Differential Drive Mobile Robot

December 17, 2022

1 Modelling

In this section, the dynamic modelling of the robot is described. The material is derived from this paper [1]. The robot configuration is depicted in Following figure:

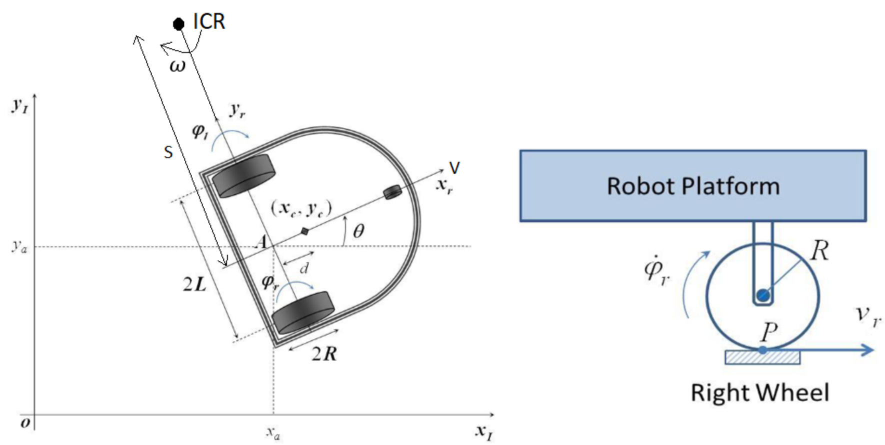

Figure 1: Caption

where $\{X_I, Y_I\}$ is the global frame fixed in the environment (inertial frame), $\{X_r, Y_r\}$ the frame attached to the robot (robot frame), $q^I = \begin{bmatrix} x_a \\ y_a \\ \theta \end{bmatrix}$ robot pose in the inertial frame. The pose of any point in the robot frame ($X^r = \begin{bmatrix} x^r \\ y^r \\ \theta^r \end{bmatrix}$)

and the inertial frame ($X^I = \begin{bmatrix} x^I \\ y^I \\ \theta^I \end{bmatrix}$) are related as

$$X^I = R(\theta)X^r = \begin{bmatrix} cos\theta & -sin\theta & 0 \\ sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ \dot{X}^I = R(\theta)\dot{X}^r \tag{1}$$

1.1 Kinematic Model

The motion of the robot is characterized by the following two non-holonomic constraints:

- No lateral slip: The robot only moves forward and backward not sideward:

$$\dot{y_a^r} = 0 \rightarrow -\dot{x}_a^I sin\theta + \dot{y}_a^I cos\theta = 0 \tag{2}$$

- Pure rolling constraint: Each wheel has one contact with the ground (no slipping or skidding). Therefore, the robot always rotates around a point referred as instantaneous center of rotation (ICR) by the angular velocity ω:

$$\omega(S+L) = v_r \\ \omega(S-L) = v_l \tag{3}$$

  Solving Eq. 3 for ω and S:

$$\omega = (v_r - v_l)/2L \\ S = L\frac{v_r+v_l}{v_r-v_l} \tag{4}$$

  Therefore the velocity of the robot is

$$V = \omega S = \frac{v_r + v_l}{2} \tag{5}$$

By combining Equations 4, 5, and 2, the robot pose in the robot frame is

$$\begin{bmatrix} \dot{x_a^r} \\ \dot{y_a^r} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \frac{R}{2} & \frac{R}{2} \\ 0 & 0 \\ \frac{R}{2L} & -\frac{R}{2L} \end{bmatrix} \begin{bmatrix} \dot{\phi_r} \\ \dot{\phi_l} \end{bmatrix} \tag{6}$$

The robot pose in the inertial frame is

$$\begin{bmatrix} \dot{x_a^I} \\ \dot{y_a^I} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \frac{R}{2}cos\theta & \frac{R}{2}cos\theta \\ \frac{R}{2}sin\theta & \frac{R}{2}sin\theta \\ \frac{R}{2L} & -\frac{R}{2L} \end{bmatrix} \begin{bmatrix} \dot{\phi_r} \\ \dot{\phi_l} \end{bmatrix} \tag{7}$$

1.2 Dynamic Model

There are three main approaches for modelling the dynamics of a robot: Lagrangian, Newton-Euler, and Kane's method. In this project, we utilize Lagrangian method to find the model. The generalized coordinates are selected as

$$q = \begin{bmatrix} x_a \\ y_a \\ \theta \\ \phi_r \\ \phi_l \end{bmatrix} \tag{8}$$

The kinetic energy of the robot is

$$\begin{gathered} T = T_c + T_{wr} + T_{wl} = \\ \tfrac{1}{2}m_c v_c^2 + \tfrac{1}{2}I_c\dot{\theta}^2 + \tfrac{1}{2}m_w v_r^2 + \tfrac{1}{2}I_m\dot{\theta}^2 + \tfrac{1}{2}I_w\dot{\phi_r}^2 + \tfrac{1}{2}m_w v_l^2 + \tfrac{1}{2}I_m\dot{\theta}^2 + \tfrac{1}{2}I_w\dot{\phi_l}^2 \\ T = \tfrac{1}{2}m(\dot{x_a}^2 + \dot{y_a}^2) - m_c d\dot{\theta}(\dot{y_a}cos\theta - \dot{x_a}sin\theta) + \tfrac{1}{2}I_w(\dot{\phi_r}^2 + \dot{\phi_l}^2) + \tfrac{1}{2}I\dot{\theta}^2 \\ m = m_c + 2m_w \\ I = I_c + m_c d^2 + 2m_w L^2 + 2I_m \end{gathered} \tag{9}$$

where m_c and I_c are the robot's mass without wheels and moment of inertia about the vertical axis through the CoM, m_w and I_w are the mass of each wheel and the moment of inertia about the wheel axis, and I_m is the wheel's moment of inertia about the wheel diameter. The dynamics of the system is

$$\begin{gathered} M(q)\dot{\eta} + C(q,\dot{q})\eta = B(q)\tau \\ M(q) = \begin{bmatrix} I_w + \frac{R^2}{4L^2}(mL^2 + I) & \frac{R^2}{4L^2}(mL^2 - I) \\ \frac{R^2}{4L^2}(mL^2 - I) & I_w + \frac{R^2}{4L^2}(mL^2 + I) \end{bmatrix} \\ C(q,\dot{q}) = \begin{bmatrix} 0 & \frac{R^2}{2L}m_c d\dot{\theta} \\ -\frac{R^2}{2L}m_c d\dot{\theta} & 0 \end{bmatrix} \\ B(q) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \end{gathered} \tag{10}$$

where $\eta = \begin{bmatrix} \phi_r \\ \phi_l \end{bmatrix}$ and $\tau = \begin{bmatrix} \tau_r \\ \tau_l \end{bmatrix}$.

1.3 Models for Simulation

For the simulation, the following model is considered

$$\begin{bmatrix} \dot{x_a} \\ \dot{y_a} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} cos\theta & 0 & 0 \\ sin\theta & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} v \\ \omega \\ \theta \end{bmatrix} \tag{11}$$

$$[\dot{v} \quad \dot{\omega} \quad th\dot{e}ta] = \begin{bmatrix} (m + \frac{2I_w}{R^2})^{-1}(m_c d\omega^2 + \frac{1}{R}(\tau_r + \tau_l)) \\ (I + \frac{2L^2 I_w}{R^2})^{-1}(-m_c d\omega v + \frac{L}{R}(\tau_r - \tau_l)) \\ \omega \end{bmatrix} \tag{12}$$

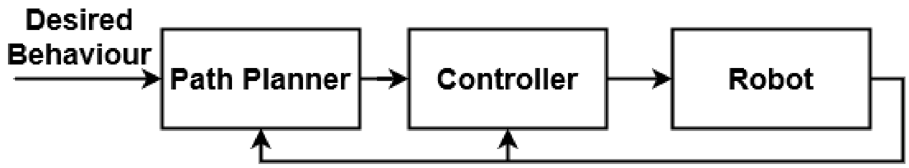

Figure 2: Control Architecture

2 Control

The control objective is to design a controller that ensures the convergence of the robot to a predefined path. The overall control architecture is depicted in Fig. 2.

2.1 Carrot-chasing

In this algorithm, the controller moves the robot toward a Virtual Target Point (VTP) named carrot. The position of the carrot is always at the line passing through the 2 waypoints a δ distance above the projected point of the robot current position on the line (see Fig. 3 taken from [2]).

The algorithm has the following steps:

- Initialize the waypoints $\rightarrow W_i = (x_i, y_i), W_{i+1} = (x_{i+1}, y_{i+1})$, the robot current position $p = (x_p, y_p)$, and the path parameter δ.
- Calculate the Line of Sight (LOS) angle $\rightarrow \theta = atan2((y_{i+1} - y_i, (x_{i+1} - x_i)))$
- Calculate the angle between p and $W_i \rightarrow \theta_u = atan2((y_p - y_i), (x_p - x_i))$
- Calculate the distance between p and $W_i \rightarrow R_u = \sqrt{(x_p - x_i)^2 + (y_p - y_i)^2}$
- Calculate the distance between p and the carrot $\rightarrow S_1 = \sqrt{\delta^2 + (R_u sin(\theta - \theta_u))^2}$
- Calculate the cross-track error angle $\rightarrow \zeta = arcsin(R_u sin(\theta - \theta_u)/S_1)$
- Calculate the desired heading angle $\rightarrow \psi_d = \zeta + \theta$
- Calculate the desired control output $\rightarrow \phi_x = k(\psi_d - \psi)$

For more details, please refer to [2].

References

[1] R. Dhaouadi and A. A. Hatab, "Dynamic modelling of differential-drive mobile robots using lagrange and newton-euler methodologies: A unified framework," *Advances in Robotics & Automation*, vol. 2, no. 2, pp. 1–7, 2013.

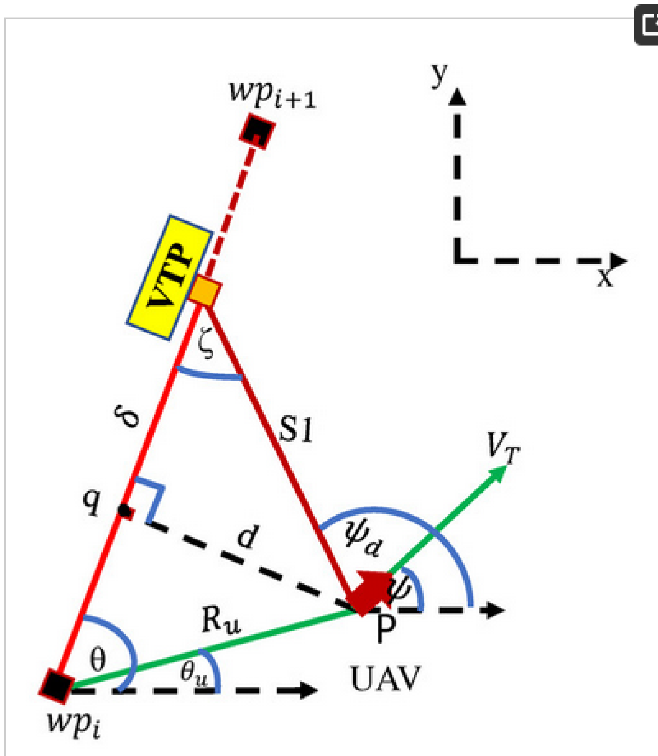

Figure 3: Carrot, waypoints, and the robot configuration.

[2] E. Safwat, W. Zhang, A. Mohsen, and M. Kassem, "Design and analysis of a robust uav flight guidance and control system based on a modified nonlinear dynamic inversion," *Applied Sciences*, vol. 9, no. 17, p. 3600, 2019.