Robot Control Basics
CS 685

Control basics

- Use some concepts from control theory to understand and learn how to control robots
- Control Theory – general field studies control and understanding of behavior of dynamical systems

  (robots, epidemics, biological systems, stock markets etc.)

Control basics

- Basic ingredients
  - state of the system $\vec{x} = [x, y, \theta]$ current position of the robot
  - dynamics behavior of the systems as a function of time (description how system state changes as a function of time)
  - system of differential equations $\dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{u})$

$$\dot{x} = v \cos\theta$$

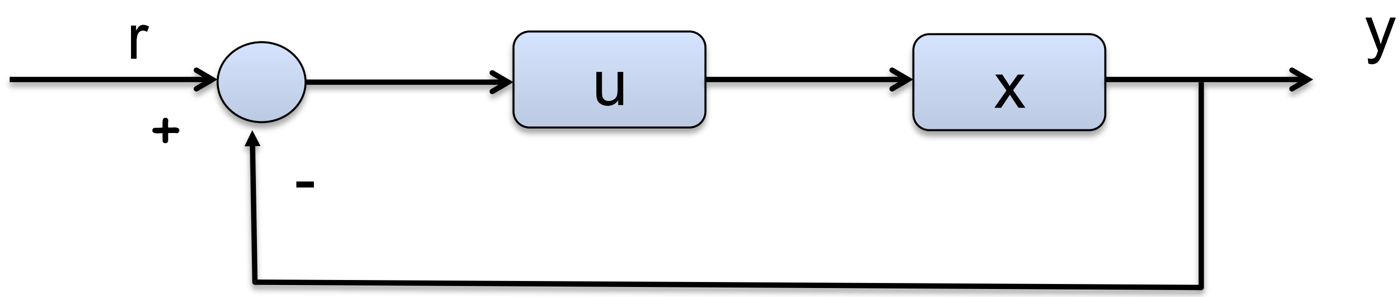
$$\dot{y} = v \sin\theta$$

$$\dot{\theta} = \omega$$

  - control input which can affect the behavior $u = [v, \omega]$
  - controller which takes some function of the goal, the state

Control basics

- Basic ingredients
  - controller which takes some function of the goal, the state
  - y output, measurement of some aspect of the state
- Feedback control – how to compute the control based on output (state) and the desired objective



- Difference equations (examples)

$$\boldsymbol{x}_{k+1} = f(\boldsymbol{x}_k, \boldsymbol{u}_k)$$

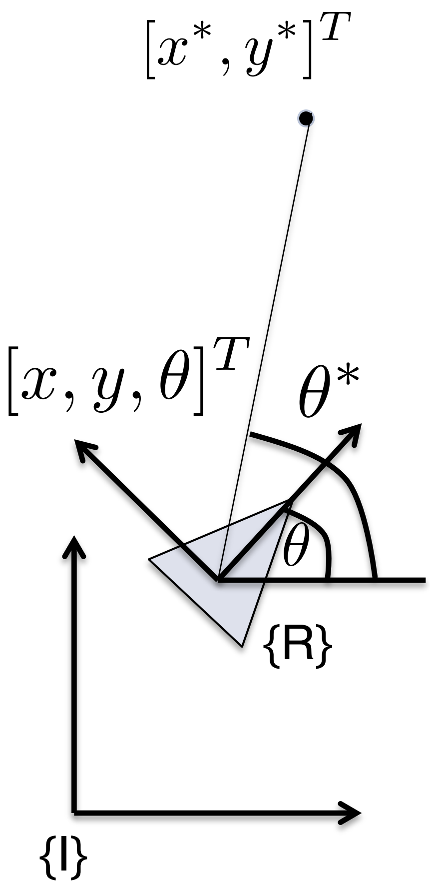
Simple control strategies

- Moving to a point – go to a point
- Consider a problem of moving to a point (x,y)
- How to control angular and linear velocity of the mobile robot
- Linear velocity – proportional to distance
- Angular velocity – steer towards the goal

- Following a line – steer toward a line
- Angular velocity proportional to the combination distance from the line and also to alignment with the line

Moving to a point

- Differential drive robot – go from the current pose $[x, y, \theta]^T$ to desired point with coordinates $[x^*, y^*]^T$



$$\theta^* = \tan^{-1} \frac{y^* - y}{x^* - x}$$

$$v = K_v \sqrt{(x^* - x)^2 + (y^* - y)^2)}$$

$$\omega = K_h(\theta^* - \theta)$$

Source P. Corke: Robotics, Vision and Control. Springer

Moving to a line

- Equation of a line

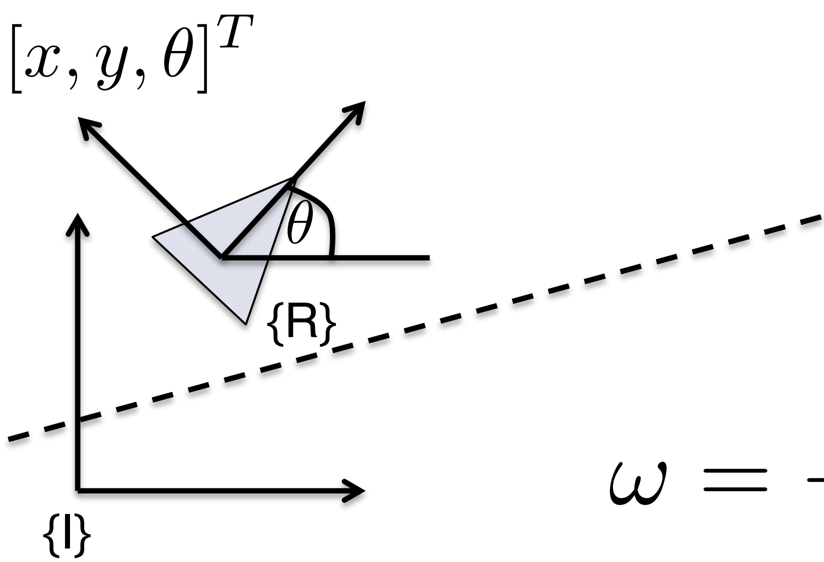
$$ax + by + c = 0$$

- Shortest distance of the robot the line

$$d = \frac{[a, b, c][x, y, 1]^T}{\sqrt{a^2 + b^2}}$$

- Orientation of the line

$$\theta^* = \tan^{-1} \frac{-a}{b}$$



$$\alpha_d = -K_d d \quad K_d > 0$$

$$\alpha_h = K_h(\theta^* - \theta)$$

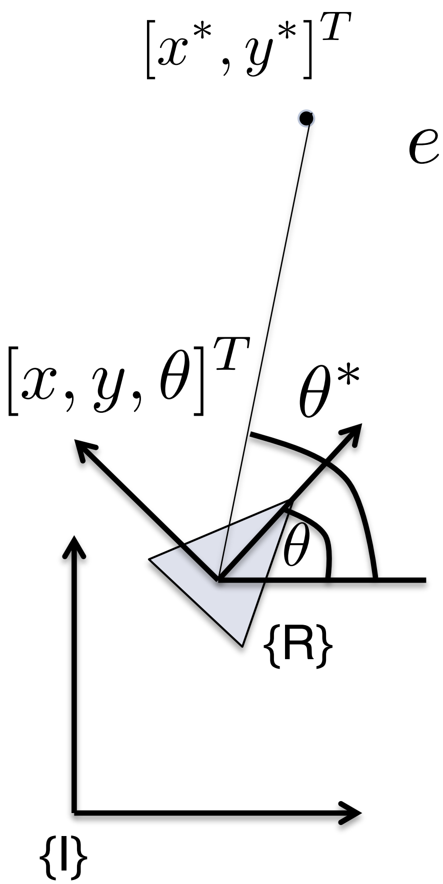
$$\omega = -K_d d + K_h(\theta^* - \theta)$$

- Steer towards the line and align the robot with the line

Following a path

- Same as going to the point – now sequence of waypoints $x(t), y(t)$

$$\theta^* = \tan^{-1} \frac{y^* - y}{x^* - x}$$



$$e = K_v \sqrt{(x^* - x)^2 + (y^* - y)^2)} - d^*$$

d^* distance behind the pursuit point

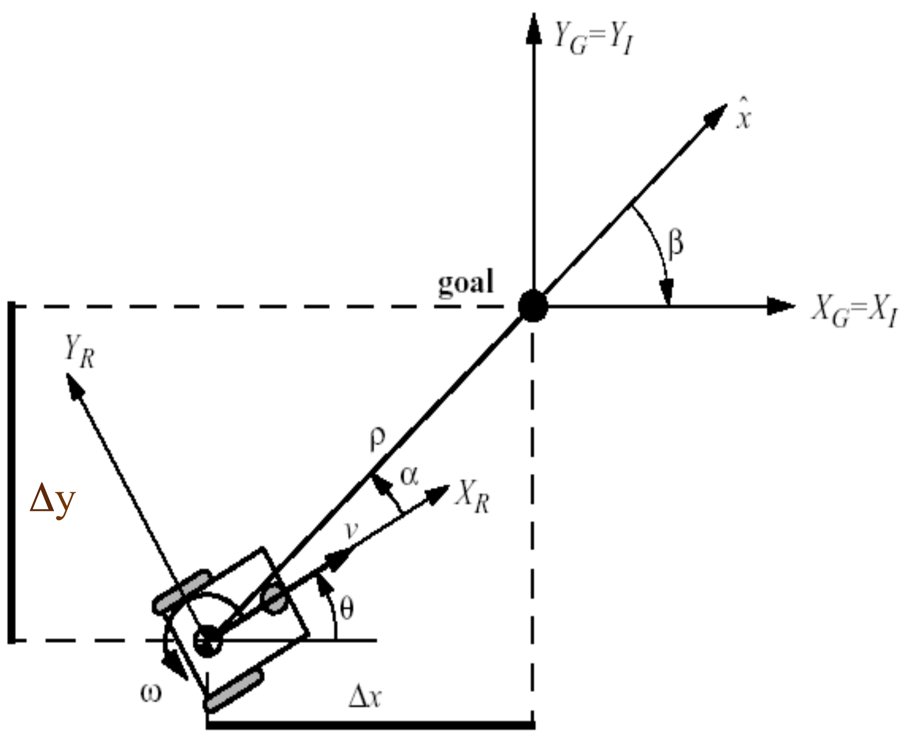
$$v = K_v e + K_i \int e dt$$

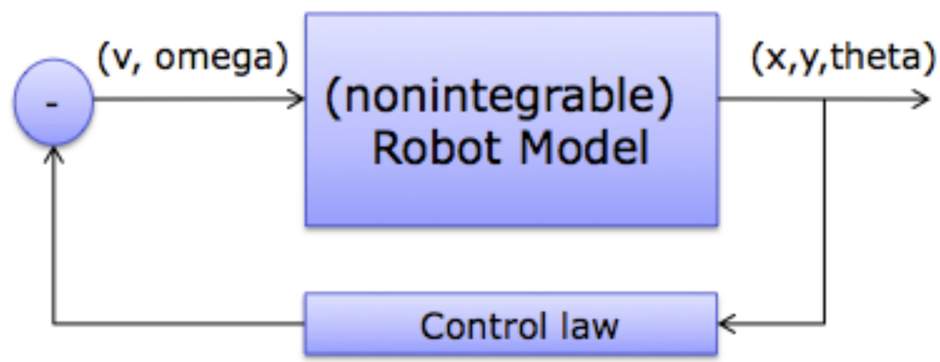
$$\omega = K_h(\theta^* - \theta)$$

Source P. Corke: Robotics, Vision and Control. Springer

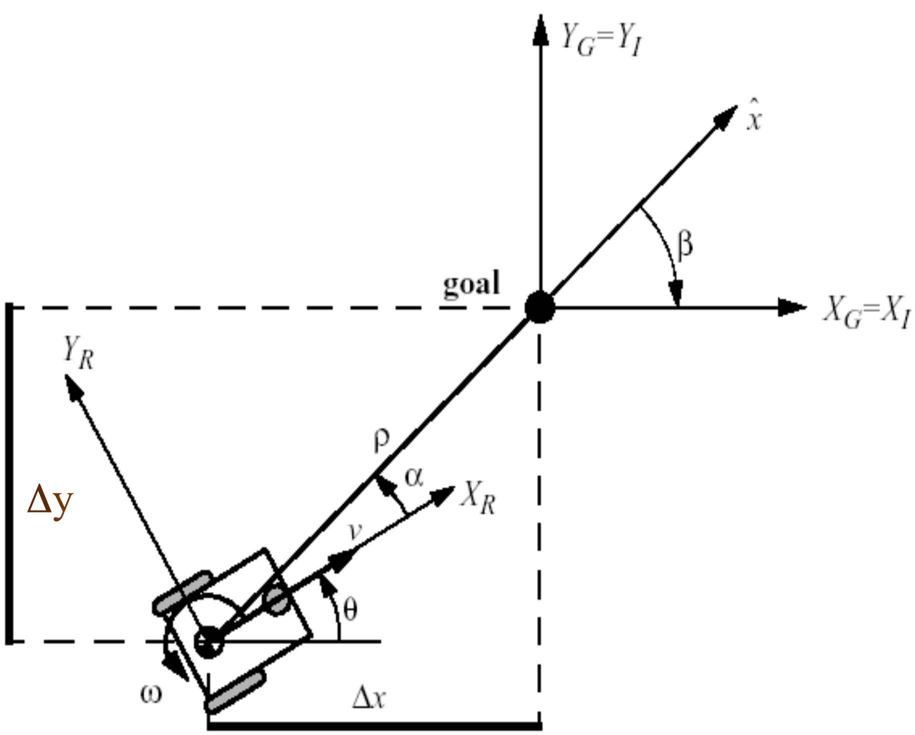
Feedback control

- More suitable alternative
- Use state feedback controller
- At each instance of time compute a control law
- Given the current error between current and desired position





Kinematic Position Control



The kinematic of a differential drive mobile robot described in the initial frame $\{x_I, y_I, \theta\}$ is given by,

$$^{I}\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}$$

relating the linear velocities in the direction of the x_I and y_I of the initial frame.

Let α denote the angle between the x_R axis of the robots reference frame and the vector connecting the center of the axle of the wheels with the final position.

Motion Control (kinematic control)

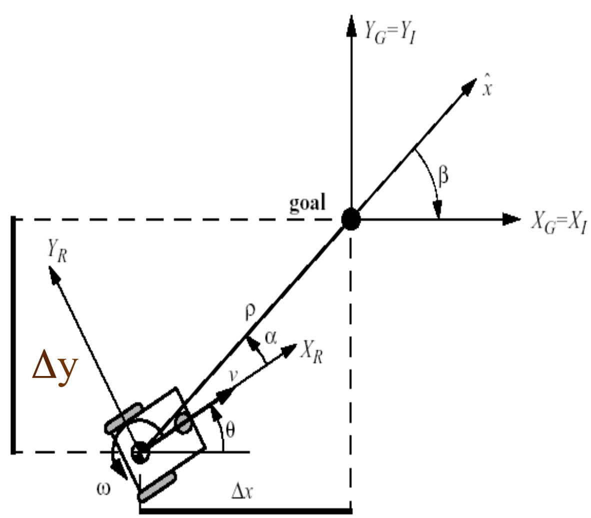
- The objective of a kinematic controller is to follow a trajectory described by its position and/or velocity profiles as function of time.
- Motion control is not straight forward because mobile robots are non-holonomic systems.
- However, it has been studied by various research groups and some adequate solutions for (kinematic) motion control of a mobile robot system are available.
- Most controllers are not considering the dynamics of the system

Motion Control: Feedback Control

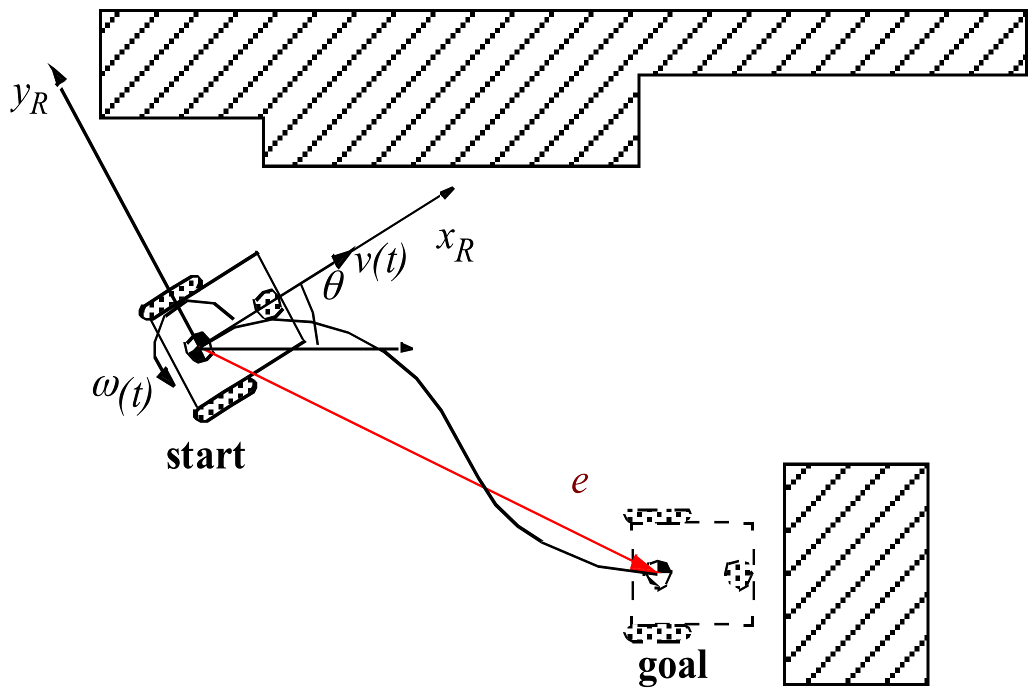
- Set intermediate positions lying on the requested path.
- Given a goal how to compute the control commands for
- linear and angular velocities to reach the desired configuration

Problem statement

- Given arbitrary position and orientation of the robot $[x, y, \theta]$
  how to reach desired goal orientation and position $[x_g, y_g, \theta_g]$



Motion Control: Feedback Control, Problem Statement



- Find a control matrix *K*, if exists

$$K = \begin{bmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \end{bmatrix}$$

- with $k_{ij}=k(t,e)$
- such that the control of $v(t)$ and $\omega(t)$

$$\begin{bmatrix} v(t) \\ \omega(t) \end{bmatrix} = K \cdot e = K \cdot {}^{R}\begin{bmatrix} x \\ y \\ \theta \end{bmatrix}$$

- drives the error e to zero.

$$\lim_{t \to \infty} e(t) = 0$$

Motion Control:

Kinematic Position Control

- The kinematic of a differential drive mobile robot described in the initial frame $\{x_I, y_I, \theta\}$ is given by,

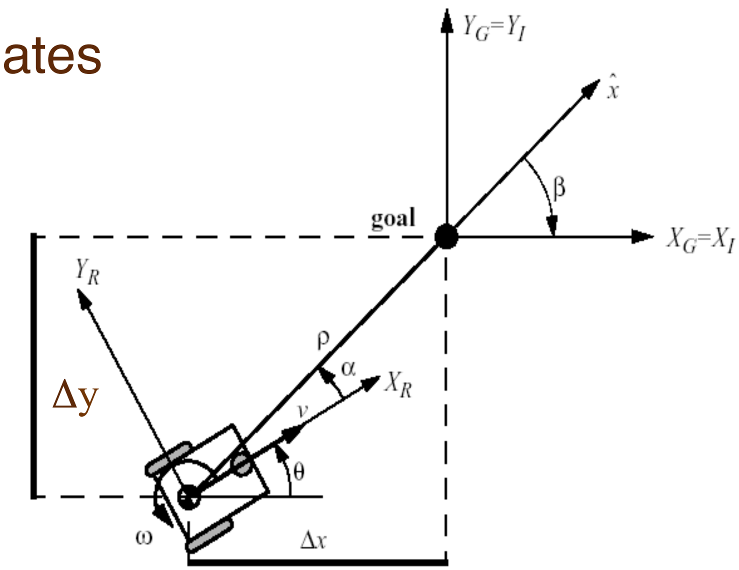
$$
{}^{I}\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}
$$

where and are the linear velocities in the direction of the x_I and y_I of the initial frame.

Let α denote the angle between the x_R axis of the robots reference frame and the vector connecting the center of the axle of the wheels with the final position.

Kinematic Position Control: Coordinates Transformation

Coordinates transformation into polar coordinates with its origin at goal position:



$$\rho = \sqrt{\Delta x^2 + \Delta y^2}$$

$$\alpha = -\theta + \operatorname{atan2}(\Delta y, \Delta x)$$

$$\beta = -\theta - \alpha$$

System description, in the new polar coordinates

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

For α from $I_1 = \left(-\frac{\pi}{2}, \frac{\pi}{2}\right]$ for $I_2 = (-\pi, -\pi/2] \cup (\pi/2, \pi]$

Kinematic Position Control: Remarks

- The coordinates transformation is not defined at $x = y = 0$; as in such a point the determinant of the Jacobian matrix of the transformation is not defined, i.e. it is unbounded

- For $\alpha \in I_1$ the forward direction of the robot points toward the goal, for $\alpha \in I_2$ it is the backward direction.

- By properly defining the forward direction of the robot at its initial configuration, it is always possible to have $\alpha \in I_1$ at $t = 0$. However this does not mean that α remains in I_1 for all time t.

Kinematic Position Control: The Control Law

- It can be shown, that with

$$v = k_\rho \rho \qquad \omega = k_\alpha \alpha + k_\beta \beta$$

  the feedback controlled system

$$\begin{bmatrix} \dot{\rho} \\ \dot{\alpha} \\ \dot{\beta} \end{bmatrix} = \begin{bmatrix} -k_\rho \rho \cos\alpha \\ k_\rho \sin\alpha - k_\alpha \alpha - k_\beta \beta \\ -k_\rho \sin\alpha \end{bmatrix}$$

- will drive the robot to $(\rho, \alpha, \beta) = (0,0,0)$
- The control signal *v* has always constant sign,
  - the direction of movement is kept positive or negative during movement
  - parking maneuver is performed always in the most natural way and without ever inverting its motion.

Question: How to select the constant parameters k's so as to achieve that the error will go to zero

- Digression – eigenvectors and eigenvalues review

Eigenvalues and Eigenvectors

- Motivated by solution to differential equations

$$A \in \Re^{n \times n} \qquad \dot{\mathbf{u}} = A\mathbf{u} \qquad A = \begin{bmatrix} 4 & -5 \\ 2 & -3 \end{bmatrix}$$

- For square matrices

$$\begin{bmatrix} \dot{v} \\ \dot{w} \end{bmatrix} = A \begin{bmatrix} v \\ w \end{bmatrix}$$

$$\dot{u} = au$$

$$u(t) = e^{at}u(0)$$

For scalar ODE's

Suppose solution have this form of exponentials

$$v(t) = e^{\lambda t} y$$
$$w(t) = e^{\lambda t} z$$

Substitute back to the equation

$$\lambda \cancel{e^{\lambda t}} y = 4\cancel{e^{\lambda t}} y - 5\cancel{e^{\lambda t}} z$$

$$\lambda \cancel{e^{\lambda t}} z = 2\cancel{e^{\lambda t}} y - 3\cancel{e^{\lambda t}} z$$

and denote x as $\mathbf{x} = \begin{bmatrix} y \\ z \end{bmatrix}$ then eq. above is $\lambda \mathbf{x} = \begin{bmatrix} 4 & -5 \\ 2 & -3 \end{bmatrix} \mathbf{x}$

Eigenvalues and Eigenvectors

$$\lambda \mathbf{x} = \begin{bmatrix} 4 & -5 \\ 2 & -3 \end{bmatrix} \mathbf{x} \qquad A\mathbf{x} = \lambda \mathbf{x}$$

eigenvector

eigenvalue

Solve the equation: $$(A - \lambda I)\mathbf{x} = 0 \qquad (1)$$

x – is in the null space of $(A - \lambda I)$

λ is chosen such that $(A - \lambda I)$ has a null space

Computation of eigenvalues and eigenvectors (for dim 2,3)

1. Compute determinant
2. Find roots (eigenvalues) of the polynomial such that determinant = 0
3. For each eigenvalue solve the equation (1)

For larger matrices – alternative ways of computation

Eigenvalues and Eigenvectors

For the previous example

$$\lambda_1 = -1, x_1 = [1, 1]^T \qquad \lambda_2 = -2, x_2 = [5, 2]^T$$

We will get special solutions to ODE $\dot{\mathbf{u}} = A\mathbf{u}$

$$A = \begin{bmatrix} 4 & -5 \\ 2 & -3 \end{bmatrix} \qquad \mathbf{u} = e^{\lambda_1 t}\begin{bmatrix} 1 \\ 1 \end{bmatrix} \qquad \mathbf{u} = e^{\lambda_2 t}\begin{bmatrix} 5 \\ 2 \end{bmatrix}$$

Their linear combination is also a solution (due to the linearity of $\dot{\mathbf{u}} = A\mathbf{u}$

$$\mathbf{u} = c_1 e^{\lambda_1 t}\begin{bmatrix} 1 \\ 1 \end{bmatrix} + c_2 e^{\lambda_1 t}\begin{bmatrix} 5 \\ 2 \end{bmatrix}$$

In the context of diff. equations – special meaning
Any solution can be expressed as linear combination
Individual solutions correspond to modes

Eigenvalues and Eigenvectors

$$A\mathbf{x} = \lambda\mathbf{x}$$

Only special vectors are eigenvectors
- such vectors whose direction will not be changed by the transformation A (only scale)
- they correspond to normal modes of the system act independently

Examples	eigenvalues	eigenvectors
$A = \begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix}$	2, 3	$\begin{bmatrix} 0 \\ 1 \end{bmatrix}; \begin{bmatrix} 1 \\ 0 \end{bmatrix}$

Whatever A does to an arbitrary vector is fully determined by its eigenvalues and eigenvectors

$$A\mathbf{x} = 2\lambda_1 v_1 + 5\lambda_2 v_2$$

Previously - Eigenvalues and Eigenvectors

For the previous example

$$\lambda_1 = -1, x_1 = [1,1]^T \qquad \lambda_2 = -2, x_2 = [5,2]^T$$

We will get special solutions to ODE $\dot{\mathbf{u}} = A\mathbf{u}$

$$A = \begin{bmatrix} 4 & -5 \\ 2 & -3 \end{bmatrix} \qquad \mathbf{u} = e^{\lambda_1 t}\begin{bmatrix} 1 \\ 1 \end{bmatrix} \qquad \mathbf{u} = e^{\lambda_2 t}\begin{bmatrix} 5 \\ 2 \end{bmatrix}$$

The linear combination is also a solution (due to the linearity of $\dot{\mathbf{u}} = A\mathbf{u}$)

$$\mathbf{u} = c_1 e^{\lambda_1 t}\begin{bmatrix} 1 \\ 1 \end{bmatrix} + c_2 e^{\lambda_1 t}\begin{bmatrix} 5 \\ 2 \end{bmatrix}$$

In the context of diff. equations – special meaning
Any solution can be expressed as linear combination
Individual solutions correspond to modes

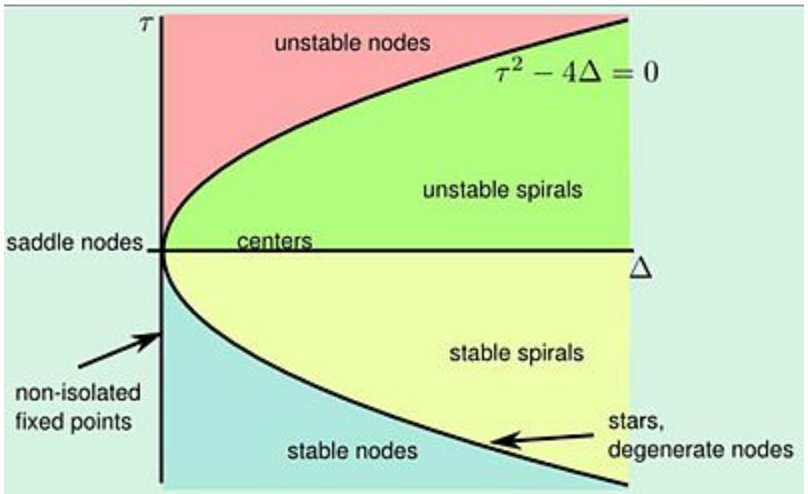
Eigenvalues of linear system

- Given linear system of differential equations

$$\dot{\mathbf{x}} = A\mathbf{x}$$

- For 2 dimesional system (A is 2 x 2), A has two eigenvalues

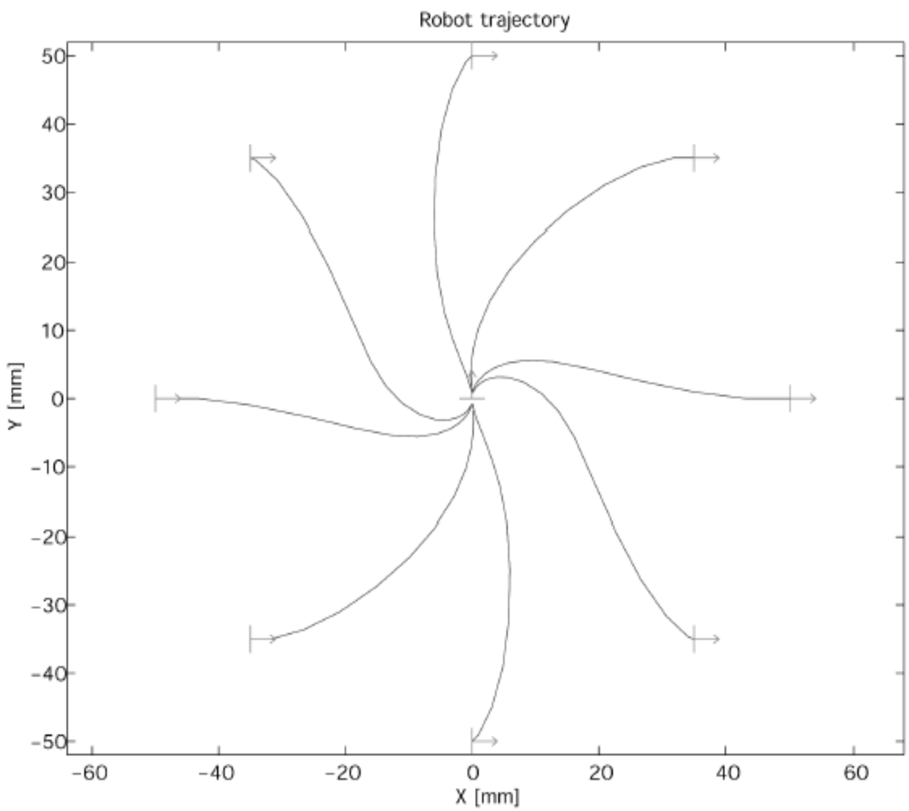
- Define $\Delta = \lambda_1 \lambda_2$ and $\tau = \lambda_1 + \lambda_2$

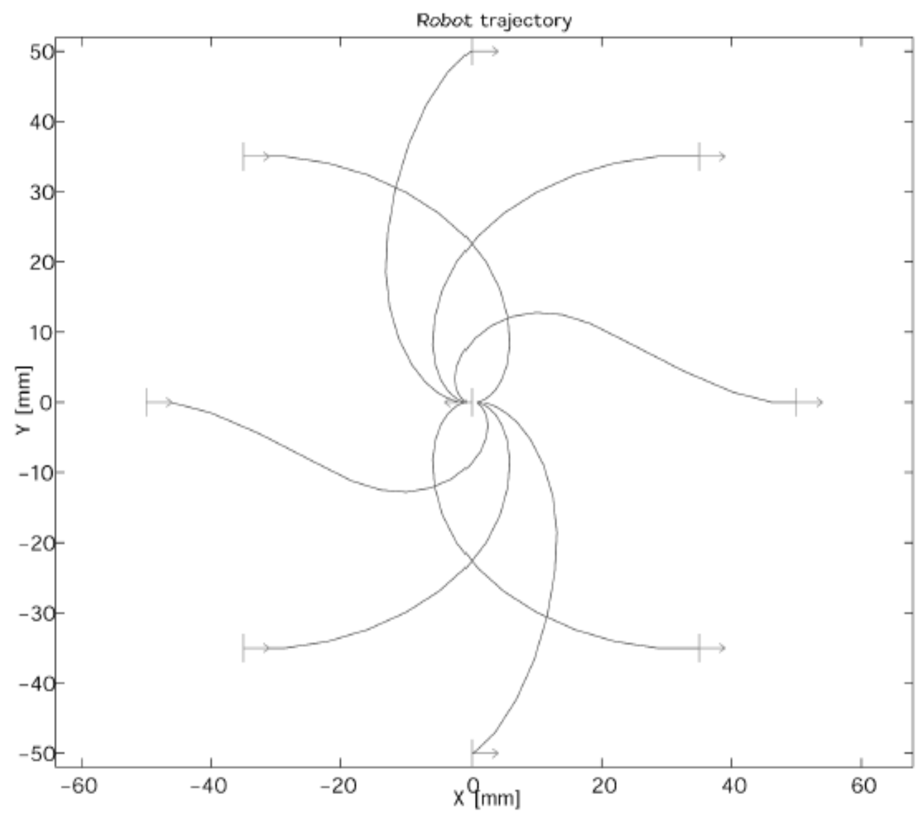


- if $\Delta < 0$ saddle node
-
- if $\Delta > 0$ we have two cases

1. $\tau > 0$ eigenvalues positive
2. $\tau < 0$ eigenvalues negative : stable nodes of the system

Kinematic Position Control: Resulting Path





Linearization

- But our system is not linear, e.g. cannot be written in the form

$$\dot{\mathbf{x}} = A\mathbf{x}$$

Some terminology

- We have derived kinematics equations of the robot

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

- Non-linear differential equation $\dot{x} = f(x, u)$
- In our case

$$
\begin{aligned}
\dot{x} &= f_1(x, y, \theta, v, \omega) \\
\dot{y} &= f_2(x, y, \theta, v, \omega) \\
\dot{\theta} &= f_3(x, y, \theta, v, \omega)
\end{aligned}
$$

Jacobian Matrix

- Suppose you have two dim function

$$f(\mathbf{x}) = \begin{bmatrix} f_1(\mathbf{x}) \\ f_2(\mathbf{x}) \end{bmatrix}$$

- Gradient operator $\nabla_{\mathbf{x}} = \begin{bmatrix} \frac{\partial}{\partial x_1} & \frac{\partial}{\partial x_2} & \cdots & \frac{\partial}{\partial x_n} \end{bmatrix}^T$

- Jacobian is defined as $F_x = J_F$

$$\mathbf{F}_{\mathbf{x}} = \begin{bmatrix} f_1(\mathbf{x}) \\ f_2(\mathbf{x}) \end{bmatrix} \cdot \begin{bmatrix} \frac{\partial}{\partial x_1} & \cdots & \frac{\partial}{\partial x_n} \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \cdots & \frac{\partial f_2}{\partial x_n} \end{bmatrix}$$

- Linearization of a function

$$F(\mathbf{x}) = F(\mathbf{x}_0) + J_F(\mathbf{x}_0)d\mathbf{x}$$

- Linearization of system of diff. equations

$$\dot{\mathbf{x}} = J_F(\mathbf{x}_0)d\mathbf{x} + F(\mathbf{x}_0)$$

Kinematic Position Control: Stability Issue

- Continuous linear time-invariant system is exponentially stable if and only of the system has eigenvalues (i.e. poles of input-to-output systems) with strictly negative real parts
- Exponential Stability is a form of asymptotic stability
- In practice the system will not "blow up" give unbounded output, when given an finite input and non-zero initial condition

Kinematic Position Control: Stability Issue

- It can further be shown, that the closed loop control system is locally exponentially stable if

$$k_\rho > 0 \quad ; \quad k_\beta < 0 \quad ; \quad k_\alpha - k_\rho > 0$$

- Proof: linearize around equilibrium
  for small *x* –> cos *x* = 1, sin *x* = *x*

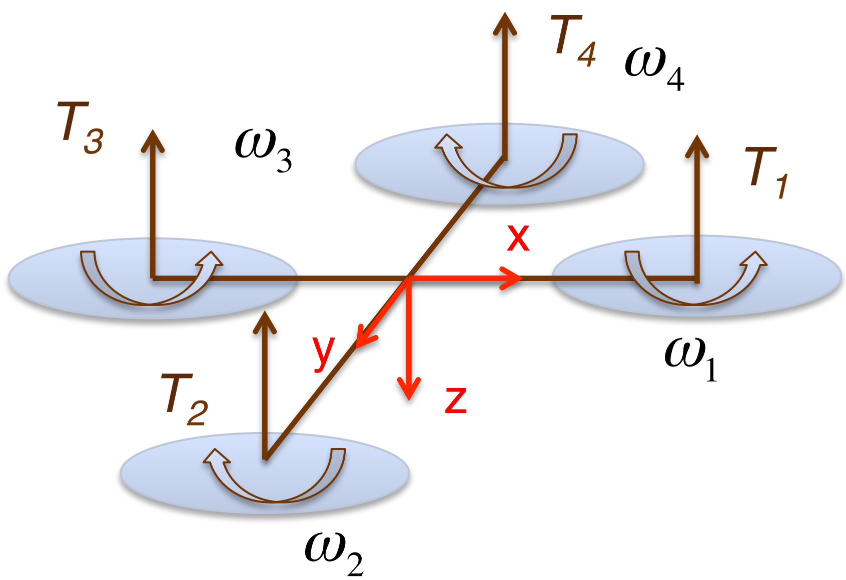
$$\begin{bmatrix} \dot{\rho} \\ \dot{\alpha} \\ \dot{\beta} \end{bmatrix} = \begin{bmatrix} -k_\rho & 0 & 0 \\ 0 & -(k_\alpha - k_\rho) & -k_\beta \\ 0 & -k_\rho & 0 \end{bmatrix} \begin{bmatrix} \rho \\ \alpha \\ \beta \end{bmatrix} \qquad A = \begin{bmatrix} -k_\rho & 0 & 0 \\ 0 & -(k_\alpha - k_\rho) & -k_\beta \\ 0 & -k_\rho & 0 \end{bmatrix}$$

- and the characteristic polynomial of the matrix *A* of all roots have negative real parts.

$$(\lambda + k_\rho)(\lambda^2 + \lambda(k_\alpha - k_\rho) - k_\rho k_\beta)$$

Quadcopters model

- Popular unmanned areal vehicles (description adopted from (Robotics, Vision and control book, P. Corke http://www.petercorke.com/RVC/)



- Upward thrust $T_i = b\omega_i^2$ moving up in the negative z dir.
- Lift const. *b* depends on air density, blade radius and chord length

Quadcopters

- Translational dynamics (Newton's law – includes mass/ acceleration/ forces) (Gravity – Total thrust (rotated to the world frame)

$$m\dot{v} = \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} - R_B^0 \begin{bmatrix} 0 \\ 0 \\ T \end{bmatrix}, \quad T = \sum_{i=1..4} T_i$$

- Rotations are generated by pairwise differences in rotor thrusts (*d* distance from the center)
- Rolling and pitching torques around *x* and *y*
- Torque in *z* – yaw torque

$$\tau_x = dT_4 - dT_2$$

$$Q_i = k\omega_i^2$$

Torque applied by the motor as opposed to Aerodynamic drag

$$\tau_x = db(\omega_4^2 - \omega_2^2)$$

$$\tau_z = (Q_1 - Q_2 + Q_3 - Q_4)$$

$$\tau_y = db(\omega_1^2 - \omega_3^2)$$

Quadcopter dynamics

- Rotational Dynamics, rot. acceleration in the airframe, Euler's eq. of motion

$$J\dot{\omega} = -\vec{\omega} \times J\vec{\omega} + \Gamma, \quad \Gamma = [\tau_x, \tau_y, \tau_z]^T$$

- Where *J is 3x3* inertia matrix
- Forces and torques acting of the airframe obtained integrating forward the eq. above and Newton's second law (prev. slide)

$$\begin{bmatrix} T \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = \begin{bmatrix} -b & -b & -b & -b \\ 0 & -db & 0 & db \\ db & 0 & -db & 0 \\ k & -k & k & -k \end{bmatrix} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} \qquad \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} = A^{-1} \begin{bmatrix} T \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix}$$

- The goal of control is then derive proper thrust and torque to achieve desired goal – compute the rotor speeds
- Substitute these to translational and rotational dynamics and get forward dynamics equations of quadropter

Inertia Matrix

- Rotational Inertia of a body in 3D is represented by a 3x3 symmetric matrix J

$$J = \begin{bmatrix} J_{xx} & J_{xy} & J_{xz} \\ J_{xy} & J_{yy} & J_{yz} \\ J_{xz} & J_{yz} & J_{zz} \end{bmatrix}$$

- Diagonal elements are moments of inertia and off-diagonal are products of inertia
- Inertia matrix is a constant and depends on the mass and the shape of the body