Article

Modeling and Control of a Front-Axle Bicycle Robot

Masiala Mavungu

University of Johannesburg, Department of Mechanical Engineering Science, South Africa
msmvp7219@gmail.com or masialam@uj.ac.za

Abstract: This paper develops feasible control strategies and associated system responses to bring an autonomous front-axle bicycle robot from specified initial conditions to final conditions such that a specific performance index is minimized. To solve the problem, the following approach is used: The feasible controls derived from the normal equations of optimality are substituted into the state and the costate systems and form a combined control-free state-costate system which is vectorized to enable and ease the application of a numerical method. A computer program written in Matlab computer programming language, codes a fourth-order Runge-Kutta numerical method and then solve the combined state-costate system of ordinary differential equations. The obtained results are the feasible bicycle robot trajectory, the feasible state functions, the feasible costate functions and the feasible control functions. Associated Computational Simulations are designed and provided to persuade on the effectiveness and the reliability of the approach.

Keywords: autonomous vehicle, bicycle Robot, modelling, optimal control, path planning, differential equation, initial value problem, Runge-Kutta, scientific computing

1. Introduction

Nowaday, there exist a lot of innovations in the areas of telecommunication and networking, remote sensing, computer vision, robotics, etc. The technology of connected devices and that of self-driving vehicles are continually impacting human lives. They cause the Industrial Operators and Managers to dream as much as possible and then to create a lot of business opportunities. They also cause the Academic Researchers to develop relevant tools and methods for vehicle modelling and path planning for efficient and reliable control and management and then create a lot of research questions and opportunities. Path planning and control of an autonomous bicycle robot are highly connected to signal processing, image processing, computer vision, control system, digital logic specially in the component of obstacle avoidance and stability.

The prediction of the dynamics of bicycle robots and other vehicle robots in general have become a current and important topic for the industrial operators and managers, and then attract the attention of engineers, mathematicians, computer scientists, physicists, etc. Such Researchers develop suitable strategies, tools and methods to solve all associated problems. There exist a certain amount of works on autonomous bicycle robots carried out since some years. For example papers [1]-[9] deal with bicycle robots and vehicle robots in general. This paper uses optimal control theory to compute feasible control strategies and feasible state trajectories of an autonomous bicycle such that the bicycle running cost is minimized.

The main contributions of this paper are the derivation of two feasible control strategies, the computation the system response defined by six feasible state functions, six feasible costate (adjoint) functions. To solve the system combining the state and the costate ordinary differential equations Matlab computer programs were developed and applied.

This paper is organized as follows: Section 2 develops different mathematical models and defines the problem as an optimal control problem. Section3 derives the Hamiltonian of the control system and solves the normal equations of optimality to obtain the expressions of the control functions. Section 4 applies Pontryagin's Minimum Principle to determine all relevant equations yielding the solutions. Section 5 develops relevant computer programs to determine the feasible control trajectories, the corresponding feasible state trajectories and all the other outputs.

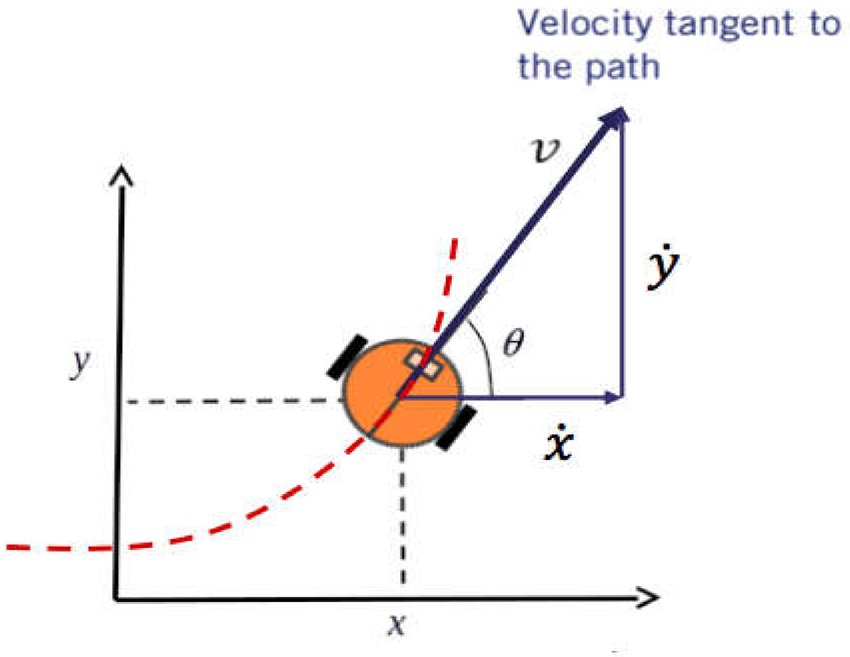

Figure 1. Vehicle Robot Geometric Model.

2. Mathematical Models

2.1. Objective functional

In this paper, the total running cost to be minimized is as follows:

$$J(\boldsymbol{u}) = J(\xi_1, \xi_2) = \int_{t_0}^{t_f} ({\xi_1}^2 + {\xi_2}^2) dt \quad (1)$$

where t_0 and t_f are respectively the bicycle motion's starting and final times, ξ_1 and ξ_2 are the reference commands which control respectively the bicycle heading angular velocity and the steering angular velocity. $h(t) = {\xi_1}^2 + {\xi_2}^2$ is the cost rate.

2.2. Control System, Kinematic Model

The motion of an autonomous bicycle is modelled as follows:

$$\frac{dx}{dt} = c_1 \omega cos(\theta + \delta) \quad (2)$$

$$\frac{dy}{dt} = c_1 \omega sin(\theta + \delta) \quad (3)$$

$$\frac{d\theta}{dt} = c_2 \omega tan(\delta) \quad (4)$$

$$\frac{d\delta}{dt} = c_3 \varphi \quad (5)$$

where $c_1 = R$ and $c_2 = R/L$ are constant of proportionality, R and L are respectively the radius of each wheel and the distance between the centre of the rear and the front wheels. (x, y) is the coordinates of the projection of the front wheel's center on the horizontal plane, θ the heading angle, δ is the steering angle, φ is the steering angular velocity.

The reference commands which regulate the bicycle angular velocity and the steering angular velocity are modelled as solution to a closed-loop system defined by:

$$\frac{d\omega}{dt} = -a_1 \omega + a_1 \xi_1 \quad (6)$$

$$\frac{d\varphi}{dt} = -a_2 \varphi + a_2 \xi_2 \quad (7)$$

ξ_1 and ξ_2 are the unknown input control functions to be developed.

The whole robot kinematic control system is

$$\frac{dx}{dt} = c_1\omega cos(\theta + \delta) \quad (8)$$
$$\frac{dy}{dt} = c_1\omega sin(\theta + \delta) \quad (9)$$
$$\frac{d\theta}{dt} = c_2\omega tan(\delta) \quad (10)$$
$$\frac{d\delta}{dt} = c_3\varphi \quad (11)$$
$$\frac{d\omega}{dt} = -a_1\omega + a_1\xi_1 \quad (12)$$
$$\frac{d\varphi}{dt} = -a_2\varphi + a_2\xi_2 \quad (13)$$

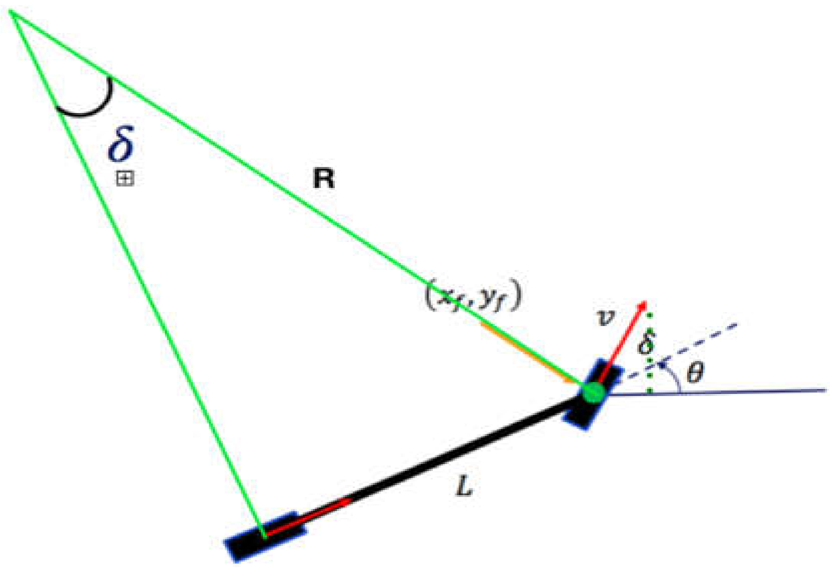

Figure 2. Front-Axle Bicycle Robot Geometric Model.

2.3. Problem Formulation

This paper addresses the following problem:

Compute the feasible control strategies and the associated feasible state functions, also called feasible robot system responses, for the autonomous bicycle to drive from a given initial state to a final state such that the total running cost of the bicycle is minimized.

3. Hamiltonian and Feasible Controls

The Hamiltonian of the system is given by

$$H(t, Y(t), \alpha(t), \omega(t), \varphi(t)) = h(t) + \sum_{k=1}^{6} \alpha_k(t) f_k(Y(t), \omega(t), \varphi(t)) \quad (14)$$

Where we have

$h(t) = \xi_1^2 + \xi_2^2$ is energy cost rate,

$f_1(\boldsymbol{Y}(t), \omega(t), \varphi(t)) = c_1\omega cos(\theta + \delta)$ is the x component of the linear velocity of the bicycle,

$f_2(\boldsymbol{Y}(t), \omega(t), \varphi(t)) = c_1\omega sin(\theta + \delta)$ is the y component of the linear velocity of the bicycle,

$f_3(\boldsymbol{Y}(t), \omega(t), \varphi(t)) = c_2\omega tan(\delta)$ is the heading angular velocity of the bicycle,

$f_4(\boldsymbol{Y}(t), \omega(t), \varphi(t)) = c_3\varphi$ is the steering angular velocity of the bicycle,

$f_5(\boldsymbol{Y}(t), \omega(t), \varphi(t)) = -a_1\omega + a_1\xi_1$ is the rate of change of the bicycle heading angular velocity,

$f_6(\boldsymbol{Y}(t), \omega(t), \varphi(t)) = -a_2\varphi + a_2\xi_2$ is the rate of change of the bicycle steering angular velocity.

$\boldsymbol{Y}(t) = (x(t), y(t), \theta(t), \delta(t), \omega(t), \varphi(t))$ is the unkown state vector function.

$\boldsymbol{\alpha}(t) = (\alpha_1(t), \alpha_2(t), \alpha_3(t), \alpha_4(t), \alpha_5(t), \alpha_6(t))$ is the unkown costate (adjoint) vector function.

The feasible control normal equations for optimality are as follows

$$\frac{\partial H}{\partial \xi_1} = 2\xi_1^* + a_1\alpha_5 = 0 \qquad (15)$$

$$\frac{\partial H}{\partial \xi_2} = 2\xi_2^* + a_2\alpha_6 = 0 \qquad (16)$$

The feasible controls are given by

$$\xi_1^* = -0.5a_1\alpha_5 \qquad (17)$$

$$\xi_2^* = -0.5a_2\alpha_6 \qquad (18)$$

4. Pontryagin's Minimum Principle

If $\boldsymbol{u}^* = (\xi_1^*, \xi_2^*)$ is the feasible control of the above problem and $\boldsymbol{Y}^* = (x^*, y^*, \theta^*, \delta^*, \omega^*, \varphi^*)$ the corresponding feasible system response, then there exists a costate vector

$\boldsymbol{\alpha}^* = (\alpha_1^*, \alpha_2^*, \alpha_3^*, \alpha_4^*, \alpha_5^*, \alpha_6^*)$ such that

$$J(\boldsymbol{u}^*) \leq J(u) \qquad (19)$$

$$\frac{dx^*}{dt} = c_1\omega^* cos(\theta^* + \delta^*) \qquad (20)$$

$$\frac{dy^*}{dt} = c_1\omega^* sin(\theta^* + \delta^*) \qquad (21)$$

$$\frac{d\theta^*}{dt} = c_2\omega^* tan(\delta^*) \qquad (22)$$

$$\frac{d\delta^*}{dt} = c_3\varphi^* \qquad (23)$$

$$\frac{d\omega^*}{dt} = -a_1\omega^* + a_1\xi_1^* \qquad (24)$$

$$\frac{d\varphi^*}{dt} = -a_2\varphi^* + a_2\xi_2^* \qquad (25)$$

$$\frac{d\alpha_1^*}{dt} = 0 \qquad (26)$$

$$\frac{d\alpha_2^*}{dt} = 0 \qquad (27)$$

$$\frac{d\alpha_3^*}{dt} = c_1\omega^*(\alpha_1^* \sin(\theta^* + \delta^*) - \alpha_2^* \cos(\theta^* + \delta^*)) \qquad (28)$$

$$\frac{d\alpha_4^*}{dt} = c_1\omega^*(\alpha_1^* \sin(\theta^* + \delta^*) - \alpha_2^* \cos(\theta^* + \delta^*)) - c_2\alpha_3^*\omega^* cos(\delta^*) \qquad (29)$$

$$\frac{d\alpha_5^*}{dt} = -(c_1\alpha_1^* cos(\theta^* + \delta^*) + c_1\alpha_2^* sin(\theta^* + \delta^*) + c_2\alpha_3^*\omega^* sin(\delta^*) - a_1\alpha_5^*) \qquad (30)$$

$$\frac{d\alpha_6^*}{dt} = -c_3\alpha_4^* + a_2\alpha_6^* \qquad (31)$$

By letting $\boldsymbol{u}^* = (\xi_1^*, \xi_2^*)$, with $\xi_1^* = -0.5a_1\alpha_5^*$ and $\xi_2^* = -0.5a_2\alpha_6^*$ for the control variables,

$z_1^* = x^*$, $z_2^* = y^*$, $z_3^* = \theta^*$, $z_4^* = \delta^*$, $z_5^* = \omega^*$ and $z_6^* = \varphi^*$ for the state variables,

$z_7^* = \alpha_1^*$, $z_8^* = \alpha_2^*$, $z_9^* = \alpha_3^*$, $z_{10}^* = \alpha_4^*$, $z_{11}^* = \alpha_5^*$ and $z_{12}^* = \alpha_6^*$ for the costate variables

and by combining all the state and costate variables into a vector as $\boldsymbol{z} = [\boldsymbol{Y}, \boldsymbol{\alpha}]$, then the combined state-costate system can be rewritten as follows:

$$\frac{dz_1^*}{dt} = c_1z_5^* cos(z_3^* + z_4^*) \qquad (32)$$

$$\frac{dz_2^*}{dt} = c_1z_5^* sin(z_3^* + z_4^*) \qquad (33)$$

$$\frac{dz_3^*}{dt} = c_2z_5^* tan(z_4^*) \qquad (34)$$

$$\frac{dz_4^*}{dt} = c_3z_6^* \qquad (35)$$

$$\frac{dz_5^*}{dt} = -a_1z_5^* + a_1\xi_1^* \qquad (36)$$

$$\frac{dz_6^*}{dt} = -a_2z_6^* + a_2\xi_2^* \qquad (37)$$

$$\frac{dz_7^*}{dt} = 0 \qquad (38)$$

$$\frac{dz_8^*}{dt} = 0 \qquad (39)$$

$$\frac{dz_9^*}{dt} = c_1z_5^*(z_7^* \sin(z_3^* + z_4^*) - z_8^* \cos(z_3^* + z_4^*)) \qquad (40)$$

$$\frac{dz_{10}^*}{dt} = c_1z_5^*(z_7^* \sin(z_3^* + z_4^*) - z_8^* \cos(z_3^* + z_4^*)) - c_2z_5^*z_9^* cos(z_4^*) \qquad (41)$$

$$\frac{dz_{11}^*}{dt} = -(c_1z_5^* cos(z_3^* + z_4^*) + c_1z_6^* sin(z_3^* + z_4^*) + c_2z_9^* sin(z_4^*) - a_1z_{11}^*) \qquad (42)$$

$$\frac{dz_{12}^*}{dt} = -c_3z_{10}^* + a_2z_{12}^* \qquad (43)$$

The initial heading angles and the final time are the parameters on which is based our performance analysis. For all the considered cases, the other state and costate variables, the initial conditions did not change. The aim is also to see how far and in which direction the vehicle can drive. In order to solve the state-costate system, I have developed an algorithm which can solve any system of ordinary differential equations. The program is written in Matlab as a set of codes coding a fourth-order Runge-Kutta numerical method. Below is the program

```
function [t,y] = runge_v2(fs,t0,tf,N,y0)
h=(tf-t0)./(N-1); % h is the step size for the discretization.
t=t0:h:tf; % t is the time vector. % N is the number of discrete points.
t=t'; t0 and tf are respectively the initial and final time.
%    y0 is the initial vector solution.
y = zeros(N,length(y0)); % y is initialized to zero.
y(1,:) = y0.'; % The solution at the starting time.
for n = 2:N
k1 = feval(fs,t(n-1),y(n-1,:));
k2 = feval(fs,t(n-1)+(h/2),y(n-1,:)+(h/2)*k1');
k3 = feval(fs,t(n-1)+(h/2),y(n-1,:)+(h/2)*k2');
k4 = feval(fs,t(n-1)+h,y(n-1,:)+h*k3');
y(n,:) = y(n-1,:)+(h/6)*(k1'+2*k2'+2*k3'+k4');
end
```

The above algorithm can be translated judiciously into any programming language. It can be called to solve any initial value problem. Let's use it to solve the above combined state-costate system of ordinary differential equations (32)-(43). The Matlab function coding such a system is as follows:

```
function dzdt= front_wheel_robot(t,z)
dzdt = zeros(12,1);
c1=2; c2=1; c3=0.75; a1=0.25; a2=0.25; % These are Constants of proportionality c1=R; c2=R/L;
dzdt (1)=c1*z(5)*cos(z(3)+z(4)); % Equation 32
dzdt (2)=c1*z(5)*sin(z(3)+z(4)); %    Equation 33
dzdt (3)=c2*z(5)*tan(z(4))*cos((z(5))); %    Equation 34
dzdt (4)= c3*z(6); %    Equation 35
dzdt (5)=-a1*z(5) + a1*(-0.5*a1*z(11)); %    Equation 36
dzdt (6)=-a2*z(6) + a2*(-0.5*a2*z(12)); %    Equation 37
dzdt (7)=0; %    Equation 38
dzdt (8)=0; %    Equation 39
dzdt (9)= c1*z(5)*(z(7)*sin(z(3)+z(4))- z(8)*cos(z(3)+z(4))); % Equation 40
dzdt (10)= c1*z(5)*(z(7)*sin(z(3)+z(4))- z(8)*cos(z(3)+z(4)))-c2*z(5)*z(9)*cos(z(4)); %    Equation 41
dzdt (11)=-(c1*z(5)*cos(z(3)+z(4))+c1*z(6)*sin(z(3)+z(4)))+ c2*z(9)*sin(z(4))-a1*z(11); % Equation 42
dzdt (12)= -c3*z(10)+a2*z(12); %    Equation 43
```

Such a function can also be written in a compact vector form as follows:

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 31 August 2022 doi:10.20944/preprints202208.0533.v1

```
function dzdt= front_wheel_robot(t,z)
dzdt = zeros(12,1);
c1=2; c2=1; c3=0.75; a1=0.25; a2=0.25; % These are Constants of proportionality c1=R; c2=R/L;
dzdt =[c1*z(5)*cos(z(3)+z(4));
       c1*z(5)*sin(z(3)+z(4));
       c2*z(5)*tan(z(4))*cos((z(5)));
       c3*z(6);
      -a1*z(5) + a1*(-0.5*a1*z(11));
      -a2*z(6) + a2*(-0.5*a2*z(12));
       0;
       0;
       c1*z(5)*(z(7)*sin(z(3)+z(4))- z(8)*cos(z(3)+z(4)));
       c1*z(5)*(z(7)*sin(z(3)+z(4))- z(8)*cos(z(3)+z(4)))-c2*z(5)*z(9)*cos(z(4));
      -(c1*z(5)*cos(z(3)+z(4))+c1*z(6)*sin(z(3)+z(4)))+ c2*z(9)*sin(z(4))-a1*z(11);
      -c3*z(10)+a2*z(12);
```

The main function is given by:

```
function main_front_wheel

clear all

clc

format short

c1=2; c2=1; c3=0.75; a1=0.25; a2=0.25;
t0=0; tf=5; N=501; h=(tf-t0)/(N-1);
t=t0:h:tf;
z=zeros(N,12); % Initialization of z
angle=[-pi/2;0;pi/2;pi];
% loop for the heading angle
for s=1:length(angle)
z0=[0;0;angle(s);0;0;0;ones(6,1)];
t=t';
[t,z]=runge_v2('front_wheel_robot',t0,tf,N,z0);
control1=-0.5*a1*z(:,11); control2=-0.5*a2*z(:,12); control=[ control1, control2];
dx= c1*z(:,5).*cos(z(:,3)+ z(:,4));
dy= c1*z(:,5).*sin(z(:,3) + z(:,4));
dtheta=c2*z(:,5).*sin(z(:,4));
dDelta= c3*z(:,6);
dOmega = a1*(-0.5*a1*z(:,11)-z(:,5));
dPhi= a2*(-0.5*a2*z(:,12)-z(:,6));
```

 doi:10.20944/preprints202208.0533.v1

```
% Feasible trajectory

subplot(2,2,s); plot(z(:,1),z(:,2),'r'); xlabel('x');ylabel('y=f(x)');
print C:\Users\Guest\Documents\20avril2022avcontrol\bicycletrajectory.png
% Control strategies

subplot(4,2,2*s - 1);plot(t,control1,'r');xlabel('Time t in seconds ');ylabel('Control1');
subplot(4,2,2*s);plot(t,control2,'r');xlabel('Time t in seconds ');ylabel('Control2');
print C:\Users\Guest\Documents\20avril2022avcontrol\bicycleControls.png
% Velocities

subplot(4,3,3*s-2);plot(t,dx,'r'); xlabel('Time t in seconds ');ylabel('x velocity');
subplot(4,3, 3*s-1);plot(t,dy,'r'); xlabel('Time t in seconds ');ylabel(' y velocity ');
subplot(4,3,3*s);plot(t, c1*abs(z(:,5)),'r'); xlabel('Time t in seconds ');ylabel('speed');
print C:\Users\Guest\Documents\20avril2022avcontrol\bicycle_velocity.png
end
```

5. Computational Simulations

From the above programs, main_front_wheel is the main function calling function runge_v2(fs,t0,tf,N,y0) to solve the system combining the state and the costate ordinary differential equations coded by function front_wheel_robot(t,z). After running the above main function, the following graphs are obtained:

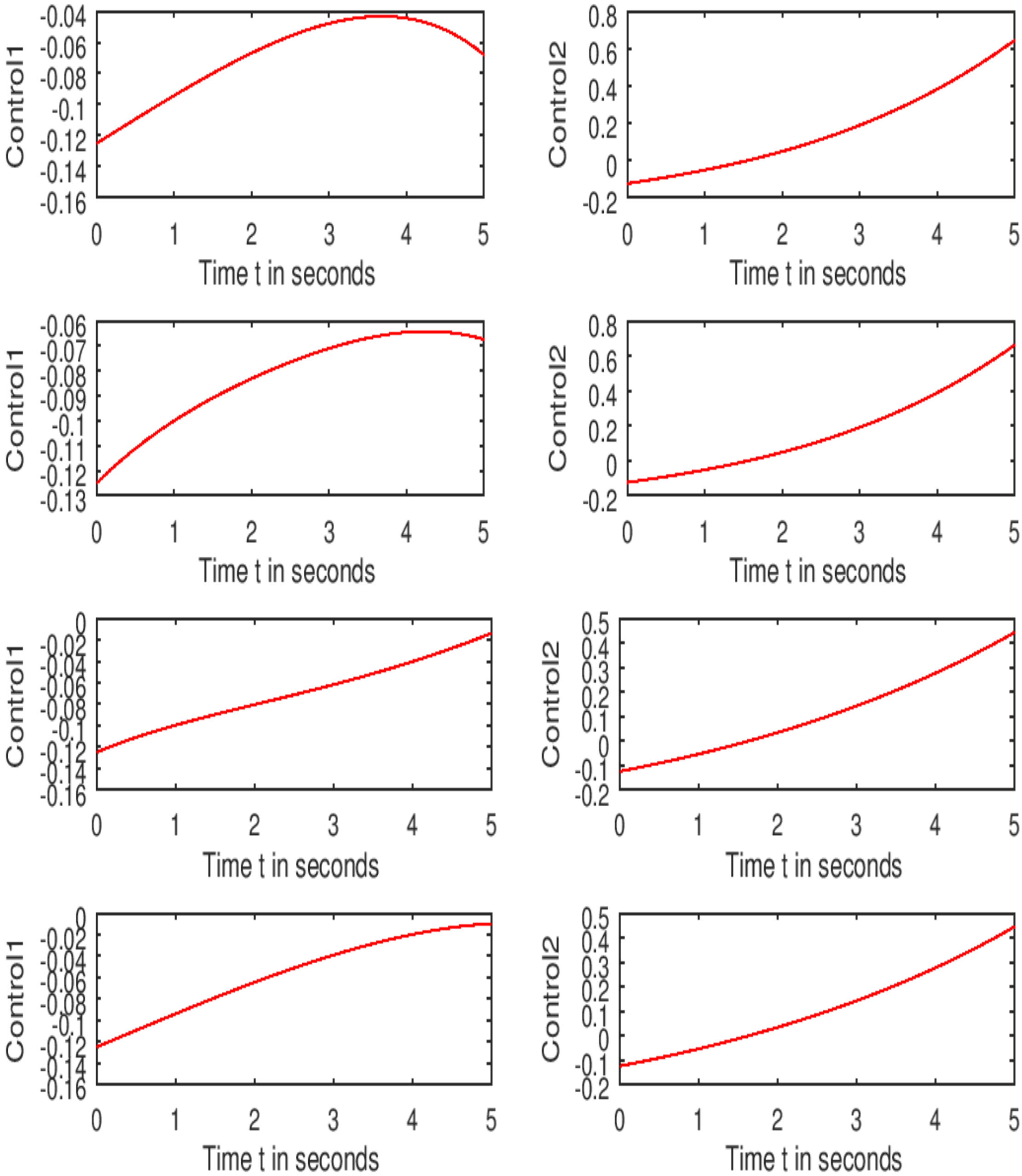

Figure 3. Feasible Control Strategies (tf=5 seconds).

Each initial angle corresponds to a pair (control1, control2) of feasible control strategies. Since we have 4 initial angles $-\frac{\pi}{2}, 0, \frac{\pi}{2}$ and π, then we have also 8 feasible control strategies. From the controls of figure 3 we have the feasible trajectories given by:

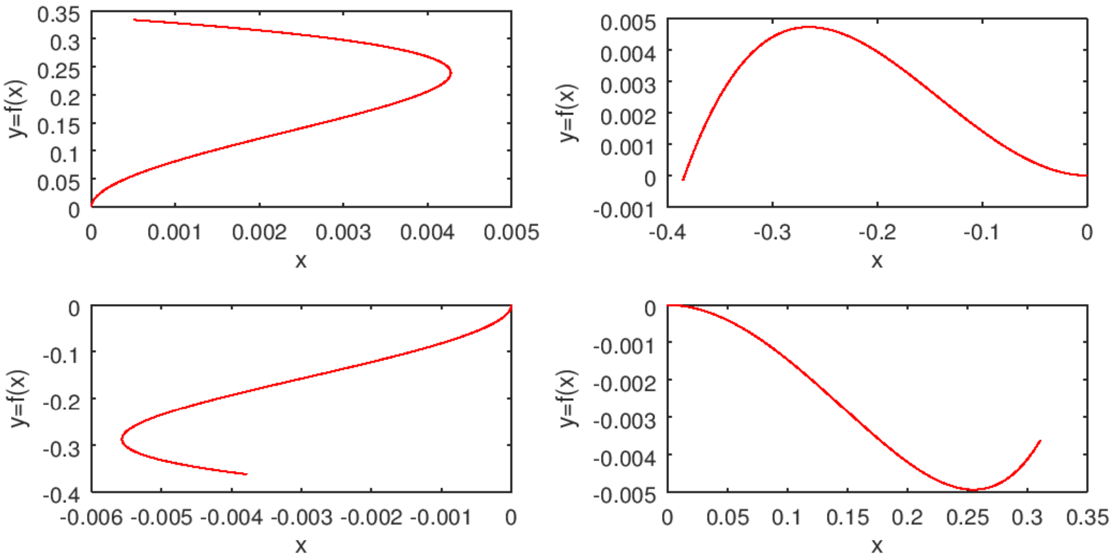

Figure 4. Feasible Robot Trajectories (tf=5 seconds).

In figure 4 each initial angle corresponds to a feasible trajectory. Since we have 4 initial angles ($-\frac{\pi}{2}, 0, \frac{\pi}{2}$ and π), then we have also 4 feasible trajectories on the (x, y) horizontal plane. The graph of the feasible trajectory was deduced from the feasible state functions by selecting only the x state function and the y state function.

 doi:10.20944/preprints202208.0533.v1

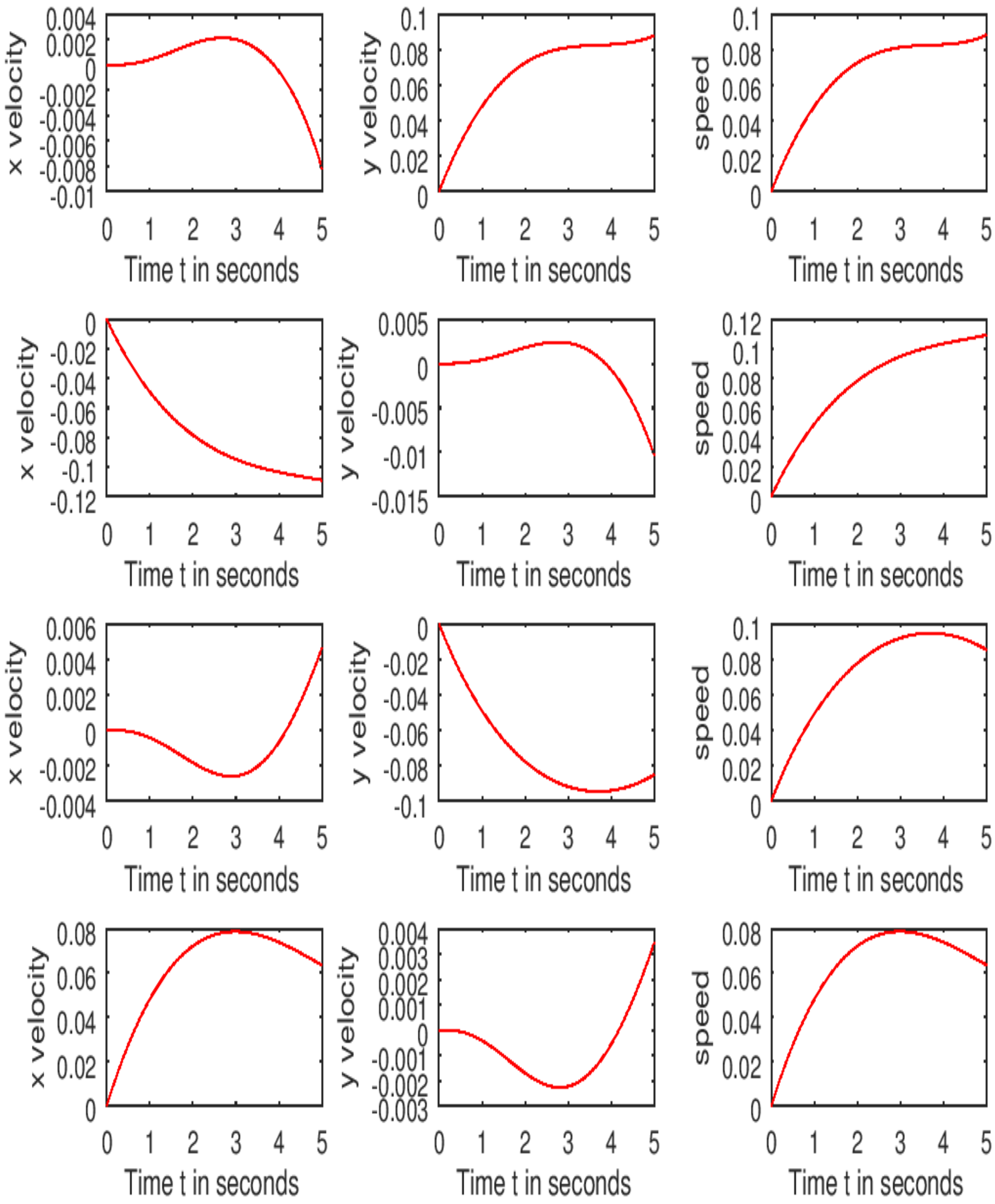

Figure 5. Feasible Velocities (tf=5 seconds).

Each initial angle corresponds to a set of three (x component, y component, speed) feasible velocities. Since we have 4 initial angles $-\frac{\pi}{2}, 0, \frac{\pi}{2}$ and π, then we have also 12 feasible velocities.

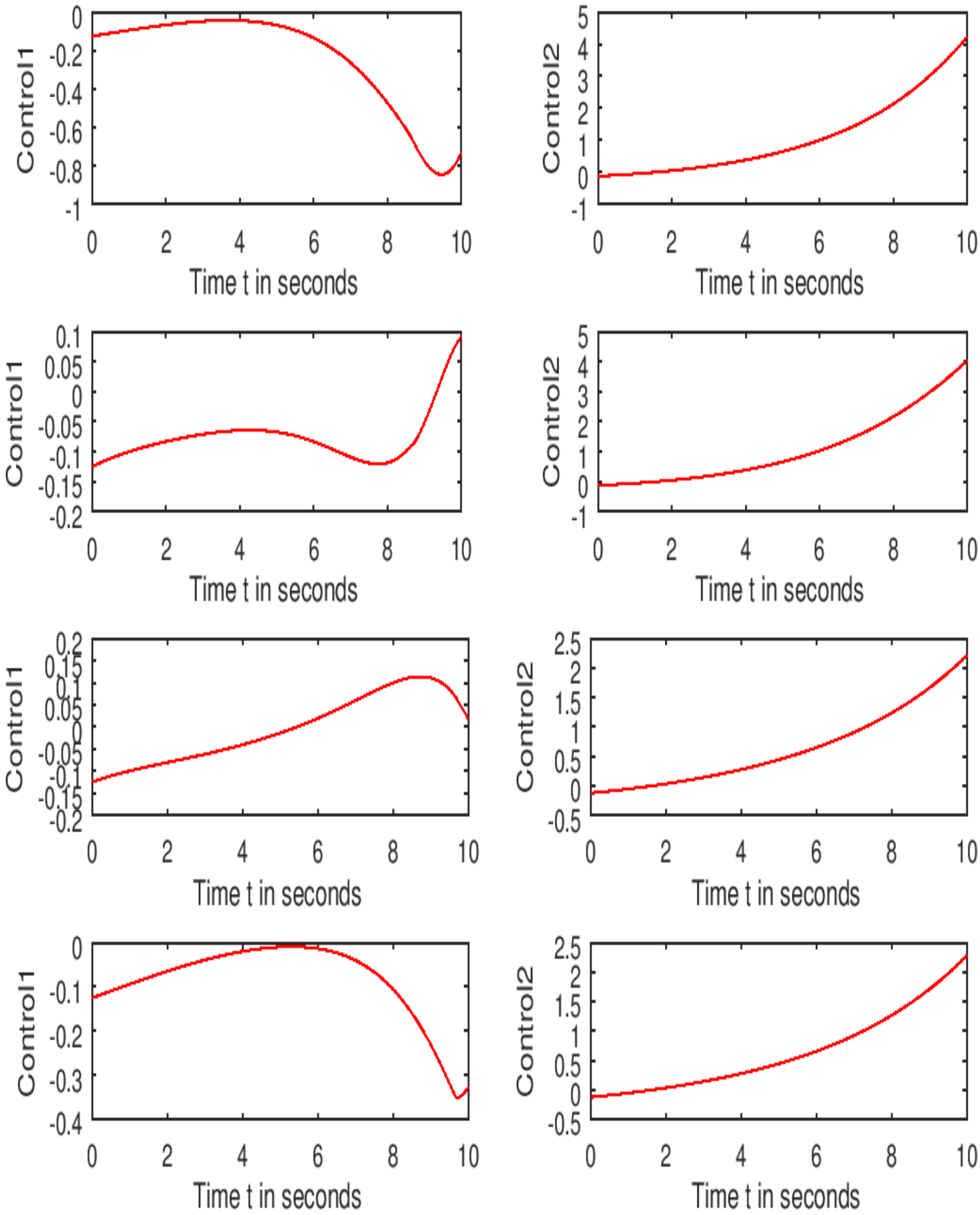

Figure 6. Feasible Control Strategies (tf=10 seconds).

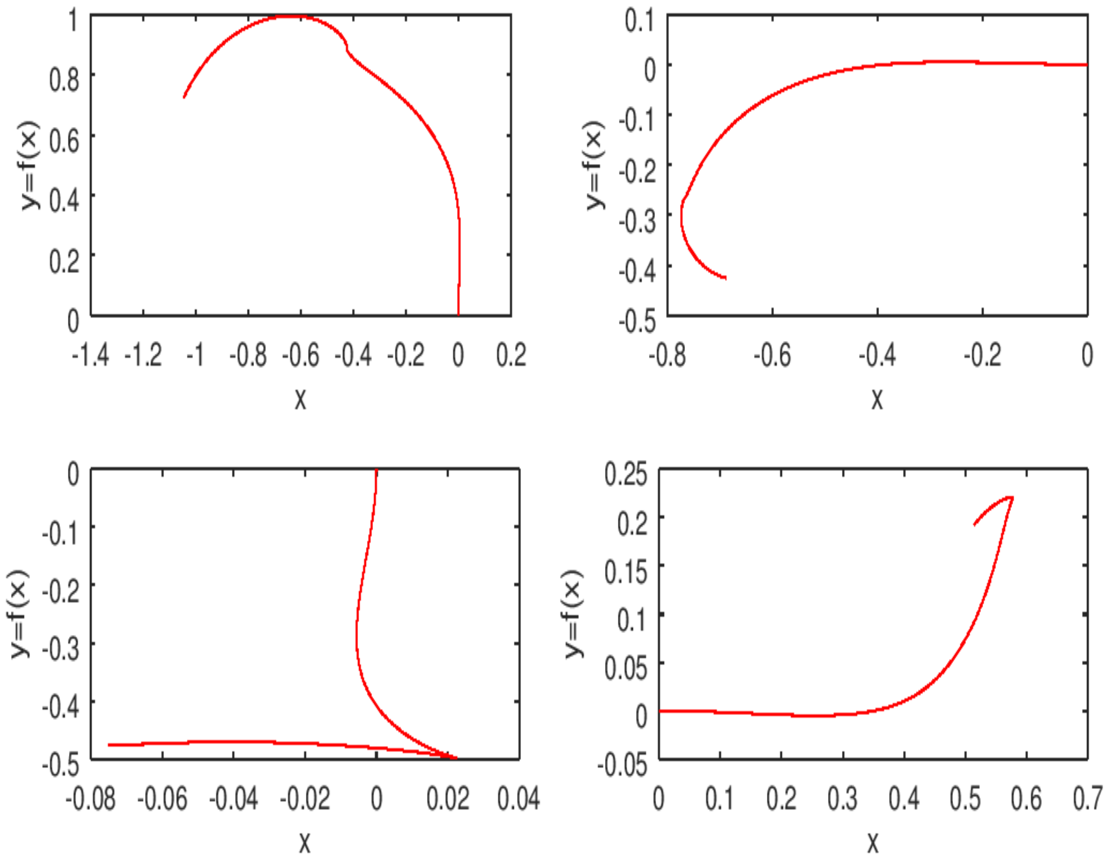

Figure 7. Feasible Robot Trajectories (tf=10 seconds).

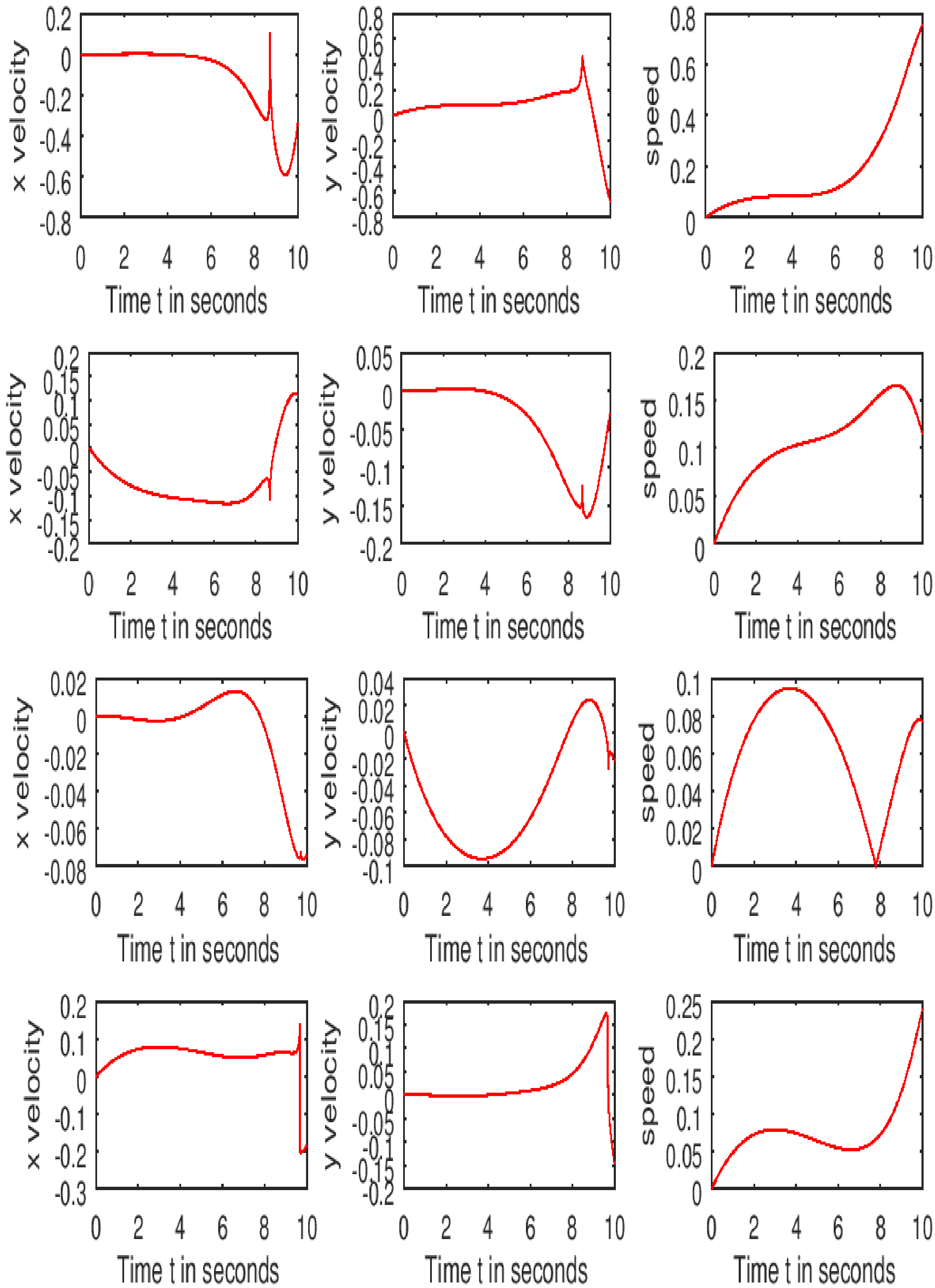

Figure 8. Feasible Velocities (tf=10 seconds).

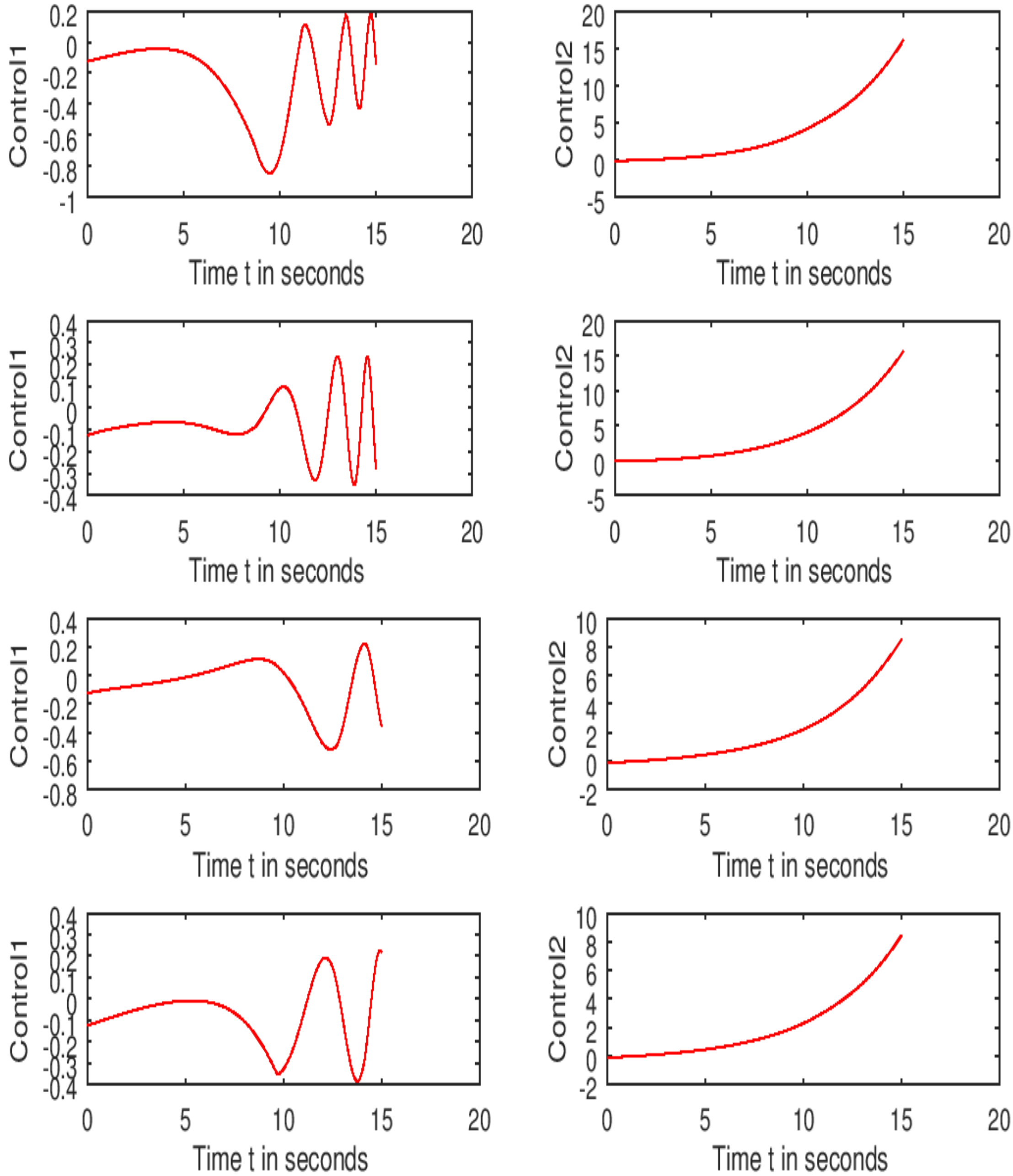

Figure 9. Feasible Control Strategies (tf=15 seconds).

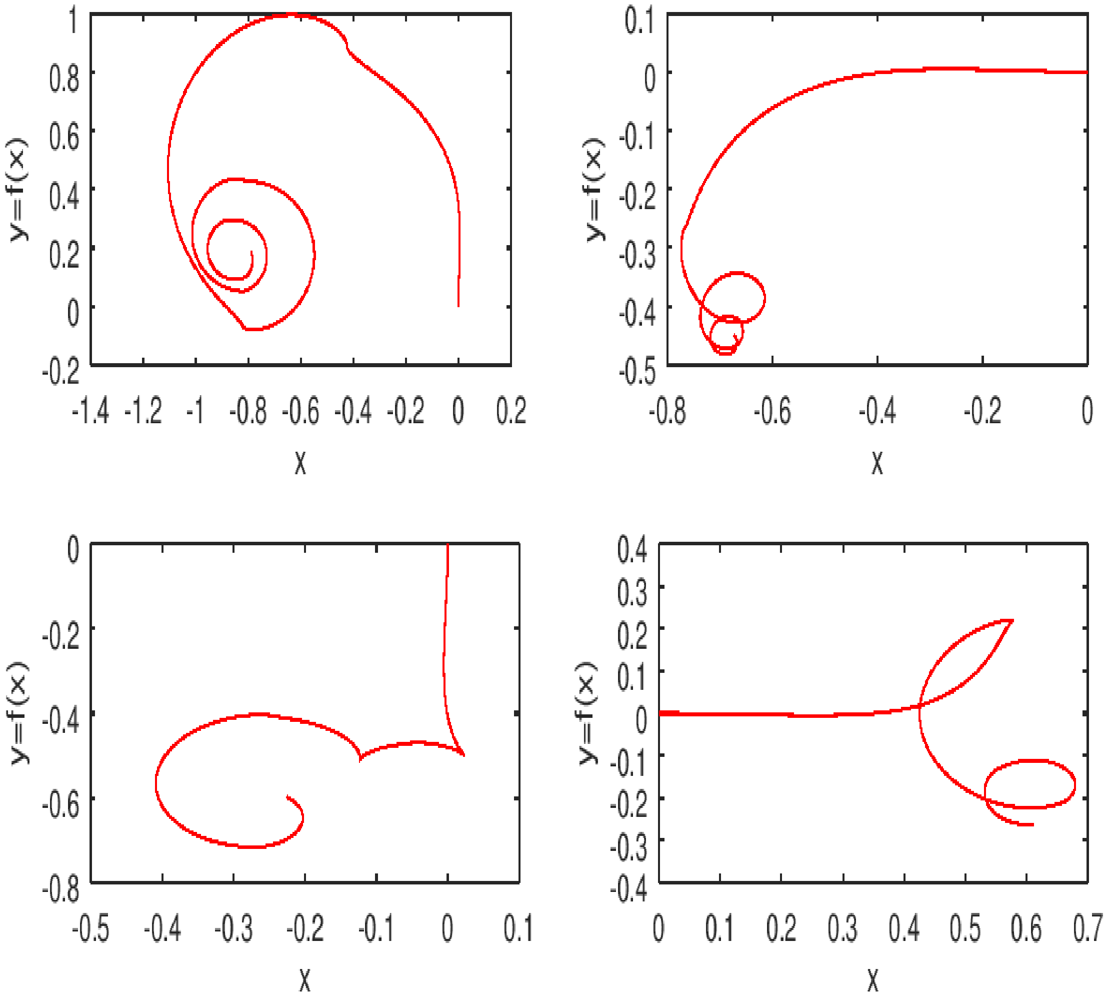

Figure 10. Feasible Robot Trajectories (tf=15 seconds).

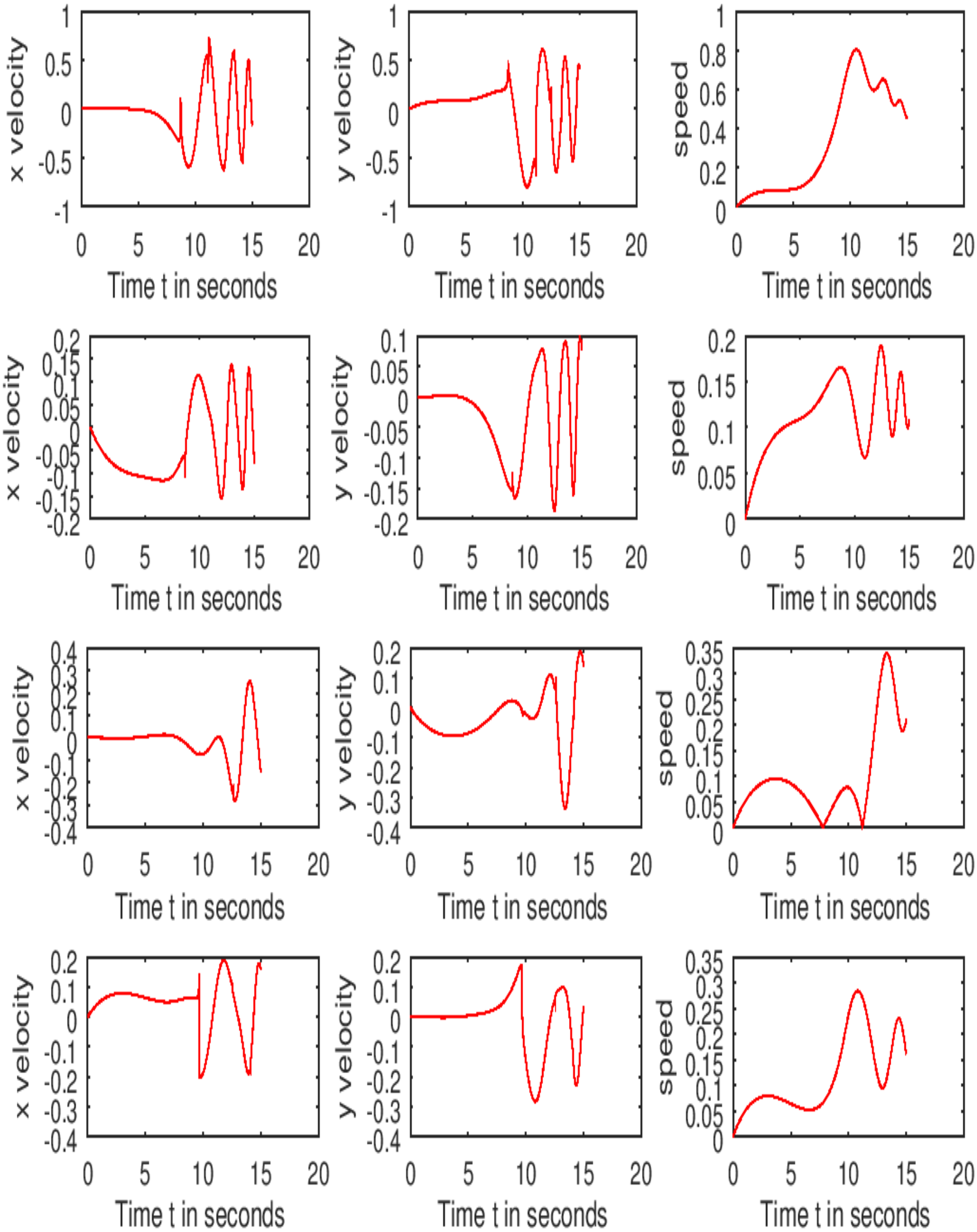

Figure 11. Feasible Velocities (tf=15 seconds).

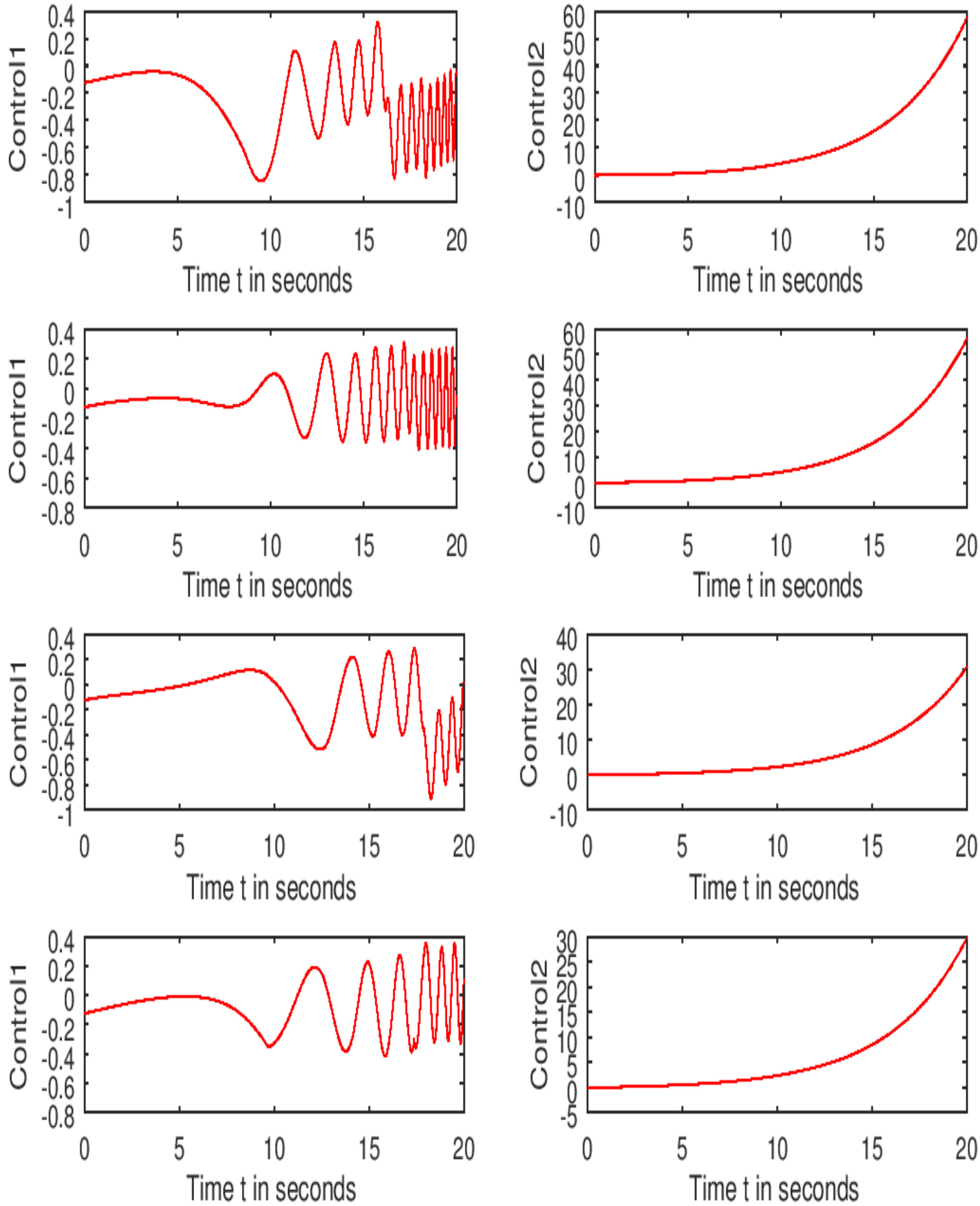

Figure 12. Feasible Control Strategies (tf=20 seconds).

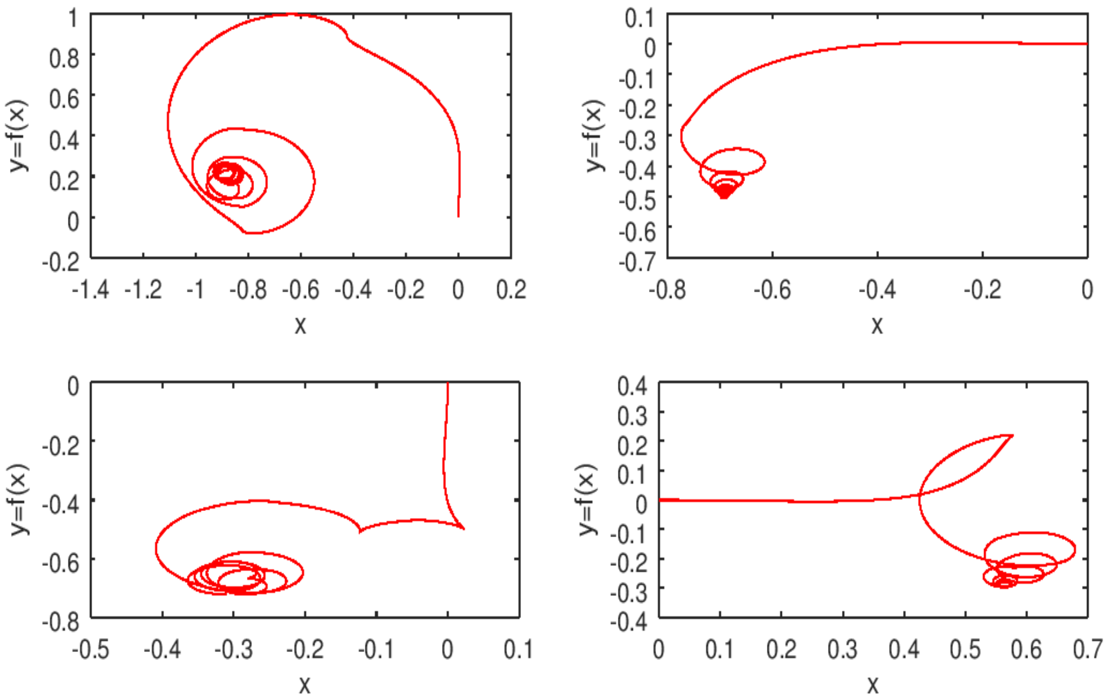

Figure 13. Feasible Robot Trajectories (tf=20 seconds).

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 31 August 2022 doi:10.20944/preprints202208.0533.v1

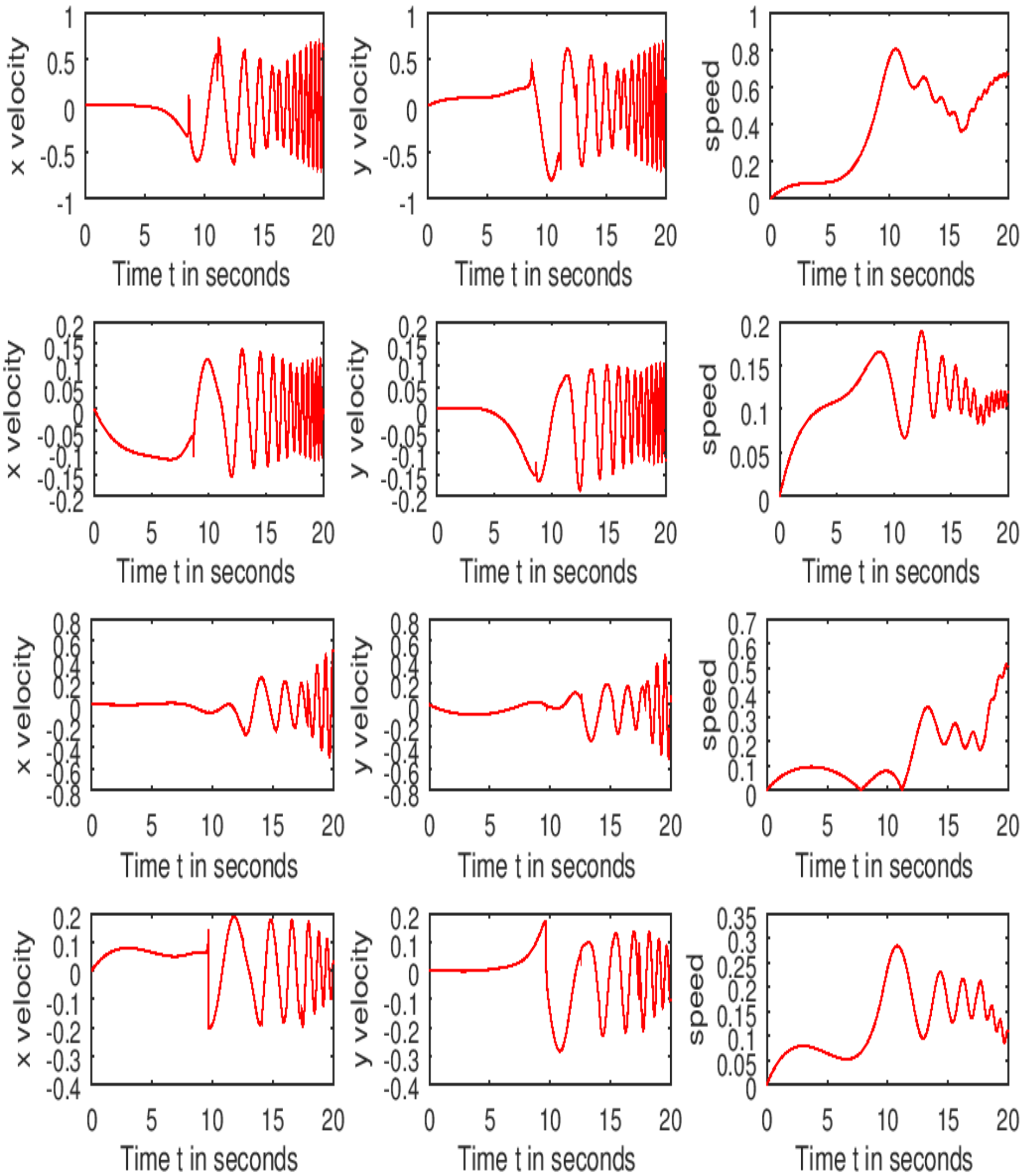

Figure 14. Feasible Velocities (tf=20 seconds).

The above graphs enable to study the performance of the bicycle Robot. Concerning the robot's trajectories, we have considered six different cases $(-\frac{\pi}{2}, 0, \frac{\pi}{2}, \pi)$ for the initial heading angle while the initial values for the other state variables remain unchanged. For each case of the final time, the feasible bicycle's trajectories (from top left to the bottom right) correspond respectively to the heading angles$(-\frac{\pi}{2}, 0, \frac{\pi}{2}, \pi)$. The same order is for the feasible control strategies.

6. Conclusion

The aim of this paper was to model and to control the kinematics of an autonomous is a bicycle robot whose reference point is the center of gravity. The model was developed in terms of a nonlinear system of six ordinary differential equations. Pontryagin's Minimum Principle was used to derive the feasible control and the costate system of ordinary differential equations. A fourth-order Runge-Kutta numerical method was used to solve the combined state-costate system of ordinary differential equations. The results were presented in terms of graphs for better illustration and understanding. They enable to predict the performance of the autonomous bicycle robot so that it can be controlled accurately and efficiently. The computer programs are useful to any reader or any researcher who is familiar with programming to learn moreFurther work will consist of developing path following feedback control strategies.

Funding: This work was supported by the University of Johannesburg, Faculty of Engineering and Built Environment.

Acknowledgments: The author of this paper thanks the university of Johannesburg for financial support.

Conflicts of Interest: This research paper does not involve a conflict of interest.

References

1. Sangduck Lee, Woonchul Ham (2002), Self Stability Strategy in Tracking Control of Unmanned electric bicycle with mass balance, IEEE / RSJ International Conference on Intelligent Robots Systems 3, 2200-2205
2. Yonghua Huang, Qizheng Liao, Shimin Wei, Lei Guo (2010), Dynamic Modeling of a bicycle Robot with front-wheel drive based on Kane's method, The 2010 IEEE International Conference on Information and Automation, 758-764
3. Javier Alonso-Mora, Andreas Breitenmosee, Paul Beadsley, Roland Siegwart (2012), Reciprocal Collision AvoidanceMultiple car-like Robots, IEEE International Conference on Robotics and Automation, 360-366.
4. Anan Suebsomran, Balancing Control of bicycle robot (2012), 2012 IEEE International Conference on Cyber Technology Systems (CYBER), 69-73.
5. Philip Polack, Florent Altche, Brigitte dAndrea-Novel, Arnaud de la Fotelle (2017), 2017 IEEE Intelligent Vehicles Symposium (IV) 812-818.
6. Xiangjun Meng, Lei Guo, Qizheng Liao (2012), Analysis of Nonholonomic Constraints about a variable structure bicycle robot, 2012 IEEE International Conference on Automation and Logistics, 628-633.
7. Yanbin Liu, Chenhui Jia, Jianhai Han (2009), Dynamics Modelling of an Unmanned bicycle with parallel mechanism adjusting stability, 2009 International Conference on Mechatronics and Automation, 1601-1605.
8. Yonghua Huang, Qizheng Liao (2010), Shimin Wei, Lei Guo, Stable balancing Motion Analysis of a Bicycle Robot with Front-wheel drive based on Moment balance (2010), 2010 International Conference on Intelligent Computation technology and Automation3, 367-371.
9. Peter I Corke, Peter Ridley(2001), Steering kinematics for a center-articulated mobile robot, IEEE Transactions on Robotics and Automation 17(2), 215-218.