'r'); xlabel('t'); ylabel('Acceleration');
print C:\Users\Guest\Documents\4july2022\Robot_Ship_Acceleration.png
disp('To plot the feasible state functions, press a key');
pause;
for k=1:6
  subplot(3,2,k); plot(t,z(:,k),'r');
  xlabel('Time t '); ylabel('State ');
end
print C:\Users\Guest\Documents\4july2022\Robot_Ship_States.png
clc
disp('To plot the feasible costate functions, press a key');
pause;
for k=1:6
  subplot(3,2,k); plot(t,z(:,6+k),'r');
  xlabel('Time t '); ylabel('Costate');
end
print C:\Users\Guest\Documents\4july2022\Robot_Ship_Costates.png
clc

w1=-0.5*c1*z(:,10); w2=-0.5*c2*z(:,11); w3=-0.5*c3*z(:,12);
control=[w1,w2,w3];
pause;
for k=1:3
  subplot(3,1,k); plot(t,z(:,k),'r');
  xlabel('Time t '); ylabel('Control');
end
print C:\Users\Guest\Documents\4july2022\Robot_Ship_Controls.png
clc
```

The state-costate system of ordinary differential equations is coded using Matlab programming language. The following cases are considered:

Case1: Initial condition $\mathbf{z0} = [0;0;0;0;0;0;-1;-1;-1;-1;-1;-1]$. In this initial condition, $[0;0;0;0;0;0]$is the initial state and $[-1;-1;-1;-1;-1;-1]$ is the initial costate.

Case2: Initial condition $\mathbf{z0} = [0;0;\frac{\pi}{2};0;0;0;-1;-1;-1;-1;-1;-1]$. In this initial condition,

$[0;0;\frac{\pi}{2};0;0;0]$ is the initial state and $[-1;-1;-1;-1;-1;-1]$ is the initial costate.

Case3: Initial condition $\mathbf{z0} = [0;0;\pi;0;0;0;-1;-1;-1;-1;-1;-1]$. In this initial condition, $[0;0;\pi;0;0;0]$is the initial state and $[-1;-1;-1;-1;-1;-1]$ is the initial costate.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 31 August 2022 doi:10.20944/preprints202208.0546.v1

Case1: Initial condition $\mathbf{z0} = [0; 0; 0; 0; 0; 0; -1; -1; -1; -1; -1; -1]$.

The other elements are in the computer programs. The following graphs are the corresponding results.

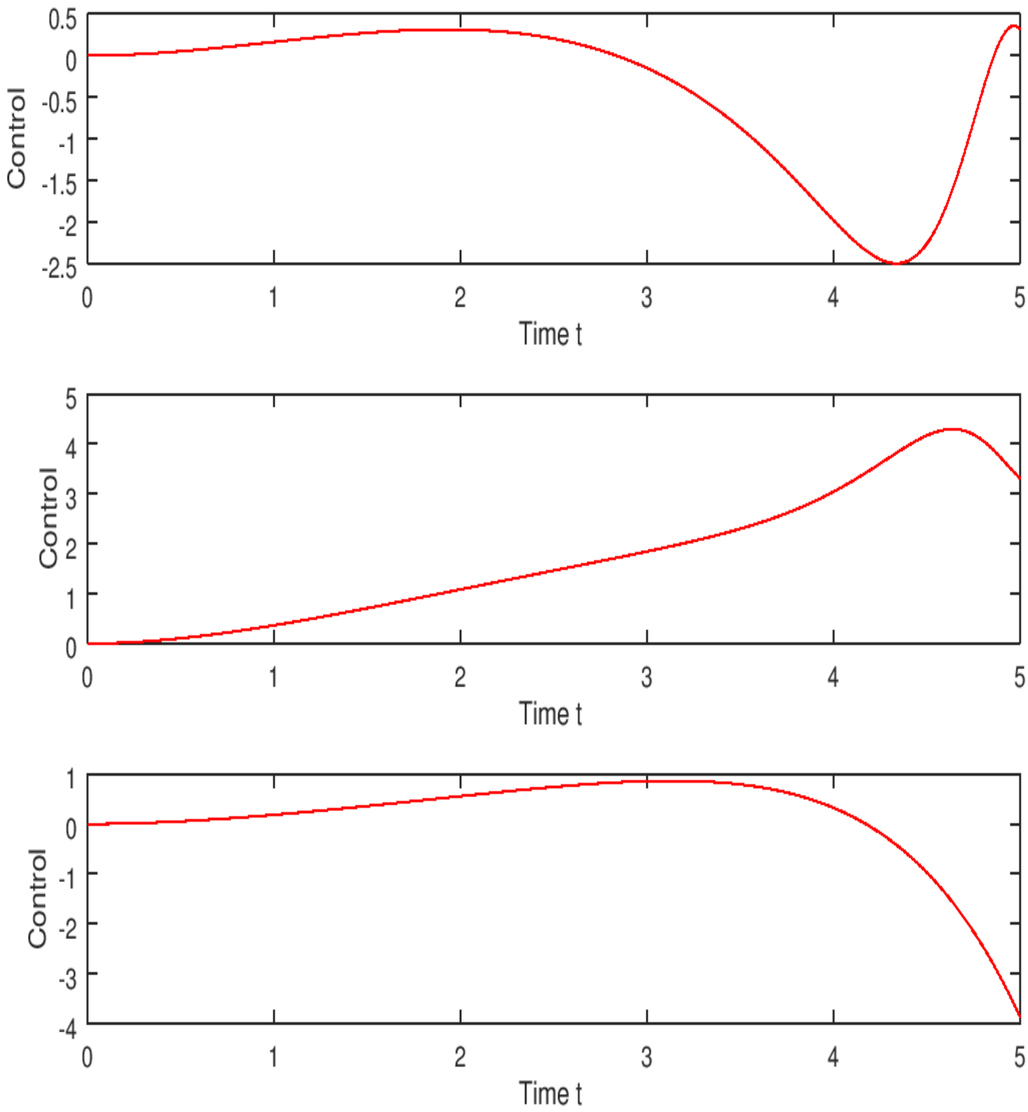

Figure 1: Feasible Control Strategies

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 31 August 2022 doi:10.20944/preprints202208.0546.v1

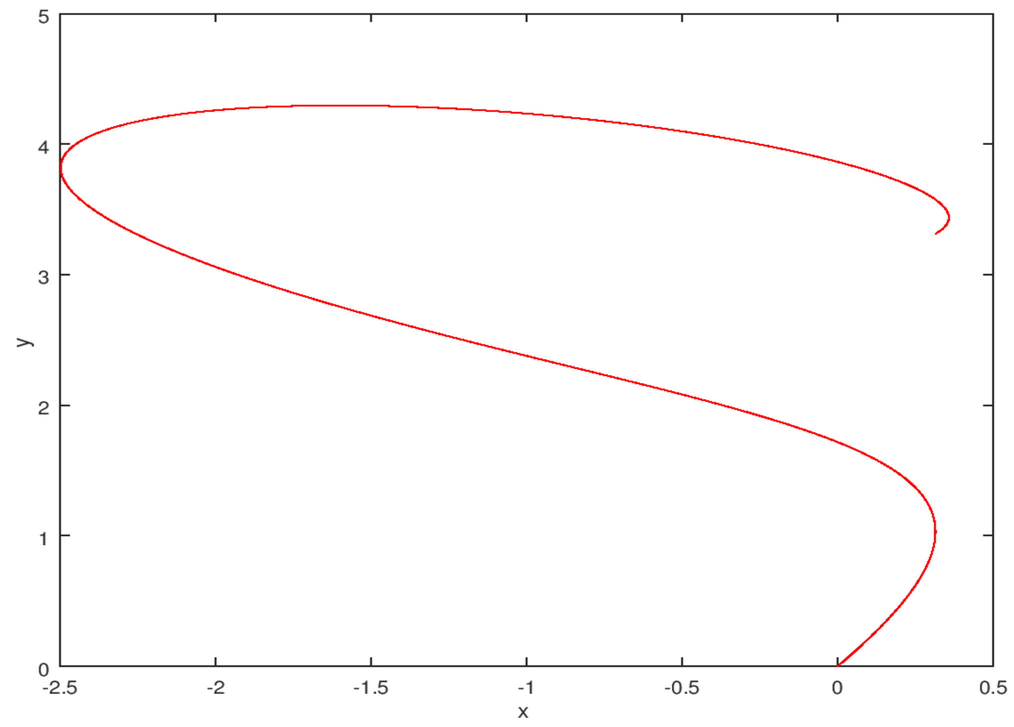

Figure 2: Feasible Trajectory

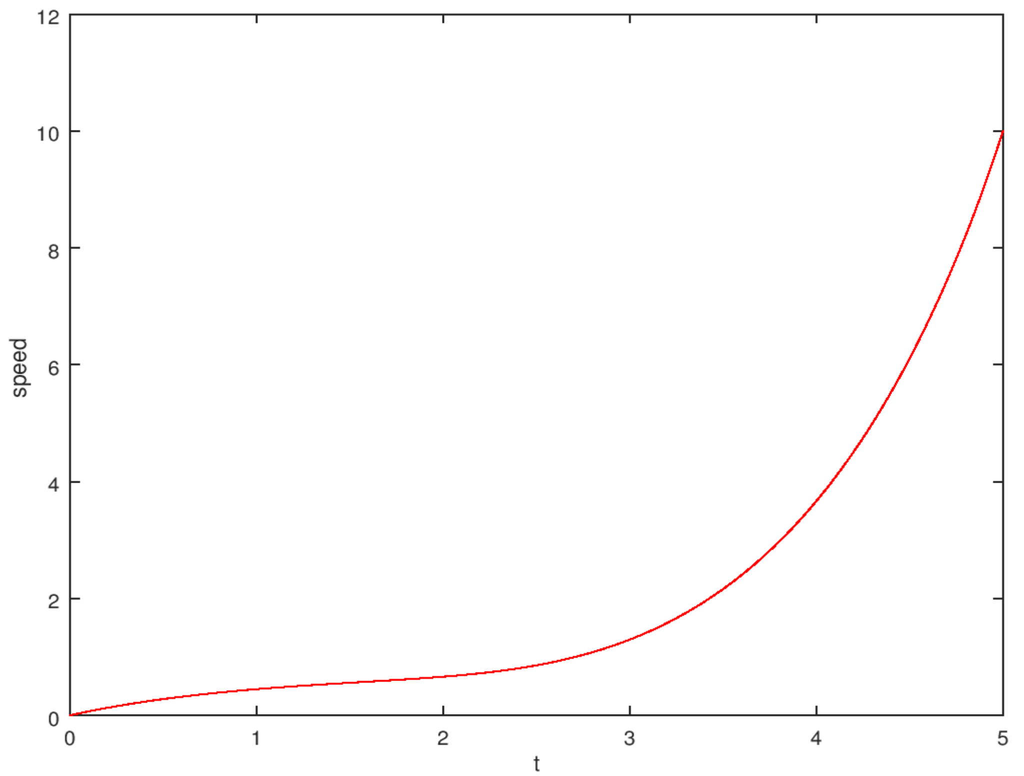

Figure 3: Feasible Speed

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 31 August 2022 doi:10.20944/preprints202208.0546.v1

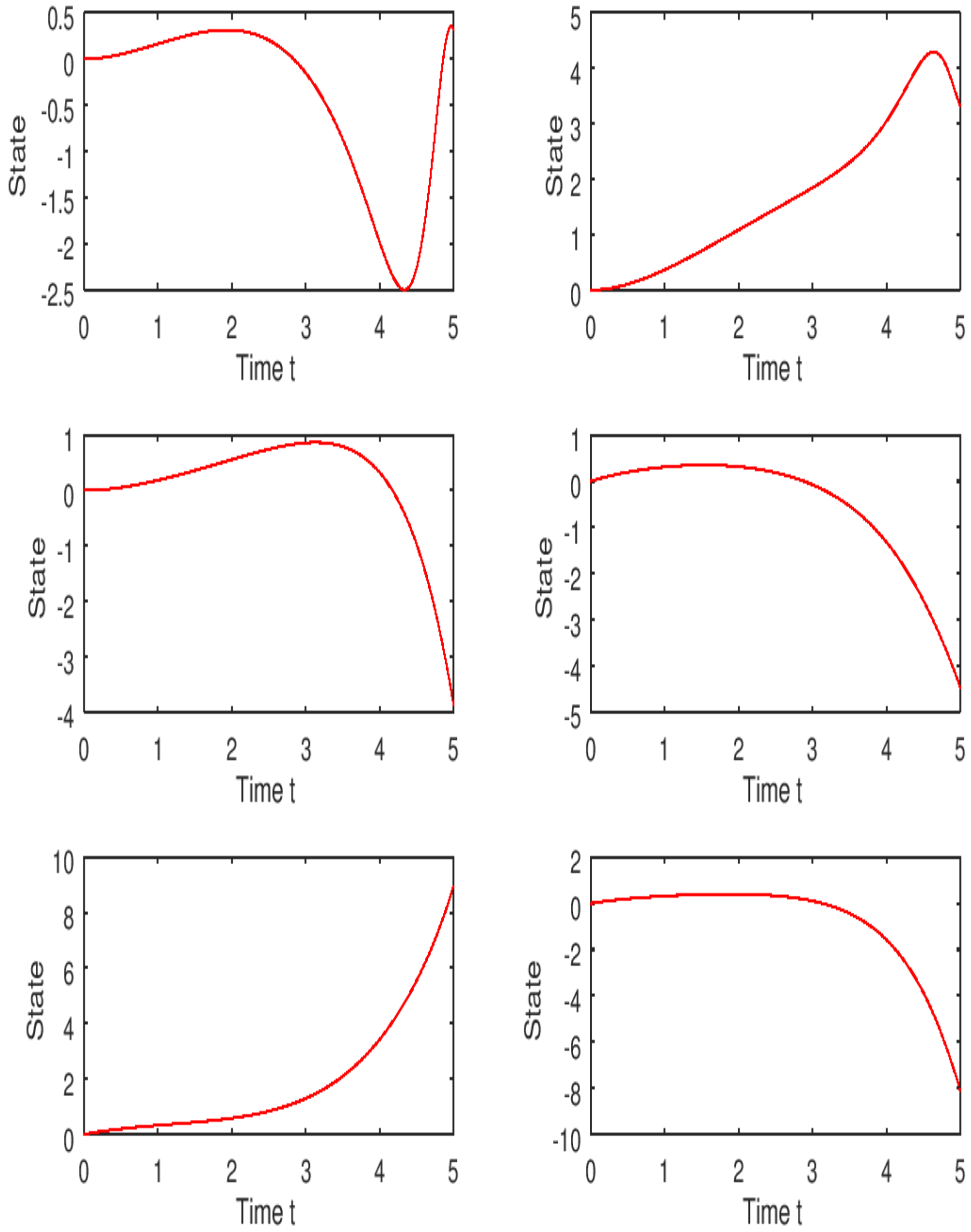

Figure 4: Feasible States

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 31 August 2022 doi:10.20944/preprints202208.0546.v1

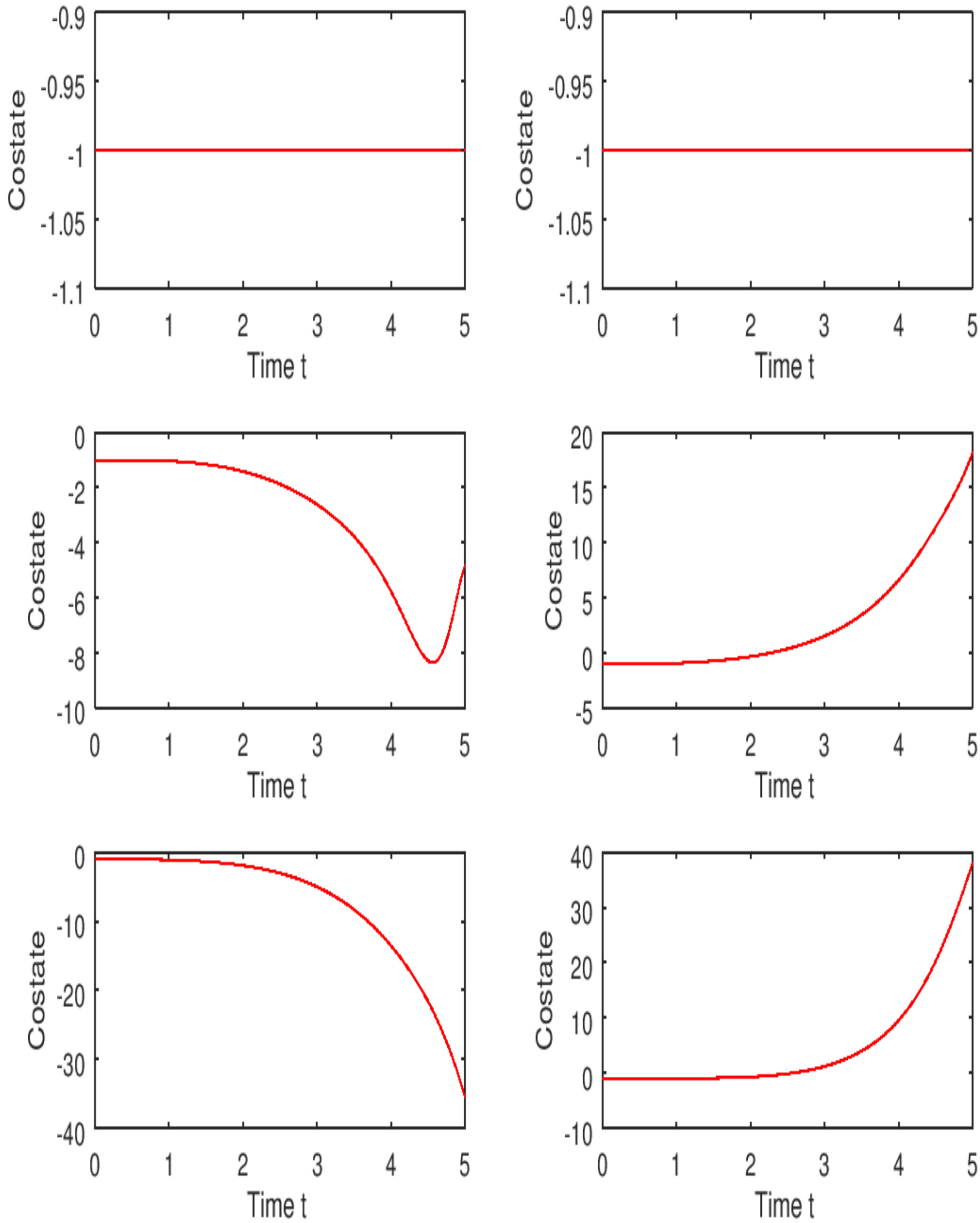

Figure 5: Feasible Costates

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 31 August 2022 doi:10.20944/preprints202208.0546.v1

Case2: Initial state **z0**=$[0;0;\frac{\pi}{2};0;0;0;-1;-1;-1;-1;-1;-1]$.; The other elements are in the computer programs. The following graphs are the corresponding results.

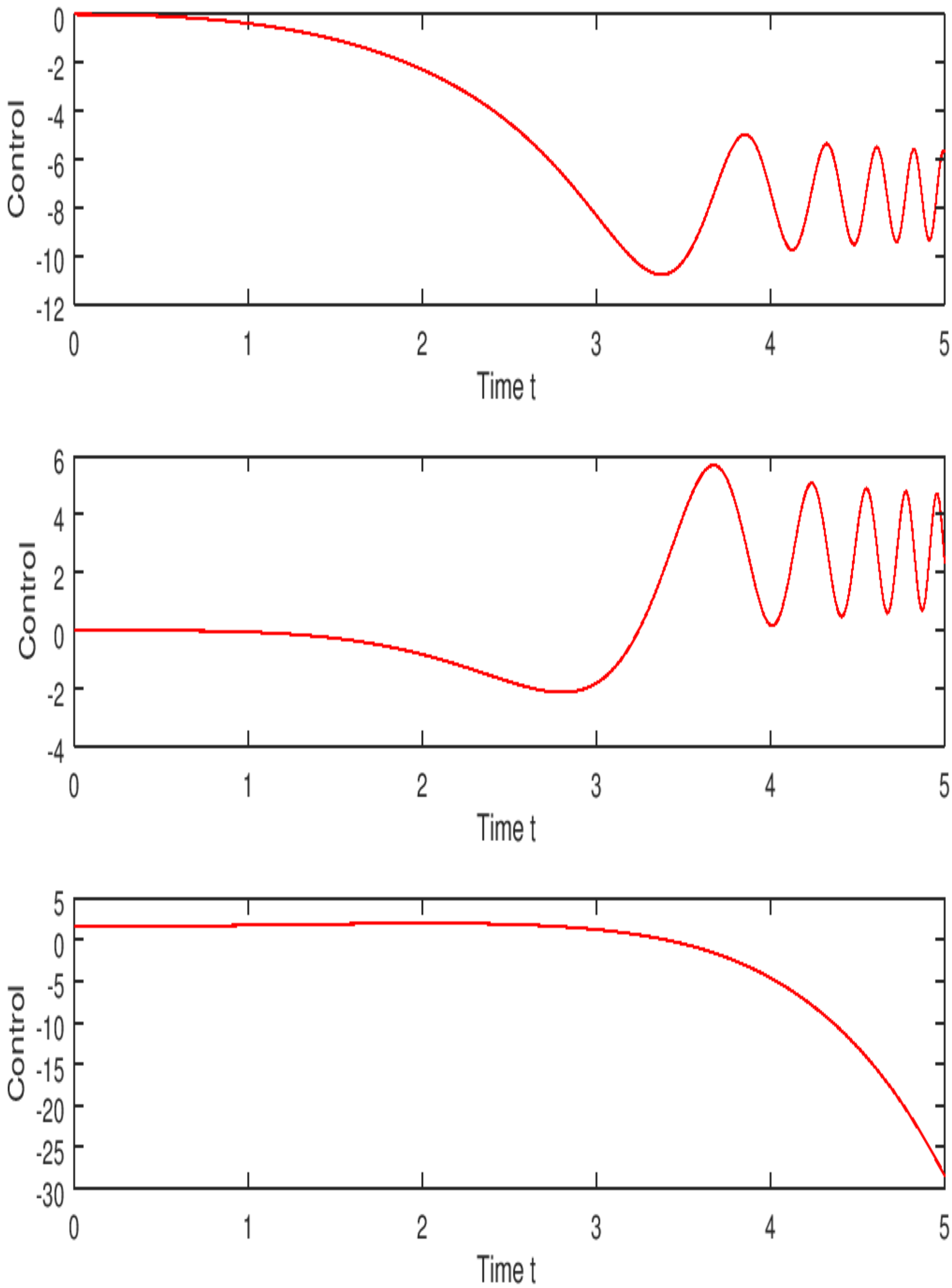

Figure 6: Feasible Control Strategies

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 31 August 2022 doi:10.20944/preprints202208.0546.v1

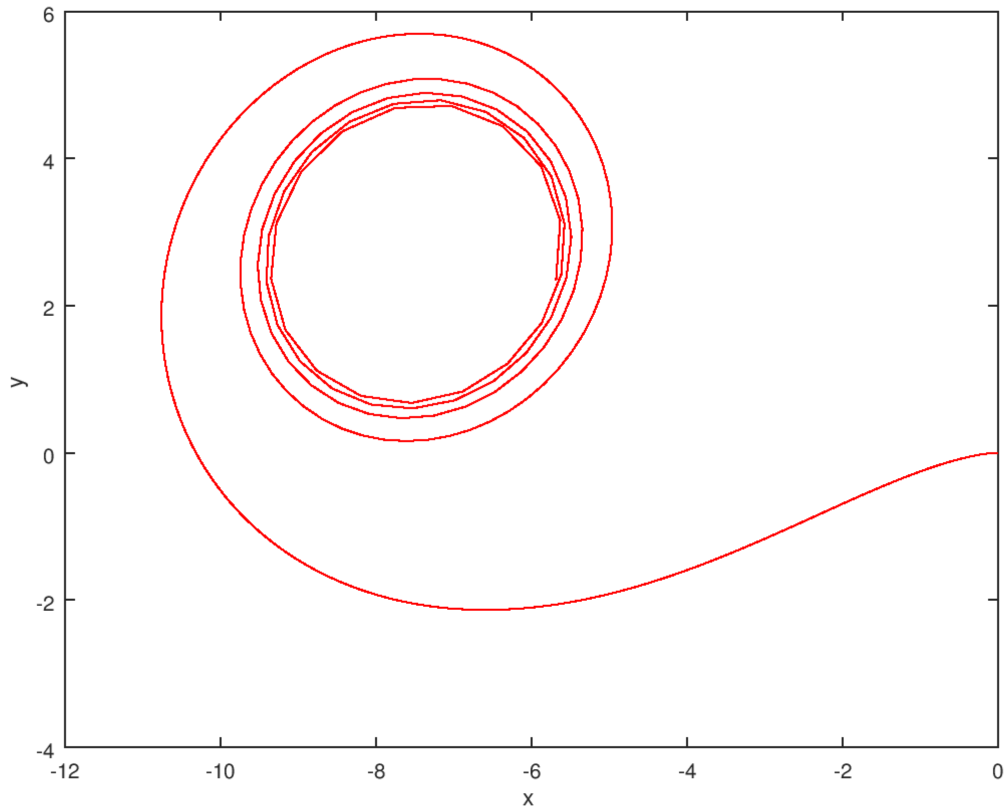

Figure 7: Feasible Trajectory

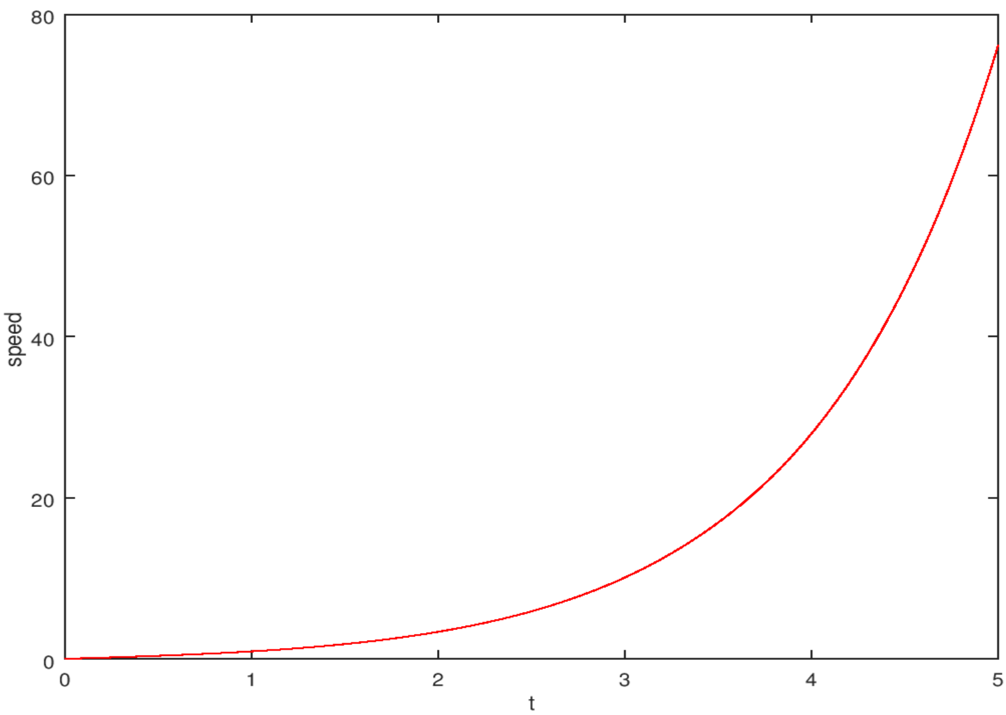

Figure 8: Feasible Speed

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 31 August 2022 doi:10.20944/preprints202208.0546.v1

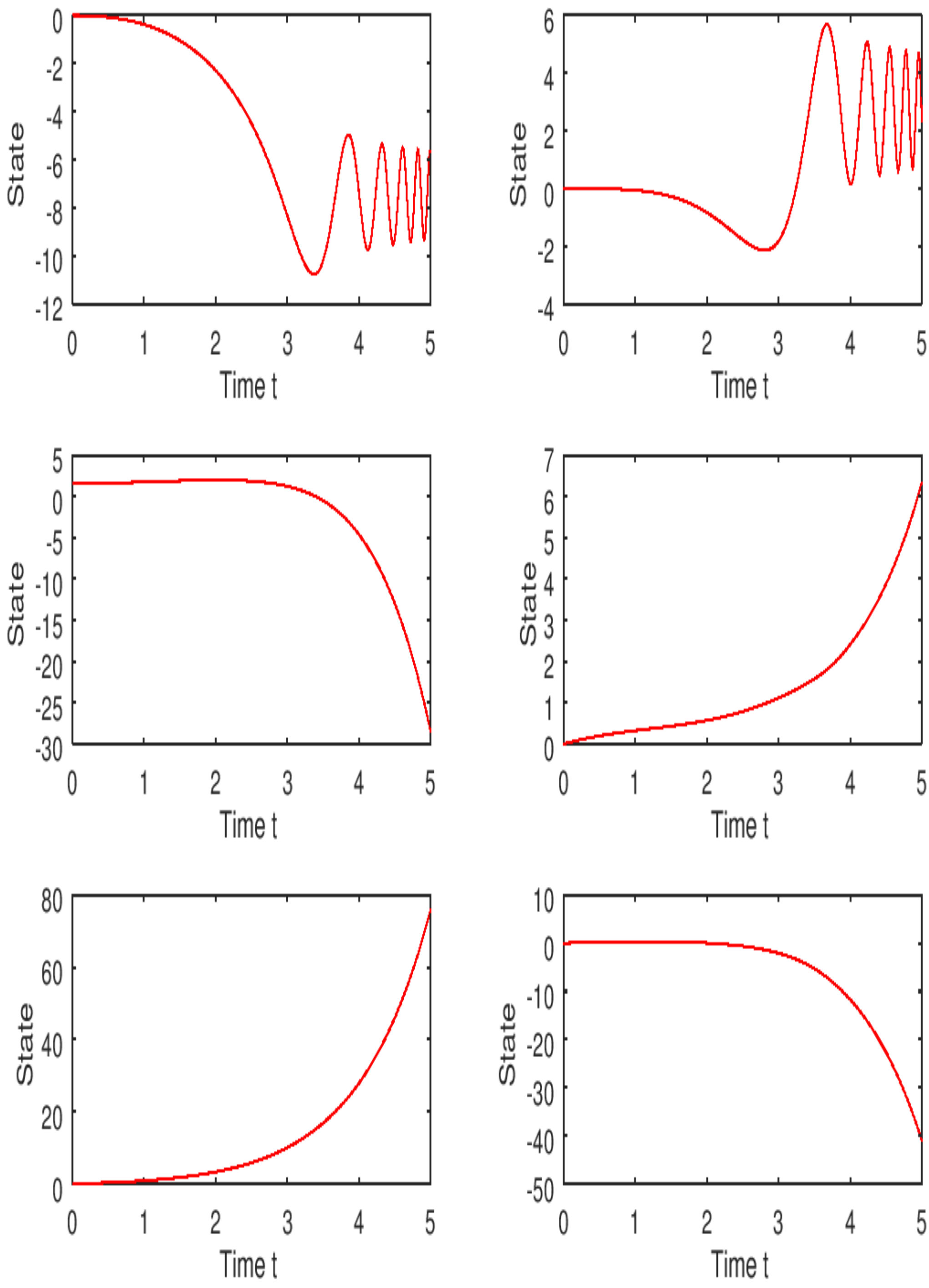

Figure 9: Feasible States

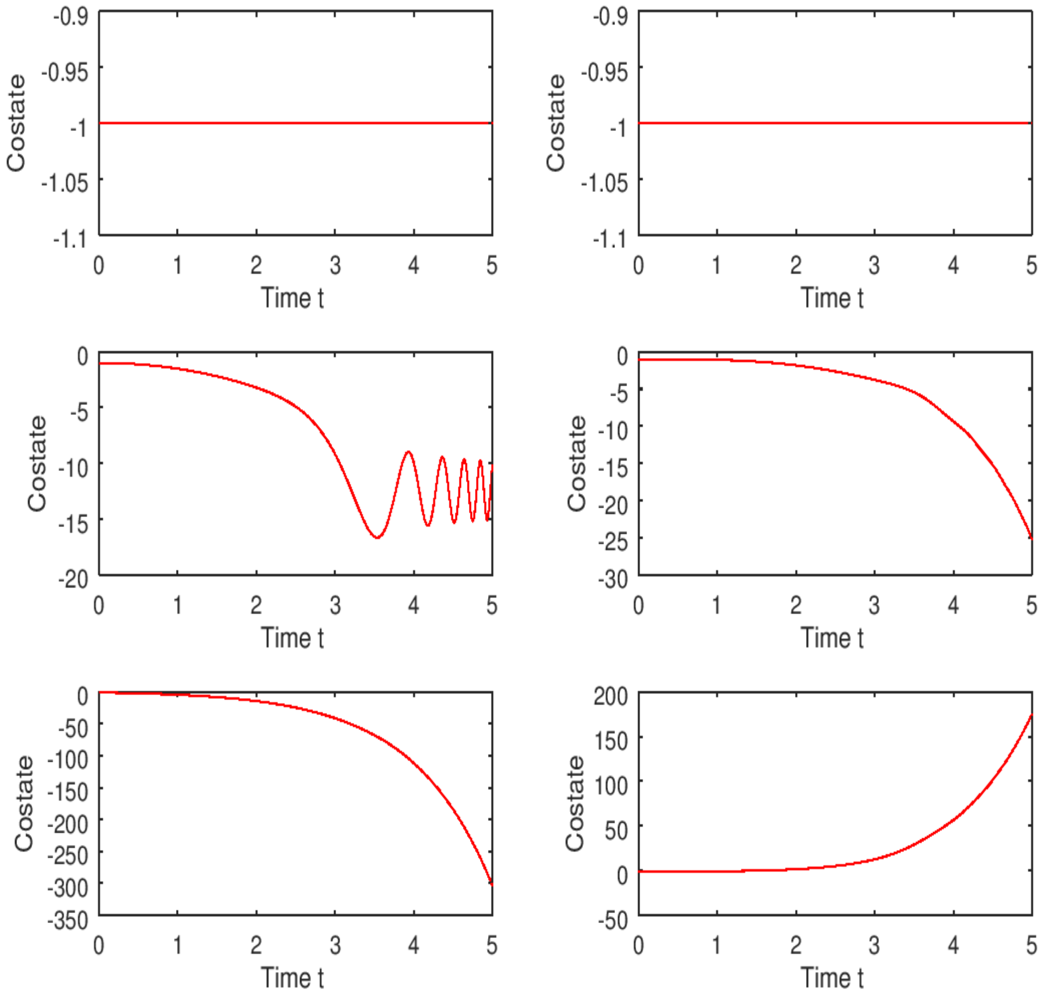

Figure 10: Feasible Costate Functions

Case3: Initial condition **z0**=$[0; 0; \pi; 0; 0; 0; -1; -1; -1; -1; -1; -1]$. The other elements are in the computer programs. The following graphs are the corresponding results.

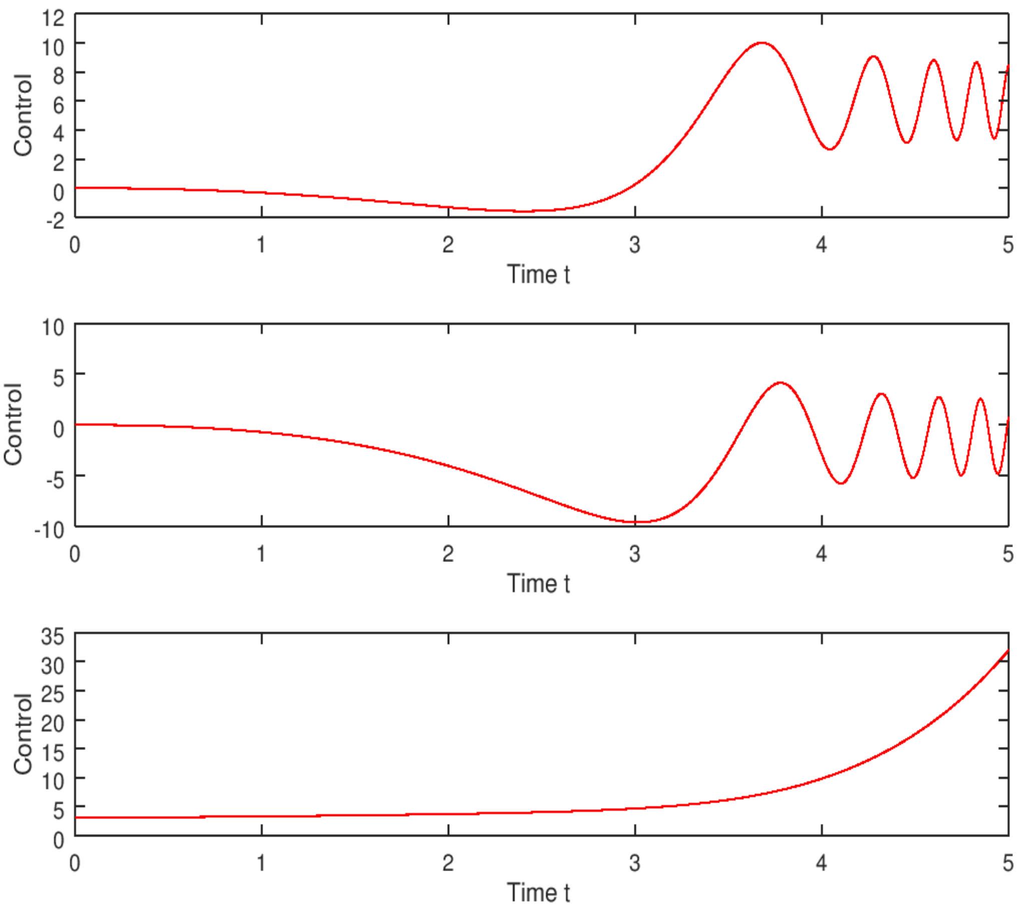

Figure 11: Feasible Control Strategies

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 31 August 2022 doi:10.20944/preprints202208.0546.v1

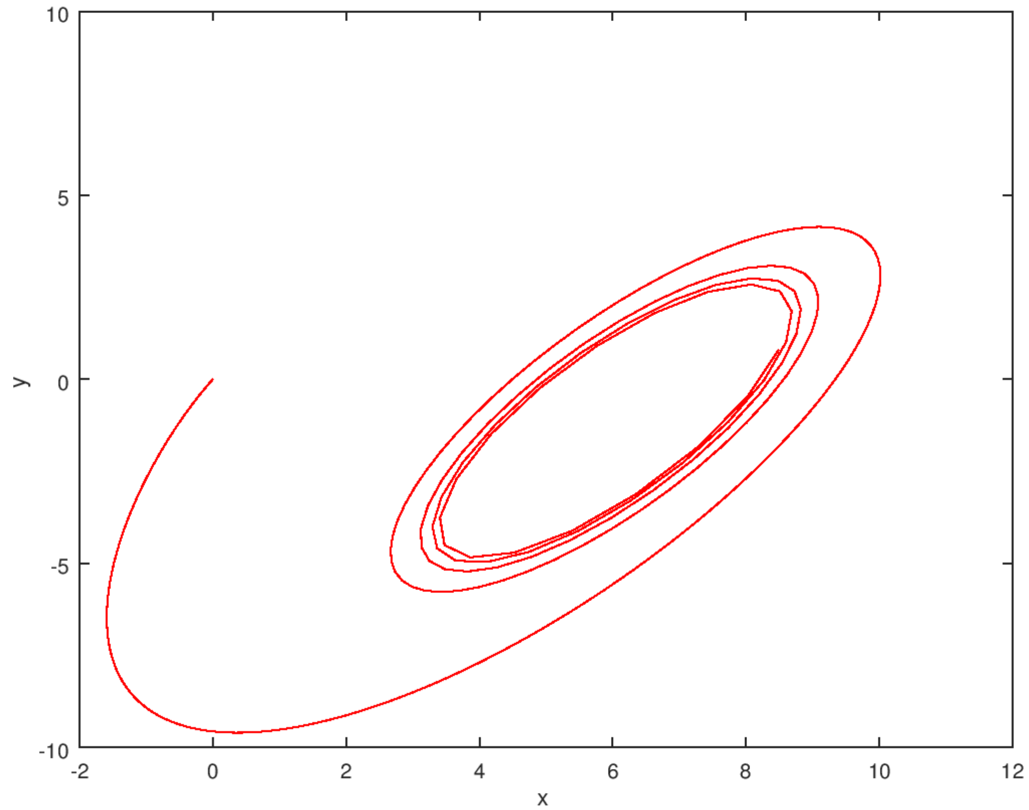

Figure 12: Feasible Trajectory

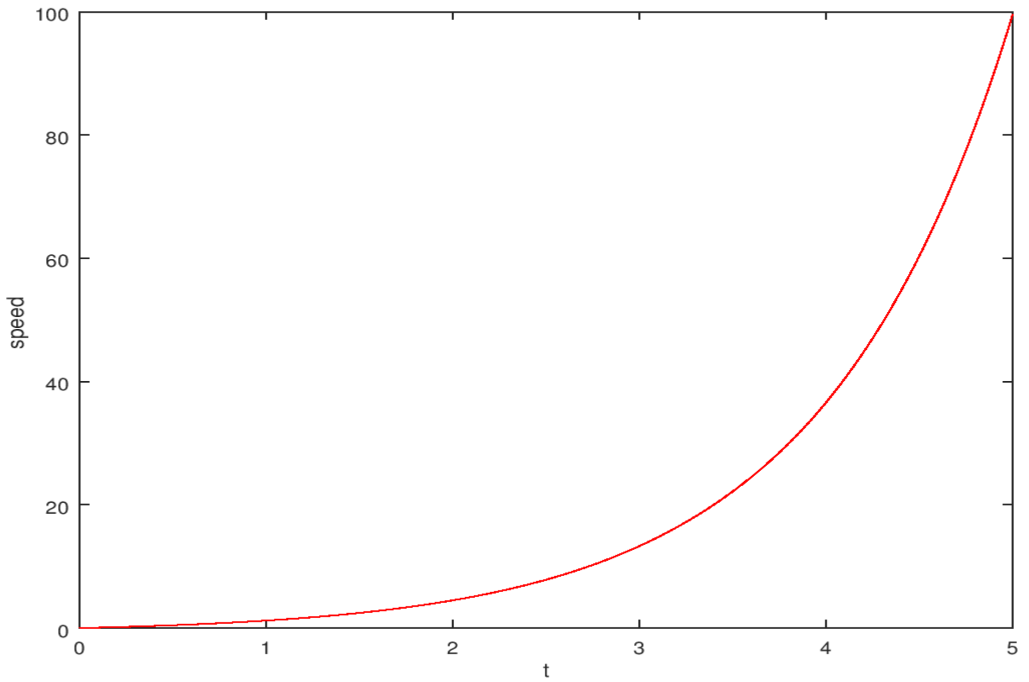

Figure 13: Feasible Speed

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 31 August 2022 doi:10.20944/preprints202208.0546.v1

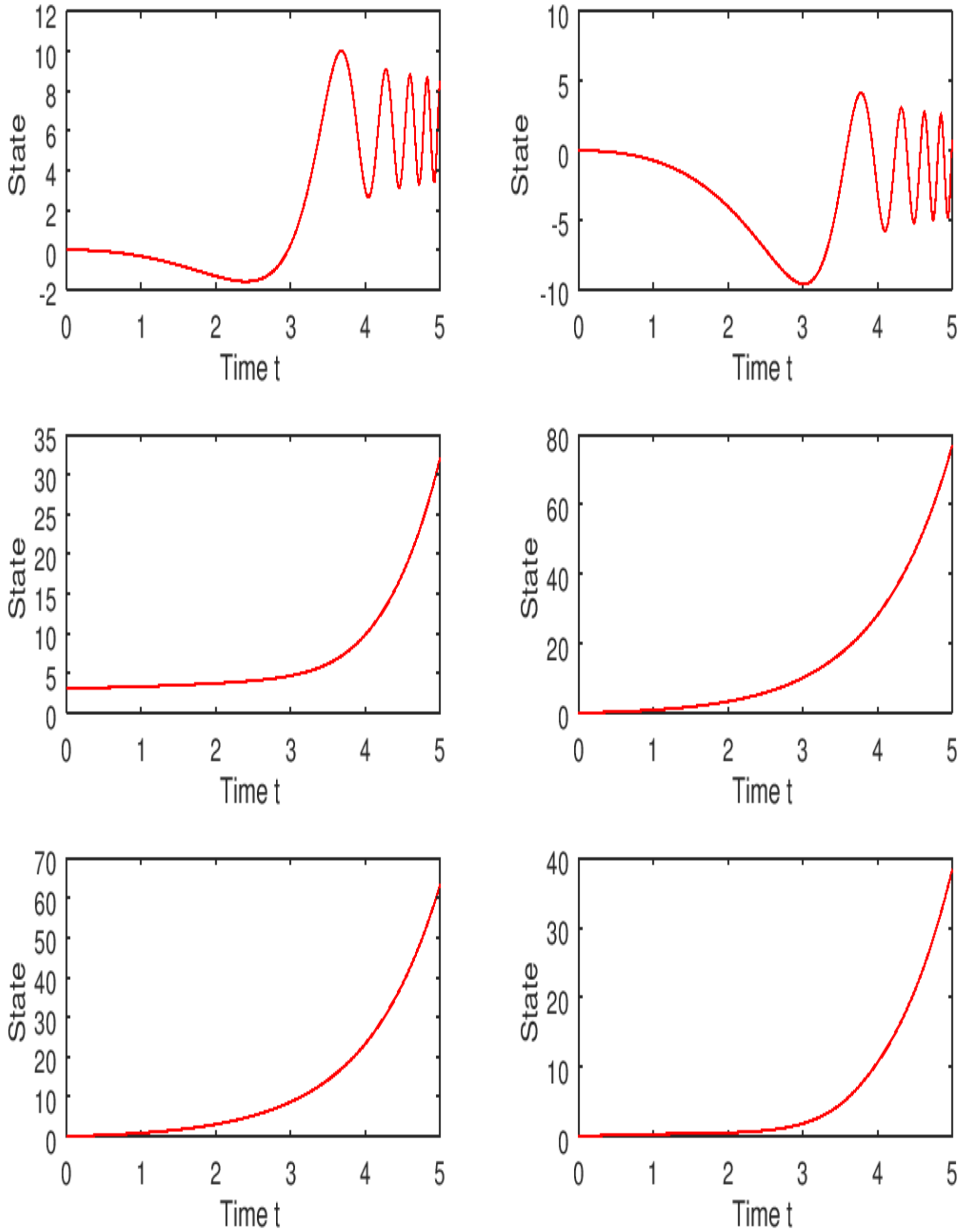

Figure 14: Feasible State functions

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 31 August 2022 doi:10.20944/preprints202208.0546.v1

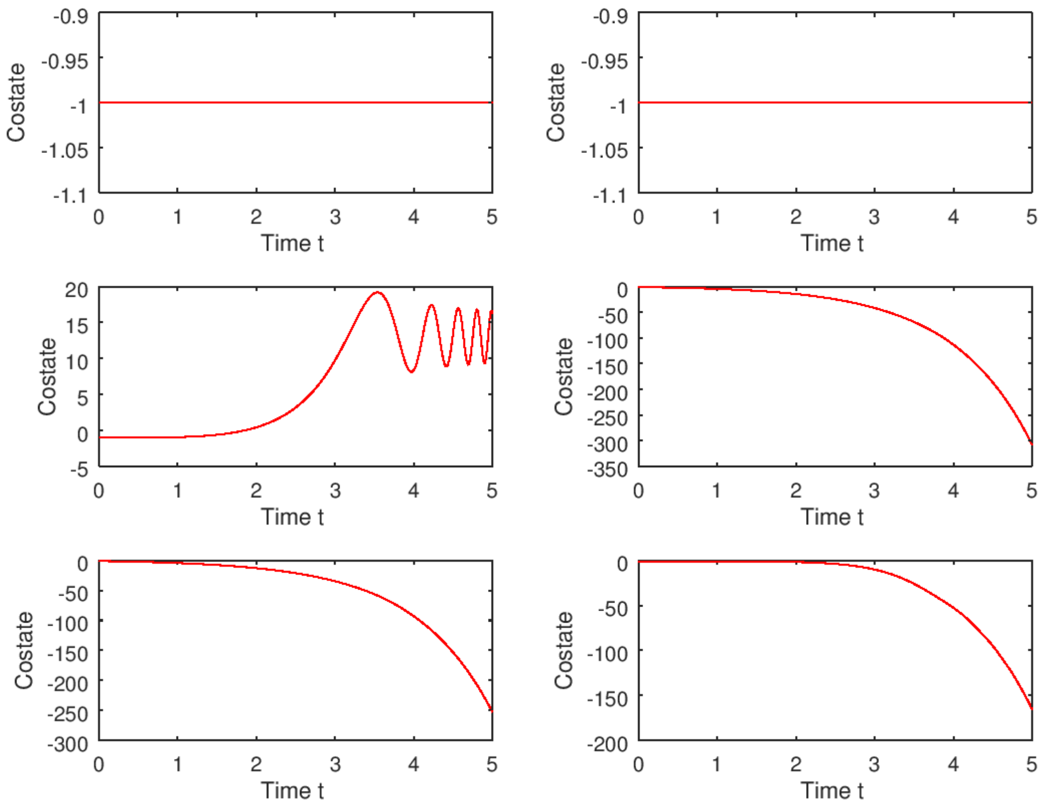

Figure 15: Feasible Costate functions

Conclusion

This paper considered the problem of controlling the navigation of an autonomous ship robot from a given initial state to a final state. Optimal control theory was used to derive the feasible control strategies and the feasible state functions as well as the feasible costate functions. The feasible trajectory is derived from the feasible state functions. Three cases for initial condition are considered from which each output was obtained. The future work will consist of developing a feedback hunting control law for an autonomous ship robot.

References

[1] Jelena L. Pisarov, Gyula Mester, The use of autonomous vehicles in transportation, Tehnika 76(2), 171-177, 2021.

[2] Peter R Wurman, Raffaelo D Andrea, Mick Mountz, coordinating hundreds of cooperative autonomous vehicles in warehouses, AI Magazine 29(1), 9-9, 2008.

[3] Isaac Kaminer, Antonio Pascoal, Eric Hallberg, Carlos Silvestre, Trajectory tracking for autonomous vehicles: An integrated approach to guidance and control, journal of Guidance, control, and Dynamics 21(1), 29-38, 1998

[4] E Borhaug, Kristin Y Pettersen, Adaptive way-point tracking control for underactuated autonomous vehicles, Proceedings of the 44th IEEE conference on Decision and Control, 4028-4034, 2005.

[5] Oskar Levander, Autonomous ships on the high seas, IEEE spectrum 54(2), 26-31, 2017.

[6]Thomas Stathe, gareth Howells, Klans McDonald Maier, Autonomous ship collision avoidance navigation concepts, technologies and techniques, The journal of navigation 61(1), 129-142, 2008

[7] Franz S Hover, Ryan M Eustice, Ayoung kim,, Brendan Englot Hordur Johannson Michael, Advanced Perception, navigation, and plannking for autonomous in-water ship hull inspection The International Journal of Robotics Research 31(12), 1445-1464, 2012.

[8] Paul Ozog, Nicholas Carlevaris-Bianco, Ayoung Kim, Ryan M Eustice, Long-term Mapping Techniques for ship Hull Inspection and surveillance using an autonomous underwater vehicle, journal of Fields Robotics 33(3) , 265-289, 2016.

[9] Vladimir Djapic, Dula Nad, Using collaborative vehicles in mine countermeasures, Oceans 10 IEEE Sydney, 1-7, 2010.

[10] Natalie Klein, Maritime autonomous vehicles within the international lawframework to enhance maritime security, International law framework to enhance maritime security, International law Studies 95(1), 8,2019.

[11] Thomas statheros, Gareth Howels and Klaus Mc Donald Maier, Autonomous chip collision Techniques, Cambridge University Press, 2007.

[12] Abril, J. , Salom, J.. & Calvo, O. (1997) Fuzzy Control of a sailboat, International Journal of Approximate Reasoning, Volume 16, 359-375.

[13] Cheekuang Tam, Richard Bucknail and Alistair Greig, Review of Collision Avoidance and Path Planning Methods for ships in close range encounters, journal of navigation,2009.

 doi:10.20944/preprints202208.0546.v1

[14] Michael Hoy, Alexey S. Matveev and Andrey V. Savkin, Algorithms for collision-free navigation of mobile robots

[15] Dmitriev, S. P., Kolesov, N. V., , Osipov, A. V & Romanycheva, G. N. (2003) , System of Intelligent of Intelligent support of a ship navigator for collision avoidance Expert System.

[16] Hwang, C. N., Yang. J. M. & Chiang, C. Y. (2001), The design of Fuzzy Collision Avoidance Expert System.

[17] Smierzchalski, , R. & Michalewicz, Z. (2000), Modeling of ship trajectory in collision situations by an evolutionary algorithm, IEEE Transactions on Evolutionary computation, volume 4, 227-241.

[18] Wilson , P. A., Harris, C. J. & Hong, X. (2003) A line of sight counteraction navigation algorithm for ship encounter collision avoidance

[19] Xiao Ming, Z. & Ito, M. (2001). The design of Fuzzy collision avoidance Expert Systemimplemented by H-Infinity autopilot, journal of Marine science and Technology, 9, 25-37.

[20] Lee, S. M., Kwon, K. Y. & Joh, J. (2004), A fuzzy logic for autonomous navigation of marine vehicles satisfying COLREG guidelines. International journal of Control, Automation and systems, 2, 171-181.

[21] Volodymyr Blintsov, Hanna Hrudimina, Mathematical modelling of autonomous underwater vehicle propulsion and steering complex operation in oblique (beveled) water flow, Eastern-European journal of Enterprise Technologies 4(9)100, 19-26,

[22] Luis Luttgens, Benjamin Jurgelucks, Heinrich Wernsing, Sylvain Roy, Christof Buskens & Kathrin Flabkamp (2022) Autonomous navigation of ships by combining optimaltrajectory planning with informed graph search, Mathematical and Computer Modelling of Dynamical Systems, Vol. 28, N0. 1,1-27.