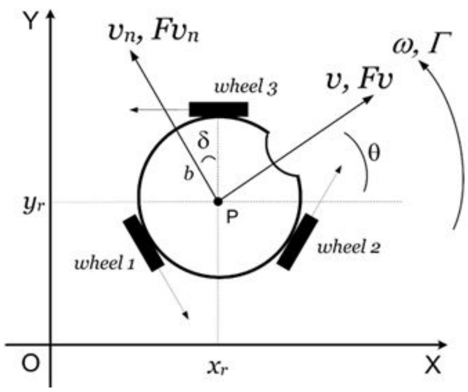
Pose Control of Three-Wheeled Omnidirectional Mobile Robot using Proportional Derivative - Super Twisting Sliding Model Control

This strategy employs a cascade control approach. The cascade control structure consists of an outer loop using a Proportional-Derivative (PD) controller to regulate the robot's velocity, ensuring it moves toward the desired position. Meanwhile, the inner loop utilizes the Super Twisting Sliding Mode Control (STSMC) method as a velocity controller, generating actuator voltage signals for each motor.

The design philosophy behind this controller is to leverage STSMC to handle the system's nonlinearities, external disturbances, and model uncertainties. Once a well-performing velocity control is achieved, implementing a PD controller for position control becomes straightforward.

Kinematics Model

schematic



Velocity transformation matrix

$$\dot{\xi} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \dot{\xi}_r$$

where

$$\xi = \begin{bmatrix} x_r & y_r & \theta \end{bmatrix}$$

- ξ is robot pose in inertial frame of reference
- ξ_r is the robot pose in non-inertial frame of reference (moving with the robot frame of reference)

State Space Model

$$\dot{X} = AX + BU + Ksign(X)$$

$$Y = CX$$

where

$$U = \begin{bmatrix} u_1(t) & u_2(t) & u_3(t) \end{bmatrix}^T \text{ and } Y = X = \begin{bmatrix} v(t) & v_n(t) & \omega(t) \end{bmatrix}^T$$

$$A = \begin{bmatrix} -\frac{3l^2K_t^2}{2MR_ar^2} - \frac{B_v}{M} & 0 & 0 \\ 0 & -\frac{3l^2K_t^2}{2MR_ar^2} - \frac{B_{vn}}{M} & 0 \\ 0 & 0 & -\frac{3b^2l^2K_t^2}{I_nR_ar^2} - \frac{B_w}{I_n} \end{bmatrix}$$

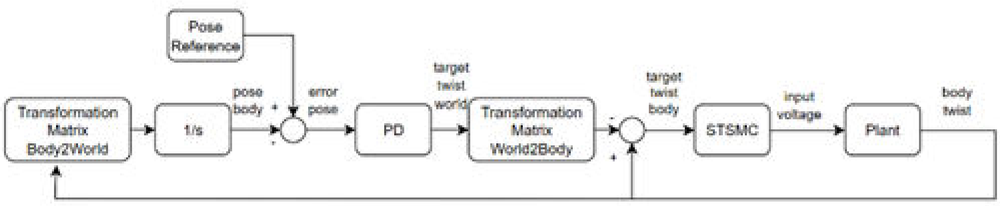
$$B = \frac{lK_t}{R_ar} \begin{bmatrix} 0 & \frac{\cos(\delta)}{M} & -\frac{\cos(\delta)}{M} \\ -\frac{1}{M} & \frac{\sin(\delta)}{M} & \frac{\sin(\delta)}{M} \\ \frac{b}{I_n} & \frac{b}{I_n} & \frac{b}{I_n} \end{bmatrix}; \quad C = I$$

$$K = \begin{bmatrix} -C_v & 0 & 0 \\ 0 & -\frac{C_{vn}}{M} & 0 \\ 0 & 0 & -\frac{C_w}{I_n} \end{bmatrix}$$

In this simulation we are going to use this for the parameters values

Simbol	*Description*	*Values*
$B_v(N/m/s)$	viscous friction coefficient related to v	0.94
$B_{v_n}(N/m/s)$	viscous friction coefficient related to v_n	0.96
$B_\omega(N/rad/s)$	viscous friction coefficient related to ω	0.01
$C_v(N)$	coulomb friction coefficient related to v	2.2
$C_{v_n}(N)$	coulomb friction coefficient related to v_n	1.5
$C_\omega(N.m)$	coulomb friction coefficient related to ω	0.099
$b(m)$	radius of the robot	0.1
$M(kg)$	mass of the robot	1.5
$I_n(kg.m^2)$	inertia moment of the robot	0.025
δ	angle	30^o
$r_1, r_2, r_3(m)$	radius of the wheels	0.035
l_1, l_2, l_3	reduction of the motors	19:1
$L_{a_{1...3}}(H)$	motor's armature inductance	0.00011
$R_{a_{1...3}}(\Omega)$	motor's armature resistance	1.69
$K_{v_{1...3}}(Volts/rad/s)$	motor's emf constant	0.0059
$K_{t_{1...3}}(N.m/A)$	motor's torque constant	0.0059

Control Strategy: Cascade Proportional Derivative - Super Twisting Sliding Model Control (STSMC)



Velocity Control (Inner Loop) using STSMC

define the sliding surface as follow

$$s = \begin{bmatrix} s_1 \\ s_2 \\ s_3 \end{bmatrix} = e(t) + \lambda \int e(t) dt$$

$$\dot{s} = \dot{e}(t) + \lambda e(t)$$

where

$$e(t) = x - x_r$$

$x = \begin{bmatrix} v & v_n & \omega \end{bmatrix}$ and x_r is the velocity reference signal

Define the reaching law using super twisting algorithm

$$\dot{s} = \phi_1(\mathrm{s}) + \upsilon$$

$$\phi_1(s) = \begin{bmatrix} -k_{11}\sqrt{|s_1|}\ sign(s_1) \\ -k_{12}\sqrt{|s_2|}\ sign(s_2) \\ -k_{13}\sqrt{|s_3|}\ sign(s_3) \end{bmatrix}$$

$$\dot{\upsilon} = \dot{\phi}_2(s) = \begin{bmatrix} -k_{21} sign(s_1) \\ -k_{22} sign(s_2) \\ -k_{23} sign(s_3) \end{bmatrix}$$

take a look at the robot dynamic

$$\dot{x} = Ax + Bu + Ksign(x)$$

with the above formulations, we can choose $u(t)$ as the following:

$$\dot{s} = \dot{e}(t) + \lambda e(t)$$

$$\dot{s} = \left(\dot{x} - \dot{x}_d\right) + \lambda(x - x_d)$$

$$\dot{s} = \left(Ax + Bu + Ksign(x) - \dot{x}_d\right) + \lambda(x - x_d)$$

then

$$u(t) = \ -B^{-1}\left((A + \lambda)x + Ksign(x) - \dot{x}_d - \lambda x_d + \phi_1(s) + \phi_2(s)\right)$$

and

$$\boldsymbol{u}(t) = \begin{cases} V_{max},\ \text{jika}\ u(t) > \ V_{max} \\ u(t),\ \text{jika}\ V_{min} \leq u(t) \leq V_{max} \\ V_{min},\ \text{jika}\ u(t) < V_{min} \end{cases}$$

that's conclude the inner loop which is just a velocity control of the mobile robot. After that, we design the outer loop to control the pose of the mobile robot using simple PD controller. We define error of pose

$$e_{inersia} = \xi_{ref} - \xi$$

$$\xi = \begin{bmatrix} x_r & y_r & \theta \end{bmatrix}$$

change frame of reference

$$e_{robot} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}^T e_{inersia}$$

then by that, we design the control input for velocity setpoint $x(t)$ as the following

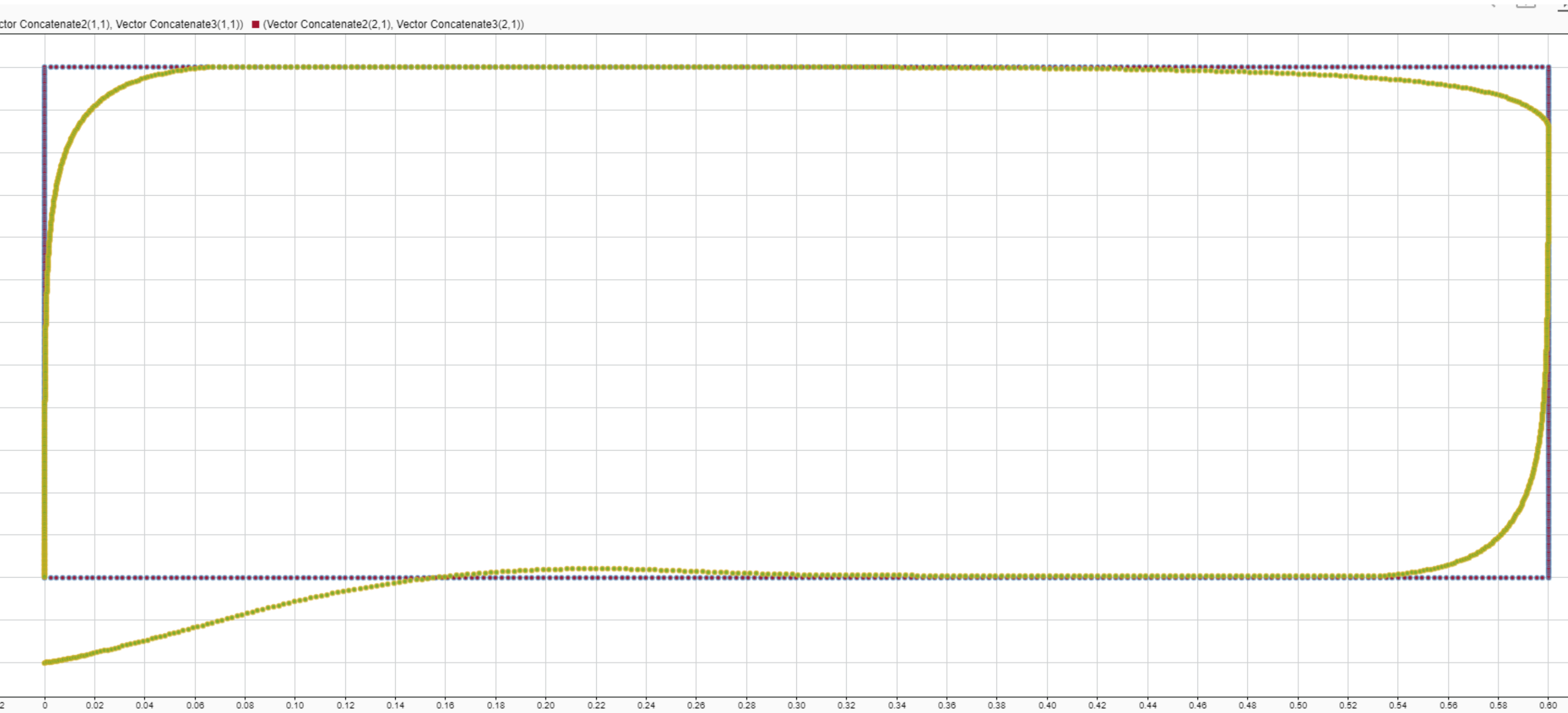
$$x(t) = K_p\, e_{robot} + K_d \frac{de_{robot}}{dt}$$

$$\boldsymbol{x}(t) = \left\{ \begin{matrix} x_{max}, \text{ jika } x(t) > x_{max} \\ x(t), \text{ jika } x_{min} \leq x(t) \leq x_{max} \\ x_{min}, \text{ jika } x(t) < x_{min} \end{matrix} \right)$$

with $x = \begin{bmatrix} v & v_n & \omega \end{bmatrix}$

Result

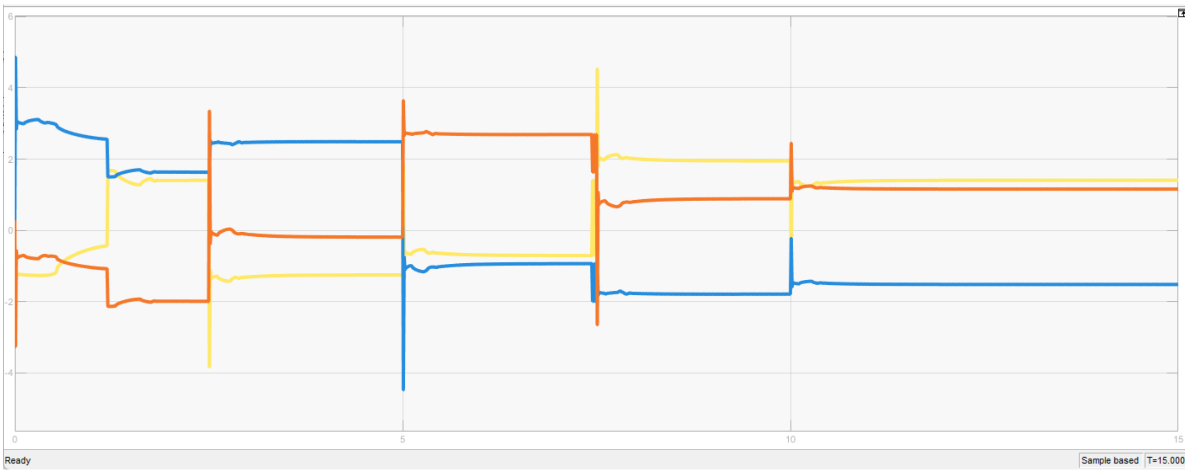
The x-y position graph of the robot (in meters). The graph below shows the robot's position compared to the given setpoint.



- The purple line is the setpoint
- The yellow line is the robot x and y position

The robot's initial position was intentionally given an offset of -0.1 on the y-axis, and the setpoint was designed to form a square trajectory with counterclockwise movement.

Control input signal (Voltage on each motor)



Simulink Simulation