

# LYAPUNOV FUNCTION BASED CONTROL OF A TYPE-(1,1) ROBOT WITHOUT USING THE REFERENCE STEERING ANGLE

Subodh Mishra  
Truong Giang Vo

# 1 Introduction

A simplified figure of the type (1,1) wheeled mobile robot is presented below in Figure 1.1.  $a$  is the distance between the wheel centre of the fixed wheels and the centre of steerable wheel.  $\beta$  is the steering angle and  $\theta$  is the orientation of the  $R_m$  frame (also called the robot frame) with respect to the  $R_O$  frame (the world frame). The objective is to design a controller based on a Lyapunov function to make the robot follow a circular trajectory of radius 2 metre and angular velocity 0.5 radian per second. In Figure 1.1,  $R$  is the perpendicular distance of the instantaneous centre of rotation (ICR) from the centre (s) of the steerable wheel.

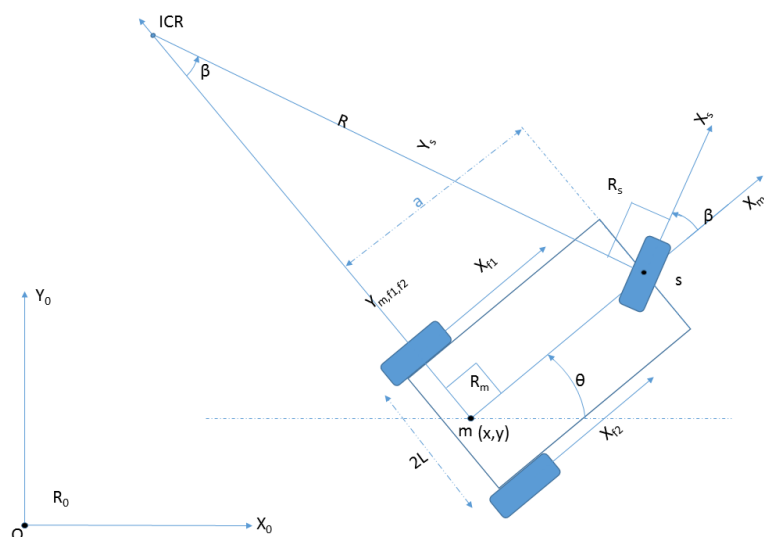


Figure 1.1: Schematic of Type 11 Robot

## 2 The Posture Kinematic Model

A possible kernel obtained from the matrix form of the no-skidding constraints for the type (1,1) wheeled mobile robot is:

$$\Sigma = \begin{bmatrix} 1 \\ 0 \\ \frac{\tan \beta}{a} \end{bmatrix}$$

Using the aforementioned kernel, the posture kinematic model of the type (1,1) robot is as follows:

$$\begin{aligned}\dot{x} &= a \cos \theta u_m \\ \dot{y} &= a \sin \theta u_m \\ \dot{\theta} &= \frac{\tan \beta}{a} u_m \\ \dot{\beta} &= u_s\end{aligned}$$

Here,  $u_m$  and  $u_s$  are the inputs to the Posture Kinematic Model. While  $u_s$  is the rate of change of orientation of the steering wheel,  $u_m$  is the net translational velocity of the robot.

### 3 Configuration Kinematic Model and Wheel Motorization

For the same kernel as discussed in the section 2, the complete Configuration Kinematic Model of the type (1,1) wheeled mobile robot is:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\beta} \\ \dot{\phi}_{f1} \\ \dot{\phi}_{f2} \\ \dot{\phi}_s \end{bmatrix} = \begin{bmatrix} a \cos \theta & 0 \\ a \sin \theta & 0 \\ \frac{\tan \beta}{a} & 0 \\ 0 & 1 \\ \frac{1}{r} + \frac{L \tan \beta}{ar} & 0 \\ \frac{1}{r} - \frac{L \tan \beta}{ar} & 0 \\ \frac{1}{r \cos \beta} & 0 \end{bmatrix} \begin{bmatrix} u_m \\ u_s \end{bmatrix}$$

Here  $\dot{\phi}_{f1}$ ,  $\dot{\phi}_{f2}$  and  $\dot{\phi}_s$  are the angular velocities of the fixed wheels and the steerable wheel about their corresponding spin axes and  $2L$  is the distance between the two fixed wheels (the origin of the robot frame,  $R_m$ , is at the centre of the common axle of the two fixed wheels).

#### 3.1 Wheel Motorisation

The degree of mobility of the type (1,1) wheeled mobile robot is  $\delta m = 1$ . Hence, only 1 out of the last three rows of the configuration kinematic models needs to be picked to complete the motorisation of this robot. Owing to mathematical simplicity and ease of inversion, the last row is chosen, which implies the spin of the steerable wheel about its y-axis is motorized. There is a singularity when  $\beta m = \frac{\pi}{2}$ , but this situation is highly unlikely and practically not allowed to occur. It should be noted that the steering joint (joint angle  $\beta$ ), which gives the rotation of the steerable wheel about its z-axis, is also motorized, but this does not come under wheel spin motorisation. The wheel spin motorisation obtained is:

$$\dot{\phi}_s = \frac{1}{r \cos \beta} u_m$$

The above equation is the inverse kinematic model.

### 4 Theoretical study

For the control design, let us consider two inputs,  $u_m$  and a new input  $u_\omega$ .  $u_\omega$  is actually the rate of change of orientation of the robot and in equation one can write:

$$\dot{\theta} = u_\omega = \frac{\tan \beta}{a} u_m$$

Using the above equation the steering angle can be found in terms of  $u_m$  and  $u_\omega$ .

$$\beta = \tan^{-1} \left( \frac{au_\omega}{u_m} \right)$$

For tracking, the point  $s$  is used. Point  $s$  is the origin of the steerable wheel frame( $R_s$ ,refer Figure 1.1). The orientation of the  $R_m$  frame with respect to the  $R_0$  frame is  $\theta$ . In the frame  $R_0$ , the posture to be tracked is:

$$\begin{aligned} x_s &= x + a \cos \theta \\ y_s &= y + a \sin \theta \\ \theta &= \theta \end{aligned}$$

Here, (x,y) is the coordinate of the point  $m$  in the  $R_0$  frame. On taking the time derivative of the equations above, the following equation are obtained:

$$\begin{aligned} \dot{x}_s &= \dot{x} - a\dot{\theta} \sin \theta \\ \dot{y}_s &= \dot{y} + a\dot{\theta} \cos \theta \\ \dot{\theta} &= \dot{\theta} \end{aligned}$$

Using the the posture kinematic model of section 2, the above equations can be written as:

$$\begin{aligned} {}^0\dot{x}_s &= \cos \theta u_m - a \sin \theta u_\omega \\ {}^0\dot{y}_s &= \sin \theta u_m + a \cos \theta u_\omega \\ \dot{\theta} &= u_\omega \end{aligned}$$

$\Rightarrow$

$$\begin{bmatrix} {}^0\dot{x}_s \\ {}^0\dot{y}_s \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta & -a \sin \theta \\ \sin \theta & a \cos \theta \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_m \\ u_\omega \end{bmatrix}$$

This is the rate of change of state variables of the actual robot. Similarly, the virtual robot (the reference), which needs to be tracked (or followed) can be described by the following equations:

$$\begin{bmatrix} {}^0\dot{x}_{sr} \\ {}^0\dot{y}_{sr} \\ \dot{\theta}_r \end{bmatrix} = \begin{bmatrix} \cos \theta_r & -a \sin \theta_r \\ \sin \theta_r & a \cos \theta_r \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_{mr} \\ u_{\omega r} \end{bmatrix}$$

All the state space equations developed above are with respect to the frame  $R_0$ .

Let error be defined as:  $error = reference - actual$ . In the  $R_m$  frame, the errors in rate of change of  $x, y$  and  $\theta$  is:

$$\begin{bmatrix} {}^m\dot{x}_e \\ {}^m\dot{y}_e \\ \dot{\theta}_e \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -\dot{\theta} \end{bmatrix} \times \begin{bmatrix} {}^m x_e \\ {}^m y_e \\ \theta \end{bmatrix} + {}^m \Omega_0(\theta) \left[ \begin{bmatrix} {}^0\dot{x}_{sr} \\ {}^0\dot{y}_{sr} \\ \dot{\theta}_r \end{bmatrix} - \begin{bmatrix} {}^0\dot{x}_s \\ {}^0\dot{y}_s \\ \dot{\theta} \end{bmatrix} \right]$$

Here,

$${}^m \Omega_0(\theta) = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Using the equations developed in this section, the error model is simplified as:

$$\begin{bmatrix} {}^m\dot{x}_e \\ {}^m\dot{y}_e \\ \dot{\theta}_e \end{bmatrix} = \begin{bmatrix} \cos \theta_e u_{mr} - a \sin \theta_e u_{\omega r} \\ \sin \theta_e u_{mr} + a \cos \theta_e u_{\omega r} \\ u_{\omega r} \end{bmatrix} + \begin{bmatrix} -1 & {}^m y_e \\ 0 & -(a + {}^m x_e) \\ 0 & -1 \end{bmatrix} \begin{bmatrix} u_m \\ u_\omega \end{bmatrix}$$

In short, the error model can be written as:  $\dot{X} = f(X, \mathbf{u})$  Here,  $X$  is the state variable vector and  $\mathbf{u}$  is the control input vector. Consider a Lyapunov function  $V(X)$  for the system  $\dot{X} = f(X, \mathbf{u})$ . Designing a Lyapunov controller means finding a control input vector  $\mathbf{u}$  to satisfy:

$$\dot{V}(X) = \frac{\partial V}{\partial X} \dot{X} = \frac{\partial V}{\partial X} f(X, \mathbf{u}) < 0$$

The Lyapunov function chosen must be positive definite. For this system, the Lyapunov function chosen is:

$$V(X) = \frac{1}{2} ({}^s x_e + {}^s y_e + \frac{{}^s \psi_e}{K_y})$$

$$\dot{V}(X) = \frac{\partial V}{\partial X} \dot{X} = \frac{\partial V}{\partial X} f(X, \mathbf{u}) < 0$$

$\Rightarrow$

$$\begin{bmatrix} {}^s x_e & {}^s y_e & \frac{{}^s \psi_e}{K_y} \end{bmatrix} \begin{bmatrix} {}^s \dot{x}_e \\ {}^s \dot{y}_e \\ \dot{\psi}_e \end{bmatrix} < 0$$

On simplifying:

$$(\cos \theta_e u_{mr} - a \sin \theta_e u_{\omega r} - u_{mr})^m x_e + \left( \frac{\sin \theta_e u_{mr} + a \cos \theta_e u_{\omega r}}{\theta_e}^m y_e + \frac{u_{\omega r}}{K_y} - \left( \frac{a^m y_e}{\theta_e} + \frac{1}{K_y} \right) u_\omega \right) \theta_e < 0$$

This inequality is satisfied if:

$$\begin{aligned} \cos \theta_e u_{mr} - a \sin \theta_e u_{\omega r} - u_m &= -K_x^m x_e \\ \left( \frac{\sin \theta_e u_{mr} + a \cos \theta_e u_{\omega r}}{\theta_e}^m y_e + \frac{u_{\omega r}}{K_y} - \left( \frac{a^m y_e}{\theta_e} + \frac{1}{K_y} \right) u_\omega \right) &= -\frac{K_\theta}{K_y} \theta_e \end{aligned}$$

$\Rightarrow$

$$\begin{aligned} u_m &= \cos \theta_e u_{mr} - a \sin \theta_e u_{\omega r} + K_x^m x_e \\ u_\omega &= \frac{K_\theta \theta_e^2 + u_{\omega r} \theta_e + K_y^m y_e (\sin \theta_e u_{mr} + a \cos \theta_e u_{\omega r})}{K_y^m y_e a + \theta_e} \end{aligned}$$

The above equations yield the value of the control inputs.

Hence,

$$\mathbf{u} = \begin{bmatrix} \cos \theta_e u_{mr} - a \sin \theta_e u_{\omega r} + K_x^m x_e \\ \frac{K_\theta \theta_e^2 + u_{\omega r} \theta_e + K_y^m y_e (\sin \theta_e u_{mr} + a \cos \theta_e u_{\omega r})}{K_y^m y_e a + \theta_e} \end{bmatrix}$$

The control objective is to force the robot to move along a circular trajectory centred at (0,0) with radius  $R = 2$  m and angular velocity of  $\omega_d = 0.5$  rad/s, this implies that the tangential velocity of the robot must be 1 m/s.

The translational velocity of the type (1,1) robot is  $\dot{x}^2 + \dot{y}^2 = u_m$  must be equal to the tangential velocity required to satisfy the control objective. If the tangential velocity is  $V_{tangential}$  then:

$$\dot{x}^2 + \dot{y}^2 = u_m = V_{tangential}$$

For the control objective specified,  $V_{tangential} = R\omega_d = 1$  m/s.

## 5 Modelling

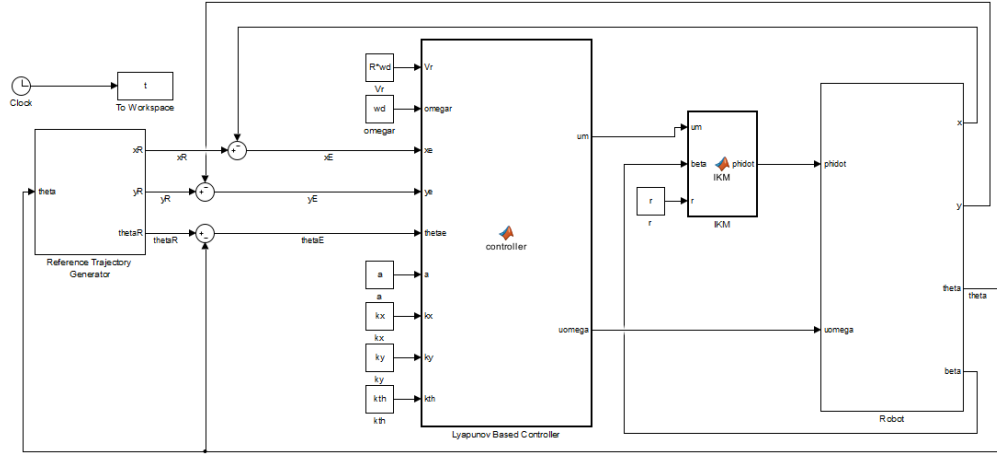


Figure 5.1: Simulink Model of Lyapunov Controlled Type (1,1) Robot

Figure 5.1 presents the `simulink` model of the Lyapunov function controlled Type(1,1) robot. A brief description of all the subsystems is presented below:

- The **Reference Trajectory Generator** subsystem generates the desired trajectory, in the  $R_m$  frame, that the robot is expected to follow.
- The **Lyapunov Based Controller** subsystem implements the Lyapunov control law deduced in the previous section. It takes the errors in position (expressed in  $R_s$  frame) & orientations, the reference tangential and angular velocities and the controller gains as input.
- The **Robot** block takes as input the steering wheel spin velocity  $\dot{\phi}$  and the vehicle angular velocity  $u_\omega$  as input. The subsystem implements the direct kinematic model and the posture kinematic model for the type (1,1) robot. There is a **Localisation** block inside this subsystem which shifts the tracking point to point  $m$ . Then a frame transformation is performed and the subsystem outputs the posture of the robot in the  $R_m$  frame.

### 5.1 Validation and Simulation

**Remark:** The fixed `step ode4` Runge Kutta solver is used with a time step of 0.001 second.

**Initial Conditions:** The initial posture of the origin of the robot frame is  $(2.3, 0, \pi)$ .

**Tuning:**  $K_x, K_y$  and  $K_\theta$

To ensure  $\dot{V} < 0$ ,  $K_x, K_y$  and  $K_\theta$  must all be positive.

- Tuning with  $K_x = 1$ ,  $K_y = 1$  and  $K_\theta = 1$ :

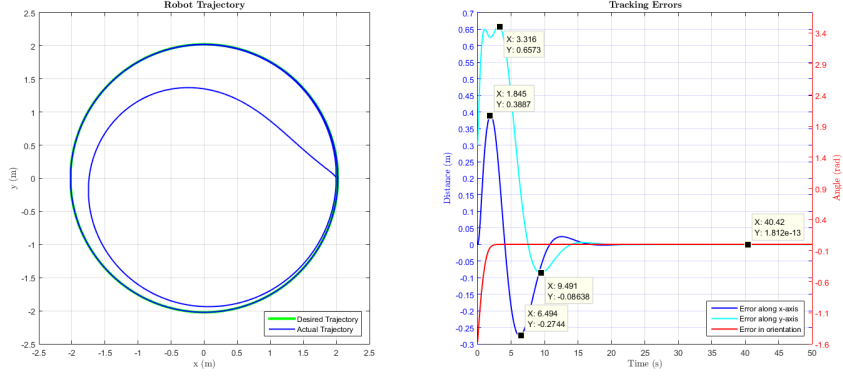


Figure 5.2: Robot Trajectory and errors in  $R_0$  frame with  $K_x = 1$ ,  $K_y = 1$  and  $K_\theta = 1$

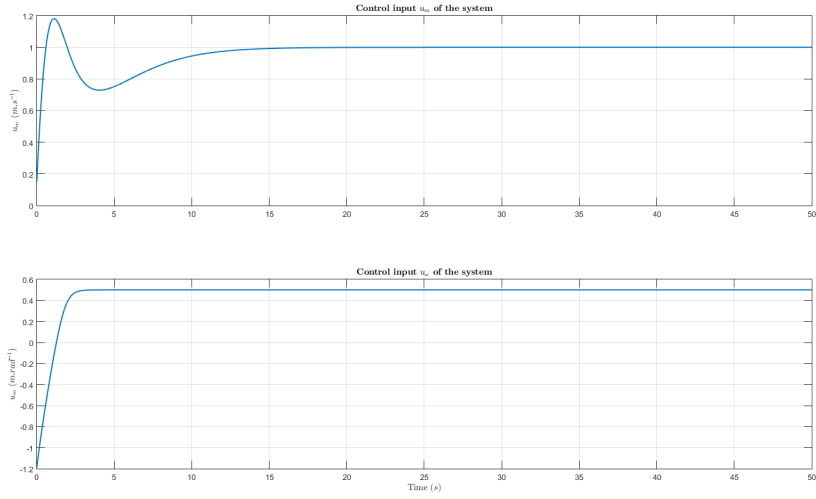


Figure 5.3: Control inputs with  $K_x = 1$ ,  $K_y = 1$  and  $K_\theta = 1$

With these gains, the settling time is high and the transients are more pronounced (Figure 5.2). This is evident from the high overshoots and undershoots in errors of both  $x$  and  $y$  axes, but error in  $\theta$  goes to zero rapidly. Though all the errors settle down to zero after approximately 15 seconds, increasing the gain may reduce the transients.

- Tuning with  $K_x = 1$ ,  $K_y = 5$  and  $K_\theta = 6$ :

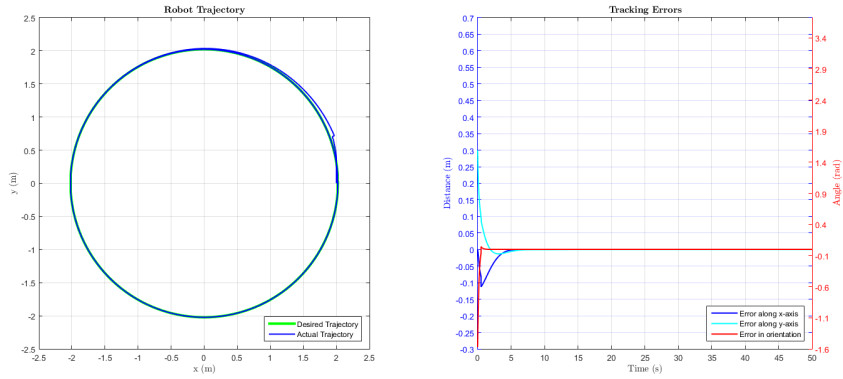


Figure 5.4: Robot Trajectory and errors in  $R_0$  frame with  $K_x = 1$ ,  $K_y = 5$  and  $K_\theta = 6$

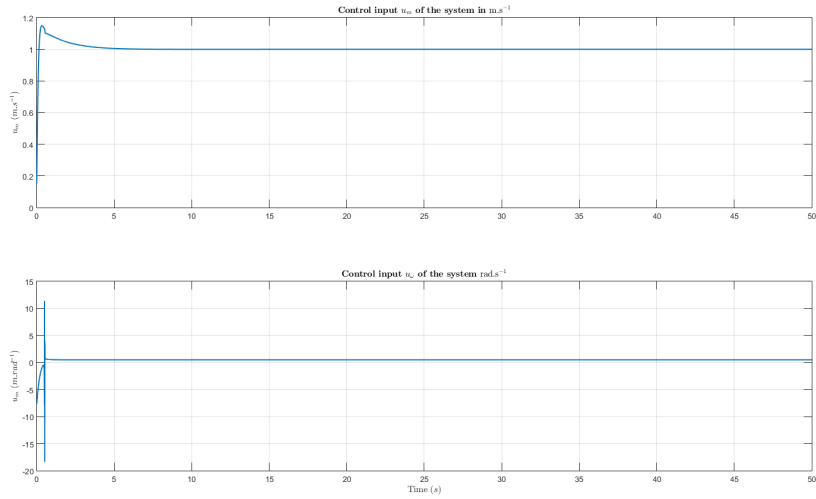


Figure 5.5: Control inputs with  $K_x = 1$ ,  $K_y = 5$  and  $K_\theta = 6$

In Figure 5.4 the effect of increased gains is presented. The settling time has reduced significantly and the undershoots and overshoots (the transients) are less pronounced. All the errors converge to zero in approximately 7 seconds. In Figure 5.5 the control signals for these gains are presented. As the gain is higher, the control input during 0.45 to 0.53 seconds is a little aggressive but this in practice will not affect a mechanical system due to its inertia.

### Effect of Parameter Variation

- **Case 1:** 10% increase in value of  $\mathbf{a}$  only

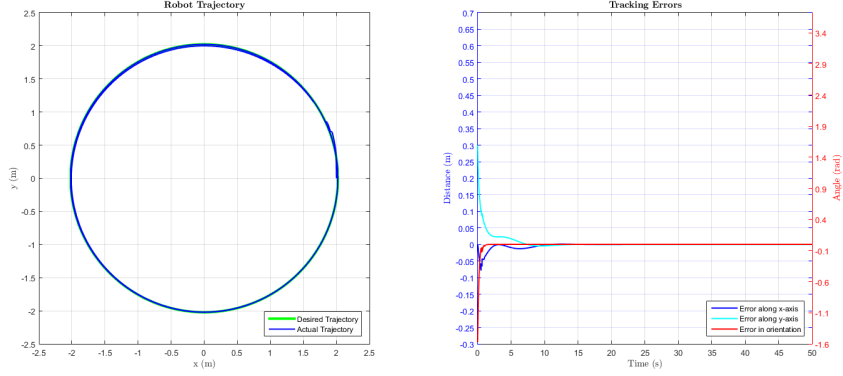


Figure 5.6: Robot Trajectory and errors in  $R_0$  frame with  $K_x = 1$ ,  $K_y = 5$  and  $K_\theta = 6$  and 10% increase in value of a

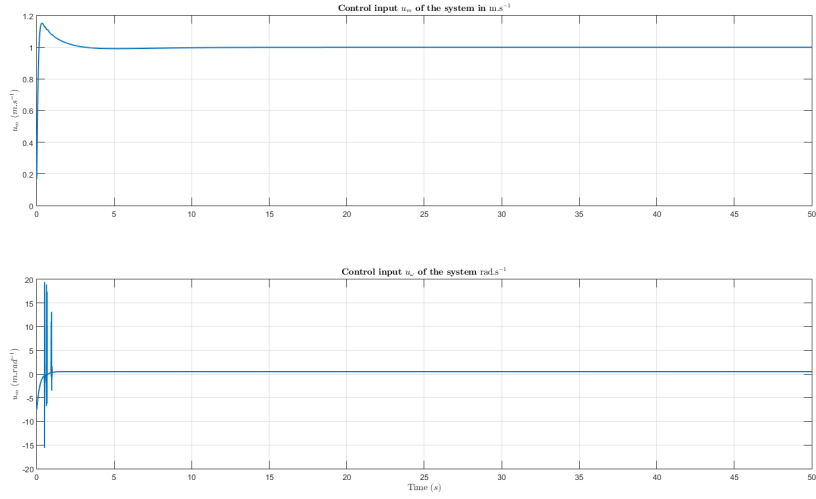


Figure 5.7: Control inputs with  $K_x = 1$ ,  $K_y = 5$  and  $K_\theta = 6$  and 10% increase in value of a

In Figure 5.6 the effect of parameter variation can be observed, the settling time has increased (approximately 15 seconds) but all the errors go down to zero. In Figure 5.7 it can be seen that if the parameter changes a little then the control signal also changes, there is a lot of chatter in the value of  $u_\omega$  in the first 1.5 seconds but gradually it settles down to the desired angular velocity of the vehicle. It is observed (Figure 5.8 and Figure 5.9) that reducing  $K_y$  and  $K_\theta$  to  $K_y = 4$  and  $K_\theta = 5$  removes all the rapid fluctuations in value of  $u_\omega$  but there is no significant change in the settling time, nevertheless, all the errors converge to zero eventually.



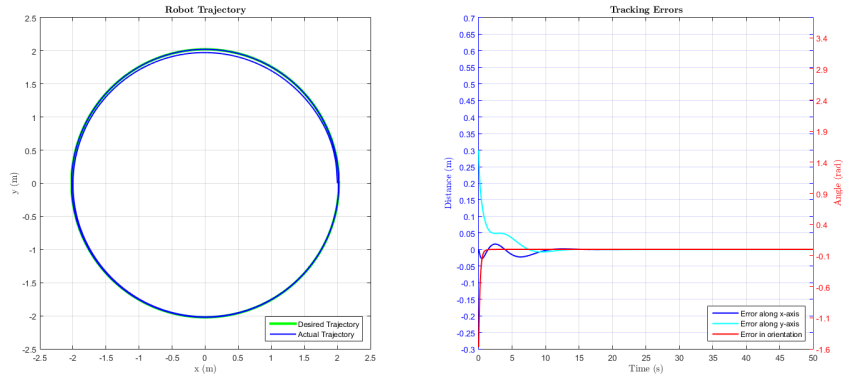


Figure 5.8: Robot Trajectory and errors in  $R_0$  frame with  $K_x = 1$ ,  $K_y = 4$  and  $K_\theta = 5$  and 10% increase in value of a

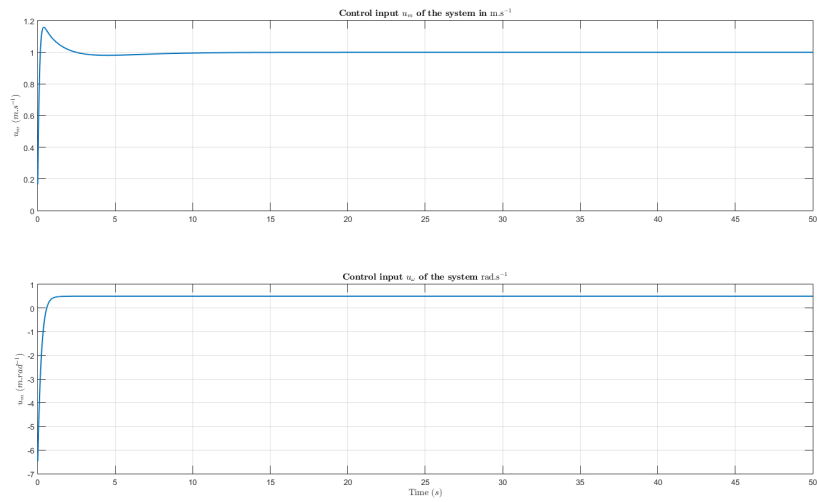


Figure 5.9: Control inputs with  $K_x = 1$ ,  $K_y = 4$  and  $K_\theta = 5$  and 10% increase in value of a

- **Case 2:** 10% increase in value of  $\mathbf{r}$  only

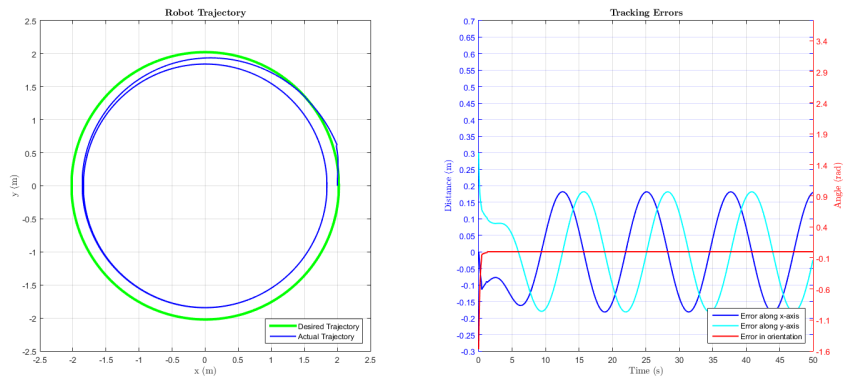


Figure 5.10: Robot Trajectory and errors in  $R_0$  frame with  $K_x = 1$ ,  $K_y = 5$  and  $K_\theta = 6$  and 10% increase in value of  $\mathbf{r}$

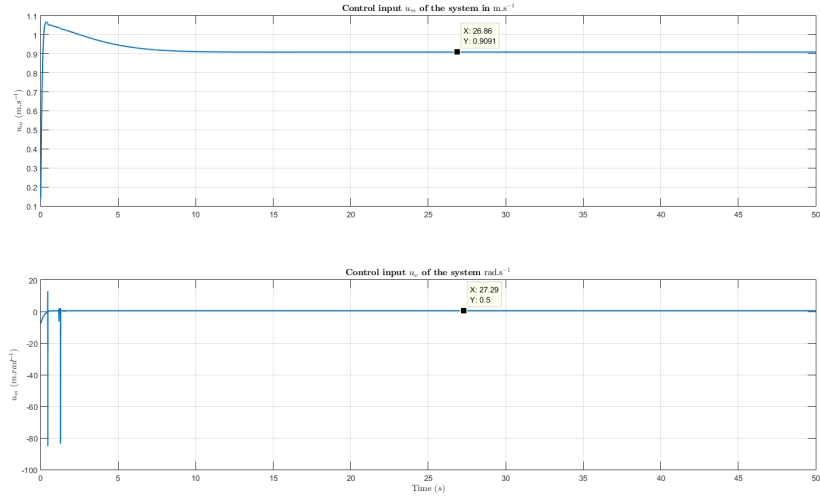


Figure 5.11: Control inputs with  $K_x = 1$ ,  $K_y = 5$  and  $K_\theta = 6$  and 10% increase in value of  $r$

In Figures 5.10 and 5.11 the effect of variation of the wheel radius  $r$  is seen. The effects are severe. The trajectory converges to a path of radius smaller than the required radius, thus explaining the oscillating errors in global  $x$  and  $y$  coordinates, but error in  $\theta$  converges to zero.

For any reasonable value of gain we take, the trajectory will converge to the path of same radius as seen in Figure 5.10 and 5.12 which is lesser than the required radius.

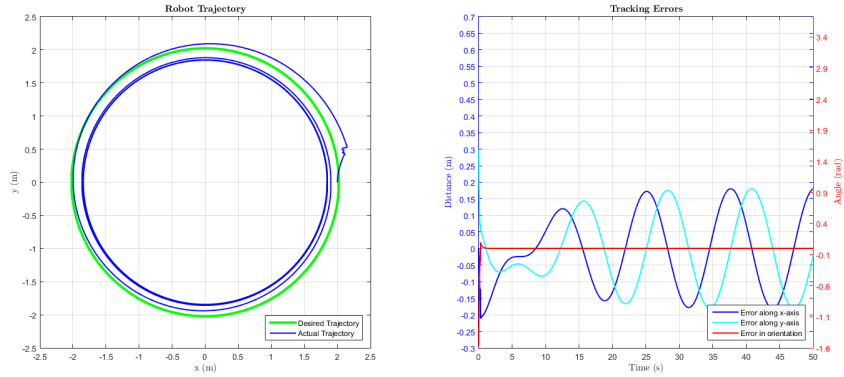


Figure 5.12: Robot Trajectory and errors in  $R_0$  frame with  $K_x = 2$ ,  $K_y = 5$  and  $K_\theta = 10$  and 10% increase in value of  $r$

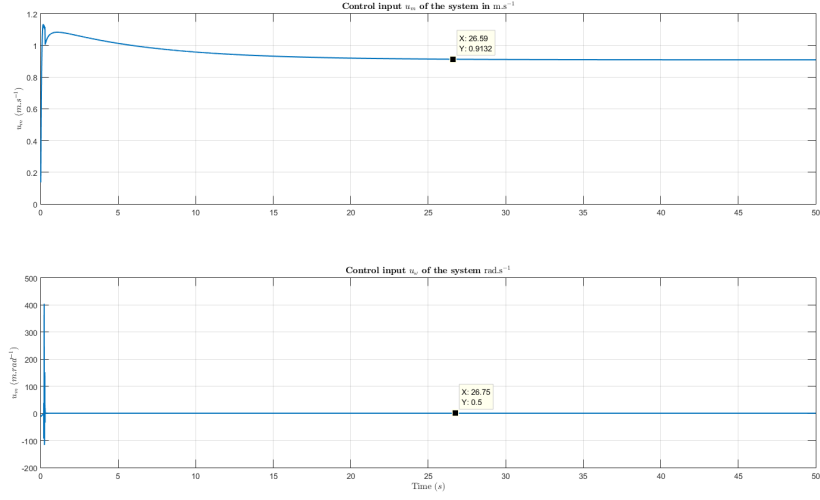


Figure 5.13: Control inputs with  $K_x = 2$ ,  $K_y = 5$  and  $K_\theta = 10$  and 10% increase in value of  $r$

The cause can be found in Figure 5.11 and 5.13 . The translational velocity  $u_m$  converges to  $0.9 m.s^{-1}$  approximately (less than  $1 m.s^{-1}$ , the reference translational velocity ) and the angular velocity converges to  $0.5 rad.s^{-1}$  (equal to the reference angular velocity), using these values the radius of the path obtained is  $r = \frac{0.9}{0.5} = 1.8$  metres (less than the desired radius). This effect of parameter variation is not mitigated even on increasing the gains, a solution could be to add an error term to the control output  $u_m$ , doing so will account for the difference in  $u_{mr}$ (the required  $u_m$ ) and the actual  $u_m$  under parameter variation.

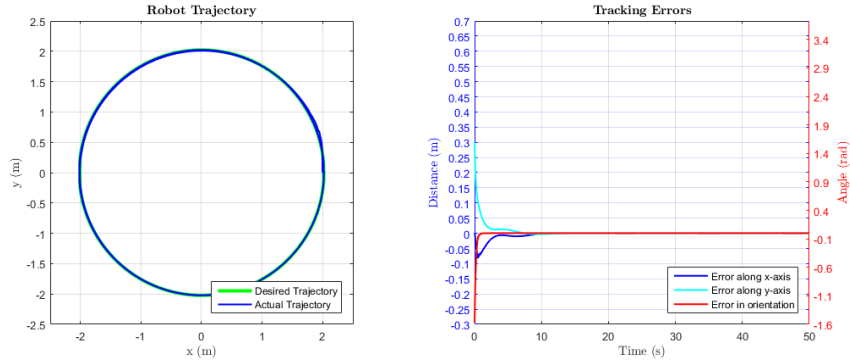


Figure 5.14: Robot Trajectory and errors in  $R_0$  frame with  $K_x = 1$ ,  $K_y = 5$  and  $K_\theta = 6$  and 10% increase in value of  $r$

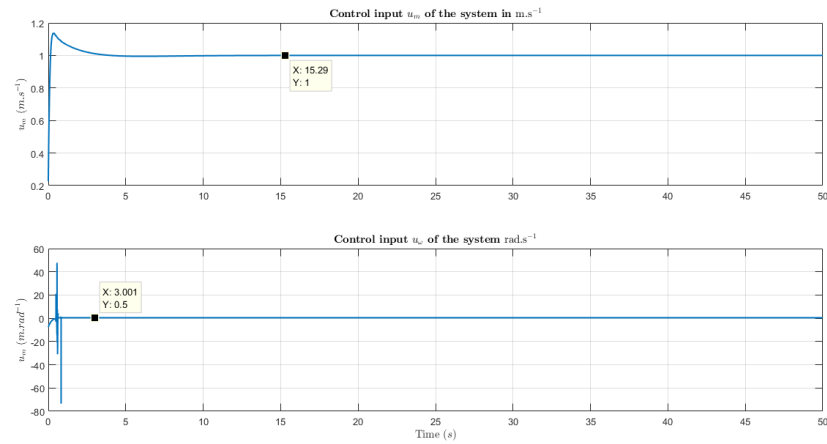


Figure 5.15: Control inputs with  $K_x = 1$ ,  $K_y = 5$  and  $K_\theta = 6$  and 10% increase in value of  $r$

The effect of adding  $u_{mr} - u_m (= 1 - 0.9 = 0.1$  for this case) to the controller output  $u_m$  is seen in Figure 5.14 and 5.15. Clearly, the response improves drastically. Thus the effect of parameter variation in this case can be mitigated by adding the error in the required control output and the actual control output.