Robot Control

EE468/CE468: Mobile Robotics

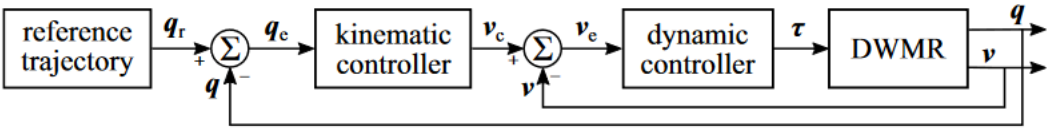
Dr. Basit Memon

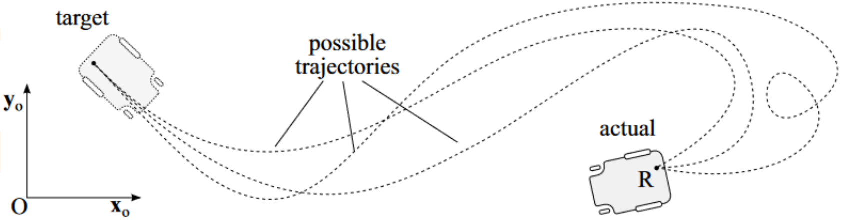
Electrical and Computer Engineering
Habib University

September 18, 2023

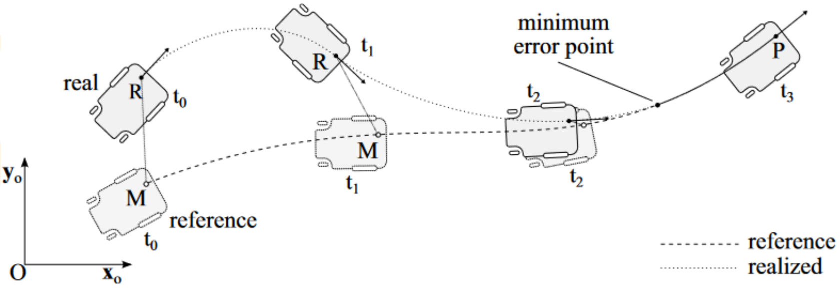
Table of Contents

Table of Contents


reference trajectory
q_r
+
q
−
q_e
kinematic controller
v_c
+
v
−
v_e
dynamic controller
τ
DWMR
q
v

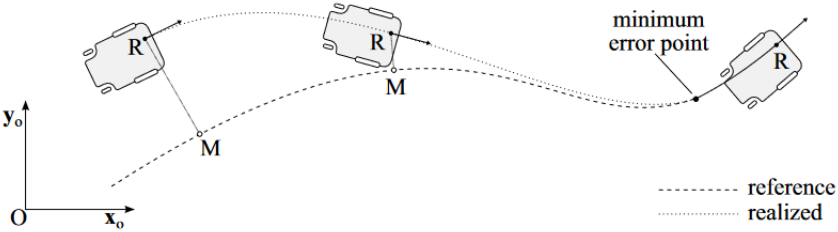


- Given a desired pose, $\xi = (x, y, \theta)$, the goal is to achieve the desired pose from the current position and orientation.
- Path followed or time taken are immaterial.



- Given a reference trajectory $(x_r(t), y_r(t), \theta_r(t))$, the goal is to bring the pose error $(x_r - x, y_r - y, \theta_r - \theta)$ to zero.

Three Common Robot Control Problems: Path Following



- Given a curve C on the plane, a nonzero velocity v_o for robot chassis, and a point P attached to the chassis, the goal is to have point P follow the curve C when robot moves with velocity v_o.

Table of Contents

Mathematical model of system for control

- Recall that continuous-time control requires a differential model of the system, which is called *dynamical system model* in control literature.
- The word *dynamical* here simply refers to time-evolving nature of system captured by differential equations.
- We can use kinematic model of robot (also differential equations) for control, and don't require model to include forces (dynamics model).
- Kinematic model is sufficient, if we have good velocity control and dynamic effects are not vital.

$$\dot{q} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \frac{r}{2}\cos\phi & \frac{r}{2}\cos\phi \\ \frac{r}{2}\sin\phi & \frac{r}{2}\sin\phi \\ -r/L & r/L \end{bmatrix} \begin{bmatrix} u_L \\ u_R \end{bmatrix}$$

- We can rewrite it as:

$$\begin{aligned} \dot{x} &= v\cos\phi \\ \dot{y} &= v\sin\phi \\ \dot{\phi} &= \omega, \end{aligned}$$

if

$$u_L = \frac{2v - \omega l}{2r} \text{ and } u_R = \frac{2v + \omega l}{2r}.$$

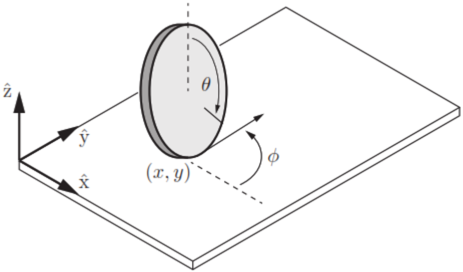


Figure: Wheel rolling without slipping

$$\dot{q} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} r\cos\phi & 0 \\ r\sin\phi & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

- u_1 is the wheel's driving speed and u_2 is heading direction turning speed.

$$\dot{x} = v\cos\phi$$
$$\dot{y} = v\sin\phi$$
$$\dot{\phi} = \omega$$

- The model for car can also be written as:

$$\dot{x} = v\cos\phi$$
$$\dot{y} = v\sin\phi$$
$$\dot{\phi} = \omega,$$

with the expressions

$$v = v$$
$$\psi = \arctan\left(\frac{\omega d}{v}\right)$$

converting the controls (v, ω) to actual controls (v, ψ).

Assumptions while using classical control for robots

- **Assumption:** We're able to measure the controlled variables, typically position and orientation of the robot, with respect to either a fixed frame or a path that the robot should follow.
- **Assumption:** Observations are continuous in time.
- **Assumption:** Observations are not corrupted by noise.

Table of Contents

Orientation Control: Set heading of robot [2]

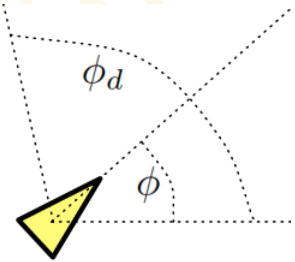
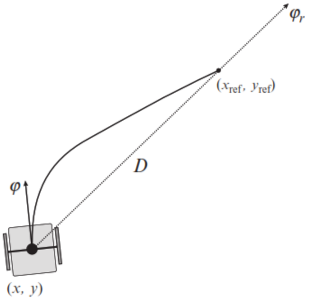


Figure: Orientation Control. Image Credit: Dr. Magnus Egerstedt

- Current orientation: $\phi(t)$.
- Desired heading: $\phi_{\mathrm{r}}(t)$.
- Example: Assume DD robot driving at a constant speed towards goal.
- Control: $\omega(t) = K\left(\phi_{\mathrm{r}}(t) - \phi(t)\right)$
- If $\phi_{\mathrm{r}}(t)$ is constant, ϕ exponentially converges to ϕ_{r}.



- We need to reach $(x_{\text{ref}}, y_{\text{ref}})$.
- First set heading $(v = 0)$ and then move.

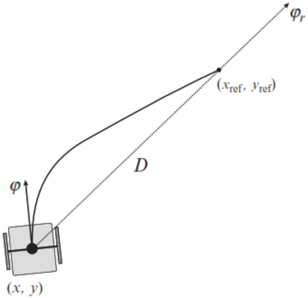
$$\phi_r(t) = \arctan \frac{y_{\text{ref}} - y(t)}{x_{\text{ref}} - x(t)}$$

$$\omega(t) = K_1 \left(\phi_r(t) - \phi(t)\right)$$

- Once we start moving, how do we stop?

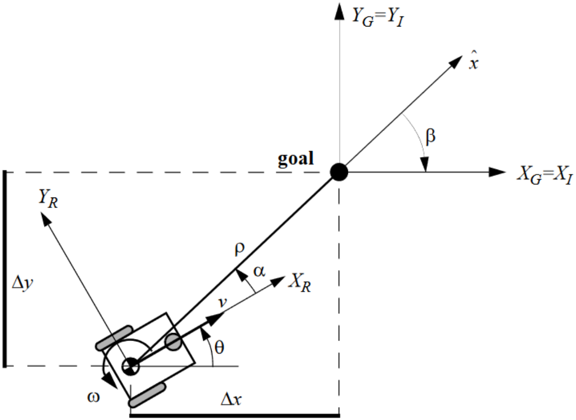
$$v(t) = K_2 \sqrt{(x_{\text{ref}} - x(t))^2 + (y_{\text{ref}} - y(t))^2}$$

Control to a reference position: Possible Issues



- Crossing the reference makes the reference orientation opposite and cause rotation of the robot.
- The control command could exceed the actuator limits if distance to reference point is large.
- Robot can overtake goal position because of noise or model errors.

Another controller for pose control [3]

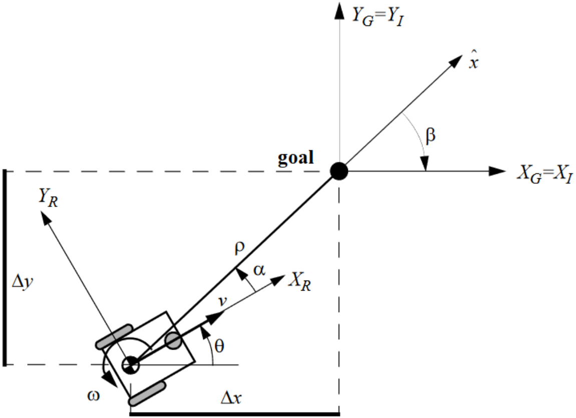


- Coordinate transformation to polar coordinates with origin at goal position:

$$\rho = \sqrt{\Delta x^2 + \Delta y^2}$$
$$\alpha = -\theta + \arctan 2\,(\Delta y, \Delta x)$$
$$\beta = -\theta - \alpha$$
$$\Rightarrow \begin{bmatrix} \dot{\rho} \\ \dot{\alpha} \\ \dot{\beta} \end{bmatrix} = \begin{bmatrix} -\cos\alpha & 0 \\ \frac{\sin\alpha}{\rho} & -1 \\ -\frac{\sin\alpha}{\rho} & 0 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}$$

when $\alpha \in (-\pi/2, \pi/2]$

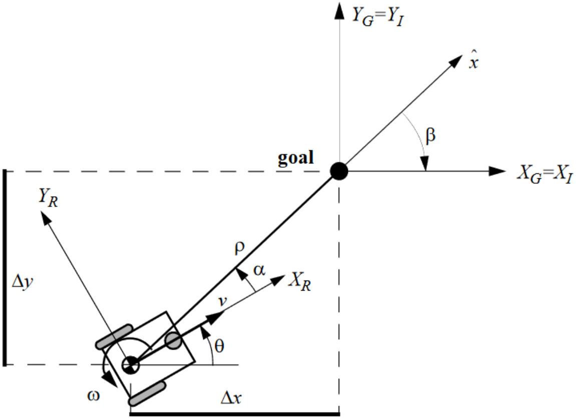
Another controller for pose control [3]



- Coordinate transformation to polar coordinates with origin at goal position:

when $\alpha \in (-\pi, -\pi/2] \cup (\pi/2, \pi]$, then by setting $v = -v$,

$$\begin{bmatrix} \dot{\rho} \\ \dot{\alpha} \\ \dot{\beta} \end{bmatrix} = \begin{bmatrix} \cos\alpha & 0 \\ -\frac{\sin\alpha}{\rho} & 1 \\ \frac{\sin\alpha}{\rho} & 0 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}$$



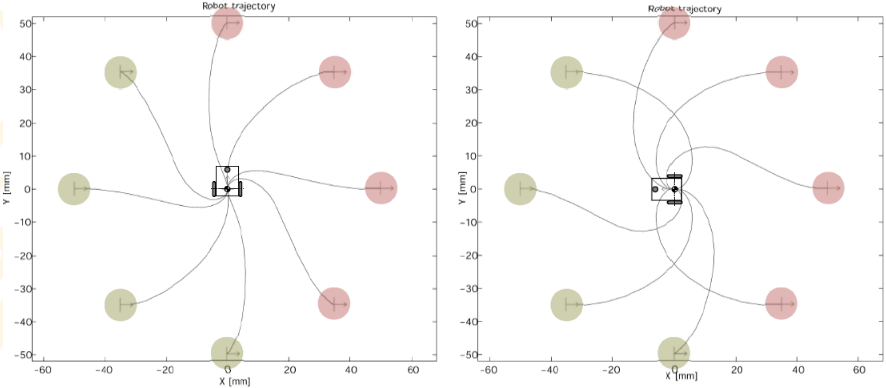
- Use the following control law:

$$
\begin{aligned}
v &= k_\rho \, \rho \\
\omega &= k_\alpha \, \alpha + k_\beta \, \beta
\end{aligned}
$$

- It requires:

$$
\begin{aligned}
k_\rho &> 0 \\
k_\beta &< 0 \\
k_\alpha - k_\rho &> 0
\end{aligned}
$$

- The goal is in the center and the initial position on the circle.



$\alpha \in I_{1/2}$

$I_1 = \left(-\frac{\pi}{2}, \frac{\pi}{2}\right]$
$\rightarrow$ forward pointing to goal

$I_2 = \left(-\pi, -\frac{\pi}{2}\right] \cup \left(\frac{\pi}{2}, \pi\right]$
$\rightarrow$ backward pointing to goal

$k = (k_\rho, k_\alpha, k_\beta) = (3, 8, -1.5)$

Autonomous Mobile Robots
Roland Siegwart, Margarita Chli, Nick Lawrance

2021 | Mobile Robots Kinematics - add ons | 21

Table of Contents

- Make the robot follow $(x_r(t), y_r(t), \phi_r(t))$.
- With only feedback, large gain is needed to make control errors small, making the approach susceptible to disturbances.
- Use Feedforward + Feedback.

- Make the robot follow $(x_r(t), y_r(t), \phi_r(t))$.
- $$\omega(t) = \frac{d}{dt}\left[\arctan\left(\frac{\dot{y}_r(t)}{\dot{x}_r(t)}\right)\right] = \frac{\dot{x}_r(t)\ddot{y}_r(t) - \dot{y}_r(t)\ddot{x}_r(t)}{\dot{x}_r^2(t) + \dot{y}_r^2(t)}$$
- $$v(t) = \sqrt{\dot{x}_r^2(t) + \dot{y}_r^2(t)}$$

- Say controls are robot velocity v and curvature κ

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \cos\psi \\ \sin\psi \\ \kappa \end{bmatrix} v$$

- Say, a reference trajectory $(x_r(t), u_r(t))$ has been generated, where x is the state vector.

One way to design controllers is to linearize nonlinear systems.

- Linearize around the reference trajectory.
- For state x, if the system dynamics are

$$\dot{x} = f(x, u, t),$$

then linearized dynamics are:

$$\dot{\delta x} = \left.\frac{\partial f}{\partial x}\right|_{(x,u)=(x_r,u_r)} \delta x + \left.\frac{\partial f}{\partial u}\right|_{(x,u)=(x_r,u_r)} \delta u$$

$$\frac{d}{dt}\begin{bmatrix}\delta x\\ \delta y\\ \delta\psi\end{bmatrix} = \begin{bmatrix}0 & 0 & -v\,s\psi\\ 0 & 0 & v\,c\psi\\ 0 & 0 & 0\end{bmatrix}\begin{bmatrix}\delta x\\ \delta y\\ \delta\psi\end{bmatrix} + \begin{bmatrix}c\psi & 0\\ s\psi & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}\delta v\\ \delta\kappa\end{bmatrix}$$

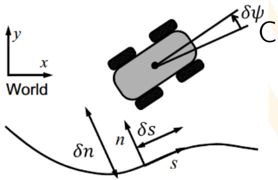


Figure: δs : Along-track error; δn : Cross-track error

Convert to body frame by multiplying with rotation matrix:

$$\begin{bmatrix}c\psi & s\psi & 0\\ -s\psi & c\psi & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}\delta\dot{x}\\ \delta\dot{y}\\ \delta\dot{\psi}\end{bmatrix} = \begin{bmatrix}0 & 0 & 0\\ 0 & 0 & v\\ 0 & 0 & 0\end{bmatrix}\begin{bmatrix}\delta x\\ \delta y\\ \delta\psi\end{bmatrix} + \begin{bmatrix}1 & 0\\ 0 & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}\delta v\\ \delta\kappa\end{bmatrix}$$

Define $\delta x = R\delta s$:

$$\begin{bmatrix}\delta\dot{s}\\ \delta\dot{n}\\ \delta\dot{\psi}\end{bmatrix} = \begin{bmatrix}0 & 0 & 0\\ 0 & 0 & V\\ 0 & 0 & 0\end{bmatrix}\begin{bmatrix}\delta s\\ \delta n\\ \delta\psi\end{bmatrix} + \begin{bmatrix}1 & 0\\ 0 & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}\delta v\\ \delta\kappa\end{bmatrix}$$

We can apply linear systems theory to this system.

- If δs is used for the complete state vector, then we have

$$\delta\dot{s}(t) = F(t)\delta s(t) + G(t)\delta u(t).$$

- This is in standard form of a state-space model:

$$\dot{x} = F(t)x + G(t)u,$$

where x and u are state and control vector respectively.

Is the system controllable?

- A system is **controllable or reachable** if it is possible to find $u(t)$ for the system, given x_0 and x_f, such that system evolves from the initial state x_0 to x_f with this control in finite time.
- Test for controllability of linear system is that the matrix

$$\begin{bmatrix} F & FG & F^2G & \cdots & F^{n-1}G \end{bmatrix},$$

is full rank.

State feedback control law

- Assuming that the state is available, feedback law $\delta u = -K\delta s$ can be used to control system to the desired state.

- Based on intuition, let's choose some terms in K to be zero:

$$u(t) = u_r(t) + \delta u(t) = u_r(t) - \begin{bmatrix} k_s & 0 & 0 \\ 0 & k_n & k_\psi \end{bmatrix} \begin{bmatrix} \delta s \\ \delta n \\ \delta \psi \end{bmatrix}$$

Left: Open loop control
Middle: Heading Compensation, $\delta\psi$ only
Right: Pose Compensation

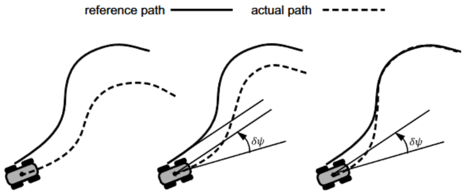


Figure: Caption here

Stability of a linear system

The closed-loop dynamics are:

$$\begin{aligned}\delta\dot{s} &= F\delta s + G\delta u \\ &= F\delta s - GK\delta s \\ \delta\dot{s} &= (F - GK)\delta s\end{aligned}$$

We want $\delta s \to 0$. **For a system, $\dot{x} = Ax$, the state $x \to 0$ iff all eigenvalues of A have negative real parts.**

$$F - GK = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & V \\ 0 & 0 & 0 \end{bmatrix} - \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} k_s & 0 & 0 \\ 0 & k_n & k_\psi \end{bmatrix}$$

Characteristic polynomial is:

$$\det(\lambda I - F + GK) = (\lambda + k_s)(\lambda^2 + \lambda k_\psi + k_n V)$$

- We need $k_s < 0$.
- $\lambda = -\frac{k_\psi}{2} \pm \frac{\sqrt{k_\psi^2 - 4k_n V}}{2}$. For fast response, we want real repeated roots: $k_\psi > 0$ and $k_\psi^2 - 4k_n V = 0$

Table of Contents

Pure Pursuit Controller

- Read the paper on PPC.
- Explain it to each other.

Develop intuition for control gains in PID.

- Play with control gains on the simulations to develop intuition about each controller.
- Read the provided paper for details of PD controller expression.

Other Control Approaches

- Lyapunov-based Design
- Feedback Linearization
- Linear Quadratic Regulator (LQR)
- Model Predictive Control (MPC)

Table of Contents

[1] **Alonzo Kelly.**
Mobile robotics: mathematics, models, and methods.
Cambridge University Press, 2013.

[2] **Gregor Klančar, Andrej Zdešar, S Blažič, and I Škrjanc.**
Wheeled mobile robotics.
Elsevier, 2017.

[3] **Roland Siegwart, Illah R Nourbakhsh, and Davide Scaramuzza.**
Autonomous mobile robots, volume 15.
MIT press, 2011.