Course Review

METR 4202: Advanced Control & Robotics

Drs Surya Singh, Paul Pounds, and Hanna Kurniawati

Lecture # 13 – Part 1 October 22, 2012

metr4202@itee.uq.edu.au

http://itee.uq.edu.au/~metr4202/

Schedule: Almost Done

Week	Date	Lecture (M: 12-1:30, 43-102)
1	23-Jul	Introduction
2	30-Jul	Representing Position & Orientation & State (Frames, Transformation Matrices & Affine Transformations)
3	6-Aug	Robot Kinematics and Dynamics
4	13-Aug	Robot Dynamics & Control
5	20-Aug	Obstacle Avoidance & Motion Planning
6	27-Aug	Sensors, Measurement and Perception
7	3-Sep	Perception (+ Prof. S. LaValle)
8	10-Sep	Computer Vision & Localization (SFM/SLAM)
9	17-Sep	Optical Flow (Prof. M. Srinivasan)
	24-Sep	*Study break*
10	1-Oct	*Public Holiday*
11	8-Oct	Localization and Navigation
12	15-Oct	Control/Planning Under Uncertainty
13	22-Oct	Course Review & Case Study

Quick Outline

1. **Course Review**
   1. Kinematics
   2. Perception
   3. Control
   4. Uncertainty

2. **Lab 3: Q & A**

3. **Case Study**

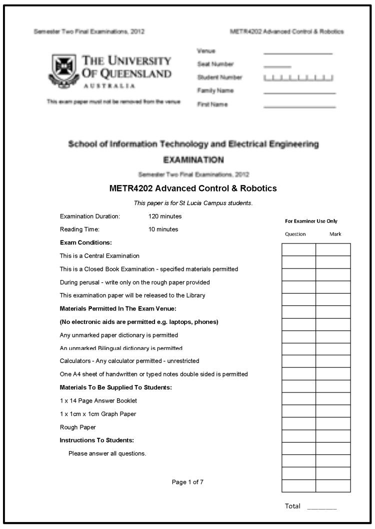
Exam Format

- Subjects:
  - Rotation Matrices: 10 %
  - Kinematics: 15 %
  - Dynamics & Motion Planning: 25 %
  - Vision: 25 %
  - State-Space Control: 25 %

- Material for the exam will come from:
  - Tutorial and Tutorial Problems: 70%
  - Lectures: 20%
  - Laboratories: 10%

- Problem Types:
  - Short answer
  - Problem based

- Write solutions in the **Answer Booklet(s)**



Semester Two Final Examinations, 2012 METR4202 Advanced Control & Robotics

THE UNIVERSITY OF QUEENSLAND AUSTRALIA

This exam paper must not be removed from the venue

Venue ____
Seat Number ____
Student Number ____
Family Name ____
First Name ____

School of Information Technology and Electrical Engineering

EXAMINATION

Semester Two Final Examinations, 2012

METR4202 Advanced Control & Robotics

This paper is for St Lucia Campus students.

Examination Duration: 120 minutes

Reading Time: 10 minutes

Exam Conditions:

This is a Central Examination

This is a Closed Book Examination - specified materials permitted

During perusal - write only on the rough paper provided

This examination paper will be released to the Library

Materials Permitted In The Exam Venue:

(No electronic aids are permitted e.g. laptops, phones)

Any unmarked paper dictionary is permitted

An unmarked Bilingual dictionary is permitted

Calculators - Any calculator permitted - unrestricted

One A4 sheet of handwritten or typed notes double sided is permitted

Materials To Be Supplied To Students:

1 x 14 Page Answer Booklet

1 x 1cm x 1cm Graph Paper

Rough Paper

Instructions To Students:

Please answer all questions.

Page 1 of 7

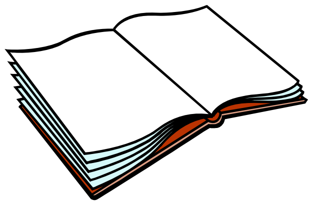
For Examiner Use Only

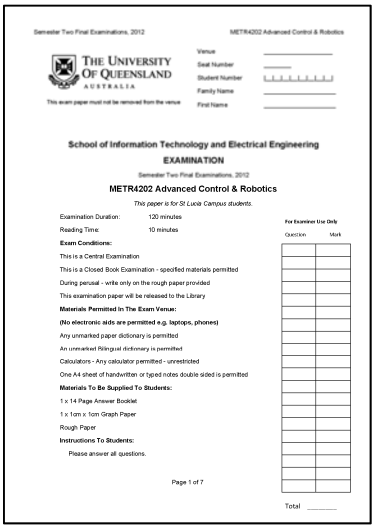
Question	Mark

Total ____

And: Yes, the Exam is **Open Book!**

- Exam is OPEN BOOK





Semester Two Final Examinations, 2012 METR4202 Advanced Control & Robotics

THE UNIVERSITY OF QUEENSLAND AUSTRALIA

This exam paper must not be removed from the venue

Venue ____

Seat Number ____

Student Number ____

Family Name ____

First Name ____

School of Information Technology and Electrical Engineering

EXAMINATION

Semester Two Final Examinations, 2012

METR4202 Advanced Control & Robotics

This paper is for St Lucia Campus students.

Examination Duration: 120 minutes

Reading Time: 10 minutes

Exam Conditions:

This is a Central Examination

This is a Closed Book Examination - specified materials permitted

During perusal - write only on the rough paper provided

This examination paper will be released to the Library

Materials Permitted In The Exam Venue:

(No electronic aids are permitted e.g. laptops, phones)

Any unmarked paper dictionary is permitted

An unmarked Bilingual dictionary is permitted

Calculators - Any calculator permitted - unrestricted

One A4 sheet of handwritten or typed notes double sided is permitted

Materials To Be Supplied To Students:

1 x 14 Page Answer Booklet

1 x 1cm x 1cm Graph Paper

Rough Paper

Instructions To Students:

Please answer all questions.

Page 1 of 7

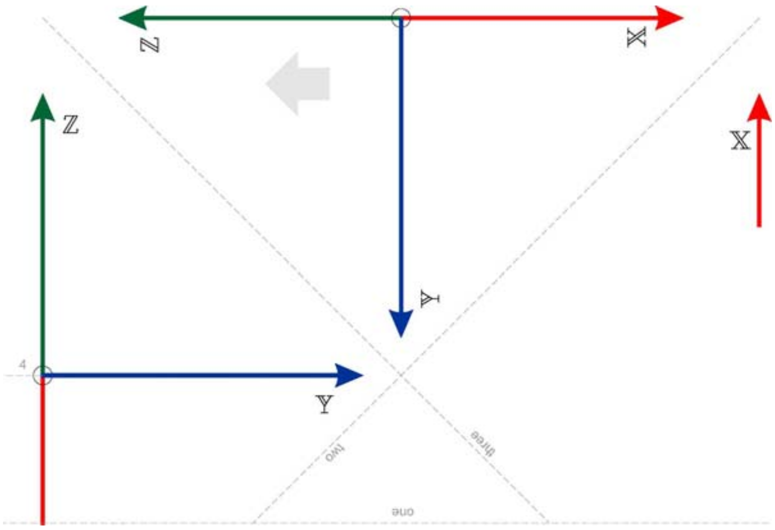
For Examiner Use Only

Question	Mark

Total ____

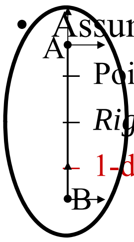
- FYI:
  - Central Exams Notified
  - Certification letter posted the class website

It All Starts With: **Frames in Space!!**

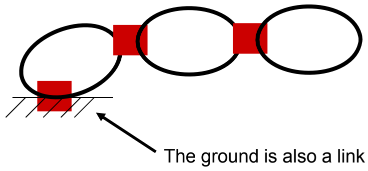


Kinematics Definition

- **Kinematics**: The study of motion in space (without regard to the forces which cause it)



- Assume:
  - Points with ***right-hand Frames***
  - *Rigid-bodies* in 3D-space (6-dof)
  - 1-dof joints: Rotary (R) or Prismatic (P) (5 constraints)

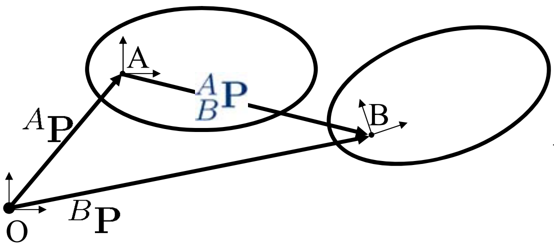


N links
M joints
→DOF = 6N-5M
→ If N=M, then DOF=N.

Position and Orientation [1]

- A **position** vectors specifies the location of a **point** in 3D (Cartesian) space

$$\mathbf{P} = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix}$$



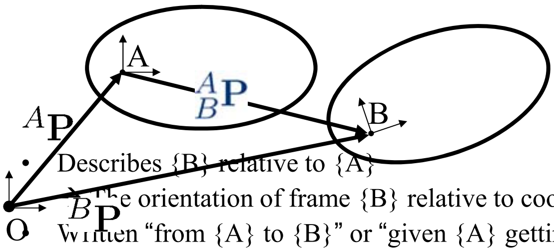
$$^{A}\mathbf{P} + {}^{A}\mathbf{P}^{B} - {}^{B}\mathbf{P} = 0$$

$$^{A}\mathbf{P}^{B} = {}^{A}\mathbf{P}_{B} = {}^{A}_{B}\mathbf{P} = \begin{bmatrix} ^{B}p_x \\ ^{B}p_y \\ ^{B}p_z \end{bmatrix} - \begin{bmatrix} ^{A}p_x \\ ^{A}p_y \\ ^{A}p_z \end{bmatrix}$$

- BUT we **also** concerned with its orientation in 3D space. This is specified as a matrix based on each **frame's unit vectors**

Position and Orientation [2]

- Orientation in 3D space:
  This is specified as a matrix based on each **frame's unit vectors**



- Describes {B} relative to {A}
  - The orientation of frame {B} relative to coordinate frame {A}
  - Written "from {A} to {B}" or "given {A} getting to {B}"

$$^{A}\mathbf{R}_{B} = {}^{A}_{B}\mathbf{R} = \left[\begin{array}{ccc} {}^{A}\hat{\imath}_{B} & {}^{A}\hat{\jmath}_{B} & {}^{A}\hat{k}_{B} \end{array}\right]$$

- **Columns** are **{B} written in {A}**

Position and Orientation [3]

- The rotations can be analysed based on the unit components …
- That is: the components of the orientation matrix are the unit vectors projected **onto** the unit directions of the reference frame

$$^{A}_{B}\mathbf{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$$

$$\begin{array}{c|c} {}^{A}_{B}R & \begin{array}{ccc} (b_x)\,\hat{\imath}_B & (b_y)\,\hat{\jmath}_B & (b_z)\,\hat{k}_B \end{array} \\ \hline \begin{array}{c} (a_x)\,\hat{\imath}_A \\ (a_y)\,\hat{\jmath}_A \\ (a_z)\,\hat{k}_A \end{array} & \begin{bmatrix} \hat{\imath}_B \cdot \hat{\imath}_A & \hat{\jmath}_B \cdot \hat{\imath}_A & \hat{k}_B \cdot \hat{\imath}_A \\ \hat{\imath}_B \cdot \hat{\jmath}_A & \hat{\jmath}_B \cdot \hat{\jmath}_A & \hat{k}_B \cdot \hat{\jmath}_A \\ \hat{\imath}_B \cdot \hat{k}_A & \hat{\jmath}_B \cdot \hat{k}_A & \hat{k}_B \cdot \hat{k}_A \end{bmatrix} \end{array}$$

Position and Orientation [4]

- Rotation is orthonormal

$$
\begin{array}{c|c}
{}^{A}_{B}R & \begin{array}{ccc} (b_x)\,\hat{\imath}_B & (b_y)\,\hat{\jmath}_B & (b_z)\,\hat{k}_B \end{array} \\
\hline
\begin{array}{c} (a_x)\,\hat{\imath}_A \\ (a_y)\,\hat{\jmath}_A \\ (a_z)\,\hat{k}_A \end{array} & \begin{bmatrix} \hat{\imath}_B \cdot \hat{\imath}_A & \hat{\jmath}_B \cdot \hat{\imath}_A & \hat{k}_B \cdot \hat{\imath}_A \\ \hat{\imath}_B \cdot \hat{\jmath}_A & \hat{\jmath}_B \cdot \hat{\jmath}_A & \hat{k}_B \cdot \hat{\jmath}_A \\ \hat{\imath}_B \cdot \hat{k}_A & \hat{\jmath}_B \cdot \hat{k}_A & \hat{k}_B \cdot \hat{k}_A \end{bmatrix}
\end{array}
$$

- The of a rotation matrix inverse = the transpose

$$\mathbf{R} \cdot \mathbf{R}^T = 1$$

→ thus, the ***rows*** are **{A} written in {B}**

$$^{B}_{A}\mathbf{R} = {}^{A}_{B}\mathbf{R}^T = {}^{A}_{B}\mathbf{R}^{-1}$$

Position and Orientation [8]

- Rotation Formula about the 3 Principal Axes by θ

X:

$$\mathbf{R}_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) \\ 0 & \sin(\theta) & \cos(\theta) \end{bmatrix}$$

Y:

$$\mathbf{R}_y = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix}$$

Z:

$$\mathbf{R}_z = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Euler Angles

- Minimal representation of orientation (α,β,γ)
- Represent a rotation about an axis of a **moving** coordinate frame
  → ${}^{A}_{B}\mathbf{R}$: Moving frame **B** w/r/t fixed A
- The location of the axis of each successive rotation depends on the previous one! …
- So, Order Matters (12 combinations, why?)
- Often Z-Y-X:
  - α: rotation about the **z** axis
  - β: rotation about the rotated **y** axis
  - γ: rotation about the twice rotated **x** axis
- Has singularities! … (e.g., β=±90°)

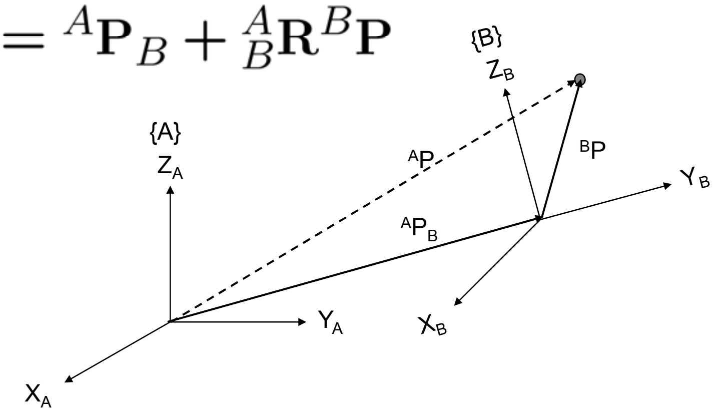
Ex: **X-Y-Z Fixed Angles**

$$
\begin{aligned}
{}^{A}R_{BXYZ}(\gamma,\beta,\alpha) &= R_Z(\alpha)R_Y(\beta)R_X(\gamma) \\
&= \begin{bmatrix} c_\alpha & -s_\alpha & 0 \\ s_\alpha & c_\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_\beta & 0 & s_\beta \\ 0 & 1 & 0 \\ -s_\beta & 0 & c_\beta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_\gamma & -s_\gamma \\ 0 & s_\gamma & c_\gamma \end{bmatrix} \\
&= \begin{bmatrix} c_\alpha c_\beta & c_\alpha s_\beta s_\gamma - s_\alpha c_\gamma & c_\alpha s_\beta c_\gamma + s_\alpha s_\gamma \\ s_\alpha c_\beta & s_\alpha s_\beta s_\gamma + c_\alpha c_\gamma & s_\alpha s_\beta c_\gamma - c_\alpha s_\gamma \\ -s_\beta & c_\beta s_\gamma & c_\beta c_\gamma \end{bmatrix}
\end{aligned}
$$

Coordinate Transformations [3]

- Composite transformation:
  {B} is moved with respect to {A}:

$$^{A}\mathbf{P} = {}^{A}\mathbf{P}_B + {}^{A}_{B}\mathbf{R}\,{}^{B}\mathbf{P}$$



Homogenous Transformation

$$\begin{bmatrix} {}^{A}R_B & {}^{A}p \\ \gamma & \rho \end{bmatrix}$$

- γ is a projective transformation
- ρ is a scaling value:

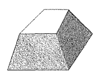
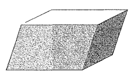
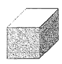
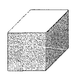
$$\widehat{p} = \begin{bmatrix} \rho p_x & \rho p_y & \rho p_z & \rho \end{bmatrix}^T$$

Projective Transformations …

Group	Matrix	Distortion	Invariant properties
Projective 8 dof	$\begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}$		Concurrency, collinearity, **order of contact**: intersection (1 pt contact); tangency (2 pt contact); inflections (3 pt contact with line); tangent discontinuities and cusps. cross ratio (ratio of ratio of lengths).
Affine 6 dof	$\begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$		Parallelism, ratio of areas, ratio of lengths on collinear or parallel lines (e.g. midpoints), linear combinations of vectors (e.g. centroids). The line at infinity, $\mathbf{l}_\infty$.
Similarity 4 dof	$\begin{bmatrix} sr_{11} & sr_{12} & t_x \\ sr_{21} & sr_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$		Ratio of lengths, angle. The circular points, $\mathbf{I}, \mathbf{J}$ (see section 2.7.3).
Euclidean 3 dof	$\begin{bmatrix} r_{11} & r_{12} & t_x \\ r_{21} & r_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$		Length, area

p.44, R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*

Projective Transformations & Other Transformations of 3D Space

Group	Matrix	Distortion	Invariant properties
Projective 15 dof	$\begin{bmatrix} \mathtt{A} & \mathbf{t} \\ \mathbf{v}^\mathsf{T} & v \end{bmatrix}$		Intersection and tangency of surfaces in contact. Sign of Gaussian curvature.
Affine 12 dof	$\begin{bmatrix} \mathtt{A} & \mathbf{t} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix}$		Parallelism of planes, volume ratios, centroids. The plane at infinity, π_∞, (see section 3.5).
Similarity 7 dof	$\begin{bmatrix} s\mathtt{R} & \mathbf{t} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix}$		The absolute conic, Ω_∞, (see section 3.6).
Euclidean 6 dof	$\begin{bmatrix} \mathtt{R} & \mathbf{t} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix}$		Volume.

p.78, R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*

Denavit-Hartenberg: A Frame Placement Convention

1. Align an axis along principal motion
   1. Rotary (**R**): align rotation axis along the z axis
   2. Prismatic (**P**): align slider travel along x axis

2. Orient so as to position x axis towards next frame

3. θ (rot z) → d (trans z) → a (trans x) → α (rot x)

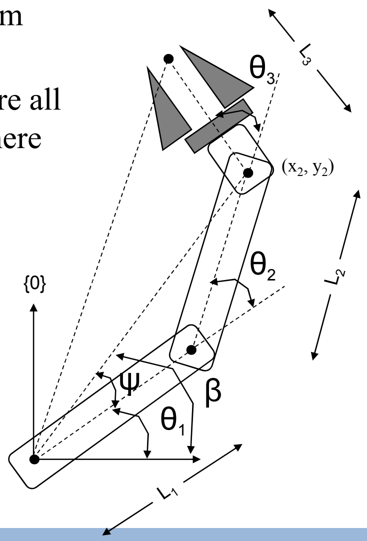
Denavit-Hartenberg → Rotation Matrix

- Each transformation is a product of 4 "basic" transformations (instead of 6)

$$
\begin{aligned}
{}^{i-1}A_i &= Rot_{z,\theta_i} Trans_{z,d_i} Trans_{x,a_i} Rot_{x,\alpha_i} \\
&= \begin{bmatrix} c_{\theta_i} & -s_{\theta_i} & 0 & 0 \\ s_{\theta_i} & c_{\theta_i} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdots \\
&\quad \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_{\alpha_i} & -s_{\alpha_i} & 0 \\ 0 & s_{\alpha_i} & c_{\alpha_i} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
&= \begin{bmatrix} c_{\theta_i} & -s_{\theta_i} c_{\alpha_i} & s_{\theta_i} s_{\alpha_i} & a_i c_{\theta_i} \\ s_{\theta_i} & c_{\theta_i} c_{\alpha_i} & -c_{\theta_i} s_{\alpha_i} & a_i s_{\theta_i} \\ 0 & s_{\alpha_i} & c_{\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned}
$$

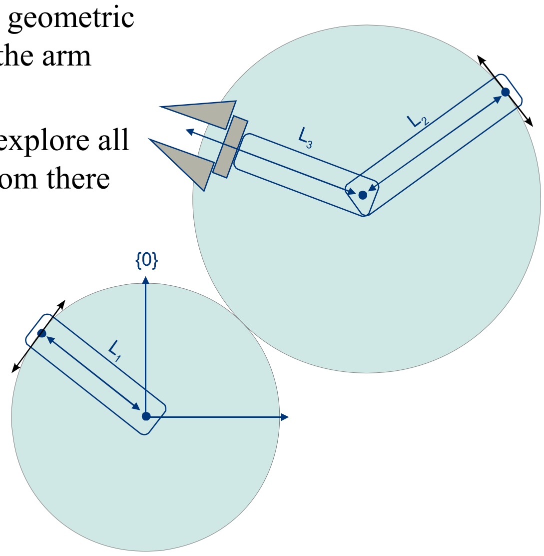
Inverse Kinematics: Geometrical Approach

- We can also consider the geometric relationships defined by the arm
- Start with what is fixed, explore all geometric possibilities from there



Inverse Kinematics: Geometrical Approach [2]

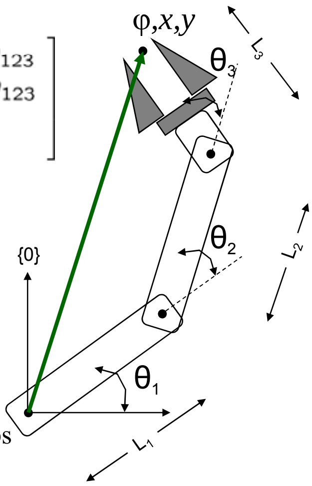
- We can also consider the geometric relationships defined by the arm
- Start with what is fixed, explore all geometric possibilities from there



Inverse Kinematics: Algebraic Approach

- We have a series of equations which define this system
- Recall, from Forward Kinematics:

$$ {}^0T_3 = \begin{bmatrix} c_{\theta_{123}} & -s_{\theta_{123}} & 0 & L_1c_{\theta_1} + L_2c_{\theta_{12}} + L_3c_{\theta_{123}} \\ s_{\theta_{123}} & c_{\theta_{123}} & 0 & L_1s_{\theta_1} + L_2s_{\theta_{12}} + L_3s_{\theta_{123}} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$



- The end-effector pose is given by

$$ {}^0T_3 = \begin{bmatrix} c_\phi & -s_\phi & 0 & x \\ s_\phi & c_\phi & 0 & y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

- Equating terms gives us a set of algebraic relationships

Dynamics

- For Manipulators, the general form is

$$ \tau = M(\Theta)\ddot{\Theta} + V(\Theta, \dot{\Theta}) + G(\Theta) $$

where
  - τ is a vector of joint torques
  - Θ is the nx1 vector of joint angles
  - M(Θ) is the nxn mass matrix
  - V(Θ, Θ)is the nx1 vector of centrifugal and Coriolis terms
  - G(Θ) is an nx1 vector of gravity terms
- Notice that all of these terms depend on Θ so the dynamics varies as the manipulator move

Dynamics – Langrangian Mechanics

- Alternatively, we can use Langrangian Mechanics to compute the dynamics of a manipulator (or other robotic system)
- The Langrangian is defined as the difference between the Kinetic and Potential energy in the system
- Using this formulation and the concept of virtual work we can find the forces and torques acting on the system.
- This may seem more involved but is often easier to formulate for complex systems

$$L = K - P$$

$$\mathbf{F} = \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\mathbf{x}}}\right) - \frac{\partial L}{\partial \mathbf{x}}$$

$$\tau = \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}}\right) - \frac{\partial L}{\partial \theta}$$

Static Forces

- We can also use the Jacobian to compute the joint torques required to maintain a particular force at the end effector
- Consider the concept of virtual work

$$F \cdot \delta\mathbf{X} = \tau \cdot \delta\theta$$

- Or

$$F^T \delta\mathbf{X} = \tau^T \delta\theta$$

- Earlier we saw that

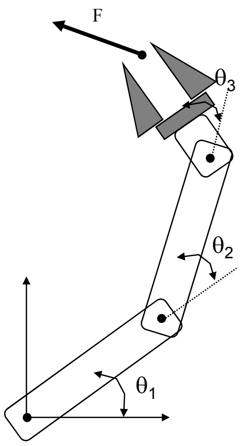
$$\delta\mathbf{X} = \mathbf{J}\delta\theta$$

- So that

$$F^T\mathbf{J} = \tau^T$$

- Or

$$\tau = \mathbf{J}^T F$$



Revisiting The Jacobian

- I told you (Lec 4, Slide 19):

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_i \end{bmatrix} = \begin{bmatrix} \frac{\partial x_1}{\partial \theta_1} & \frac{\partial x_1}{\partial \theta_2} & \cdots & \frac{\partial x_1}{\partial \theta_j} \\ \frac{\partial x_2}{\partial \theta_1} & \frac{\partial x_2}{\partial \theta_2} & \cdots & \frac{\partial x_2}{\partial \theta_j} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial x_i}{\partial \theta_1} & \frac{\partial x_i}{\partial \theta_2} & \cdots & \frac{\partial x_i}{\partial \theta_j} \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \vdots \\ \dot{\theta}_j \end{bmatrix}$$

- True, but we can be more "explicit"

Jacobian: **Explicit Form**

- For a serial chain (robot): The velocity of a link with respect to the proceeding link is dependent on the type of link that connects them
- If the joint is **prismatic** (**ϵ=1**), then $\mathbf{v}_i \quad \frac{dz}{dt}$
- If the joint is **revolute** (**ϵ=0**), then $\frac{d}{dt}$(in the $\hat{k}$ direction)

$$v \quad \sum_{i\,1}^{N} {}_i v_i \quad \bar{\ }_i \quad {}_i \quad \mathbf{p}^i_{i\,1} \qquad \sum_{i\,1}^{N} \bar{\ }_i \,\dot{\boldsymbol{\theta}}_i \qquad \sum_{i\,1}^{N} \bar{\ }_i\mathbf{z}_i \quad \dot{}_i$$

$$v \quad J_v\dot{\mathbf{q}} \qquad\qquad \boldsymbol{\omega} \quad J\,\dot{\mathbf{q}}$$

- Combining them (with **v**=(Δx, Δθ))

$$J \quad \begin{matrix} J_v \\ J \end{matrix}$$

Jacobian: **Explicit Form [2]**

- The overall Jacobian takes the form

$$J \quad \begin{matrix} \frac{x_P}{q_1} & \cdots & \frac{x_P}{q_n} \\ {}^{-}{}_{1}z_1 & \cdots & {}^{-}{}_{1}z_n \end{matrix}$$

- The Jacobian for a particular frame (F) can be expressed:

$${}^{F}J \quad \begin{matrix} \frac{{}^{F}x_P}{q_1} & \cdots & \frac{{}^{F}x_P}{q_n} \\ {}^{-}{}_{1}{}^{F}z_1 & \cdots & {}^{-}{}_{1}{}^{F}z_n \end{matrix}$$

Where: ${}^{F}\mathbf{z}_i \quad {}^{F}_{i}R\,{}^{i}\mathbf{z}_i \quad \& \quad {}^{i}\mathbf{z}_i \quad 0 \quad 0 \quad 1$

Dynamics – Langrangian Mechanics [2]

$L \quad K \quad P, \dot{\boldsymbol{\theta}}$: Generalized Velocities, M : Mass Matrix

$$\boldsymbol{\tau} \quad \sum_{i\,1}^{N} {}_{i} \quad \frac{d}{dt}\frac{K}{\dot{\boldsymbol{\theta}}} \quad \frac{K}{\boldsymbol{\theta}} \quad \frac{P}{\boldsymbol{\theta}}$$

$$K = \tfrac{1}{2}\dot{\theta}^T M(\theta)\dot{\theta}$$

$$\frac{d}{dt}\left(\frac{\partial K}{\partial \dot{\theta}}\right) = \frac{d}{dt}\left(\frac{\partial}{\partial \dot{\theta}}\left(\tfrac{1}{2}\dot{\theta}^T M(\theta)\dot{\theta}\right)\right) = \frac{d}{dt}\left(M\dot{\theta}\right) = M\ddot{\theta}+\dot{M}\dot{\theta}$$

$$\rightarrow \frac{d}{dt}\left(\frac{\partial K}{\partial \dot{\theta}}\right) - \frac{\partial K}{\partial \theta} = \left[M\ddot{\theta}+\dot{M}\dot{\theta}\right] - \left[\tfrac{1}{2}\dot{\theta}^T M(\theta)\dot{\theta}\right] = M\ddot{\theta} + \underbrace{\left\{\dot{M}\dot{\theta} - \tfrac{1}{2}\begin{bmatrix} \dot{\theta}^T \frac{\partial M}{\partial \theta_1}\dot{\theta} \\ \vdots \\ \dot{\theta}^T \frac{\partial M}{\partial \theta_n}\dot{\theta} \end{bmatrix}\right\}}_{\mathrm{v}(\theta,\dot{\theta})}$$

$$\mathrm{v}\left(\theta,\dot{\theta}\right) = \underbrace{C(\theta)\left[\dot{\theta}^2\right]}_{\text{Centrifugal}} + \underbrace{B(\theta)\left[\dot{\theta}\dot{\theta}\right]}_{\text{Coriolis}}$$

$$\boldsymbol{\tau} \quad M \quad \ddot{\boldsymbol{\theta}} \quad \mathbf{v}\ \boldsymbol{\theta},\dot{\boldsymbol{\theta}} \quad \mathbf{g}\ \boldsymbol{\theta}$$

Dynamics – Langrangian Mechanics [3]

- The Mass Matrix: Determining via the Jacobian!

$$K = \sum_{i=1}^{N} K_i$$

$$K_i = \tfrac{1}{2}\left(m_i v_{C_i}^T v_{C_i} + \omega_i^T I_{C_i}\omega_i\right)$$

$$v_{C_i} = J_{v_i}\dot{\theta} \qquad J_{v_i} = \begin{bmatrix} \frac{\partial \mathbf{p}_{C_1}}{\partial \theta_1} & \cdots & \frac{\partial \mathbf{p}_{C_i}}{\partial \theta_i} & \underbrace{0}_{i+1} & \cdots & \underbrace{0}_{n} \end{bmatrix}$$

$$\omega_i = J_{\omega_i}\dot{\theta} \qquad J_{\omega_i} = \begin{bmatrix} \bar{\varepsilon}_1 Z_1 & \cdots & \bar{\varepsilon}_i Z_i & \underbrace{0}_{i+1} & \cdots & \underbrace{0}_{n} \end{bmatrix}$$

$$\therefore M = \sum_{i=1}^{N}\left(m_i J_{v_i}^T J_{v_i} + J_{\omega_i}^T I_{C_i} J_{\omega_i}\right)$$

! M is symmetric, positive definite $\quad m_{ij} \quad m_{ji}, \dot{\boldsymbol{\theta}}^T M \dot{\boldsymbol{\theta}} \quad 0$

Generalized Coordinates

- A significant feature of the Lagrangian Formulation is that any convenient coordinates can be used to derive the system.
- Go from Joint → Generalized
  - Define **p**: $\quad d\mathbf{p} \quad \mathbf{J}d\mathbf{q}$

$$\mathbf{q} \quad q_1 \quad \cdots \quad q_n \qquad \mathbf{p} \quad p_1 \quad \cdots \quad p_n$$

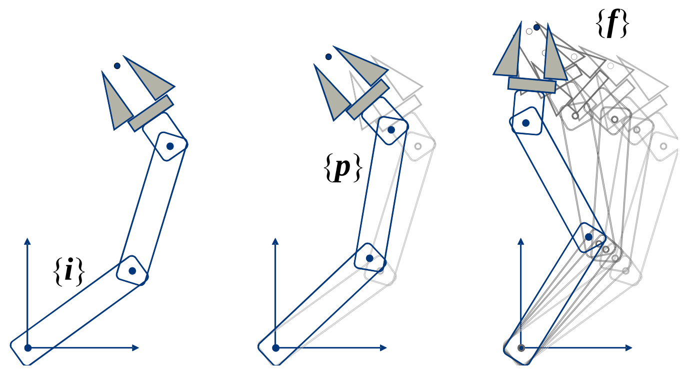
➔Thus: the kinetic energy and gravity terms become

$$KE \quad \tfrac{1}{2}\dot{\mathbf{p}}^T\mathbf{H}^*\dot{\mathbf{p}} \qquad \mathbf{G}^* \quad \mathbf{J}^{\ 1\ T}\,\mathbf{G}$$

where: $\mathbf{H}^* \quad \mathbf{J}^{\ 1\ T}\mathbf{H}\mathbf{J}^{\ 1}$

Trajectory Generation

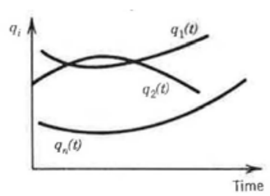
- The goal is to get from an initial position $\{\boldsymbol{i}\}$ to a final position $\{\boldsymbol{f}\}$ via a path points $\{\boldsymbol{p}\}$



Joint Space

Consider only the **joint positions** as a function of time

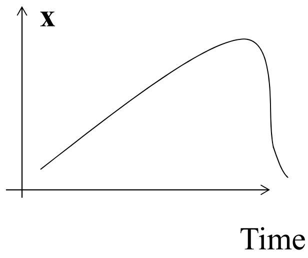
- + Since we control the joints, this is more direct
- -- If we want to follow a particular trajectory, <u>not easy</u>
  - at best lots of intermediate points
  - No guarantee that you can solve the Inverse Kinematics for all path points



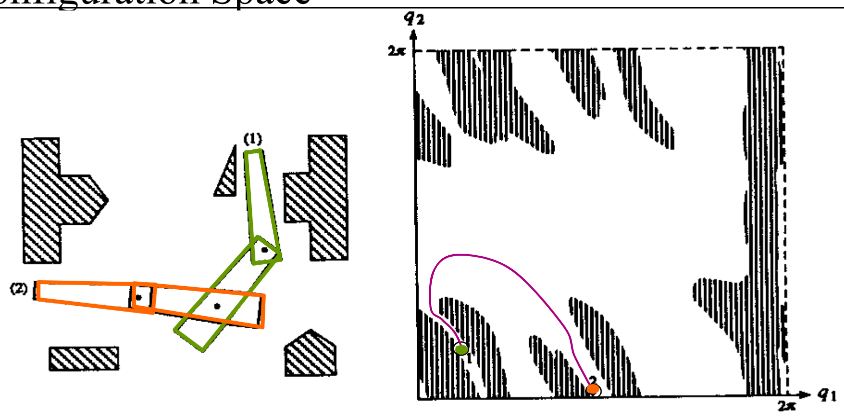
Cartesian Workspace

Consider the **Cartesian positions** as a function of time

- + Can track shapes exactly
- -- We need to solve the inverse kinematics and dynamics



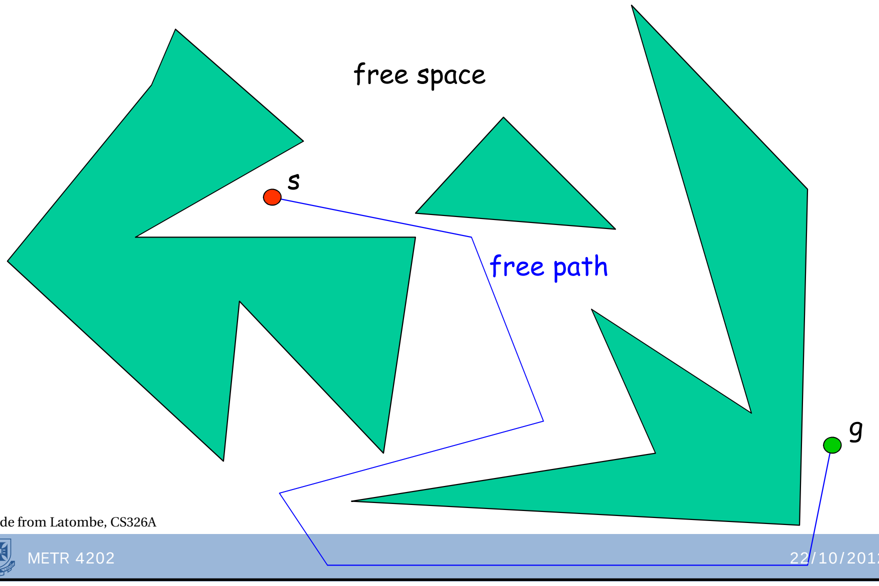
Configuration Space



- A robot configuration is a specification of the positions of all robot points relative to a fixed coordinate system
- Usually a configuration is expressed as a "vector" of position/orientation parameters

Slide from Latombe, CS326A

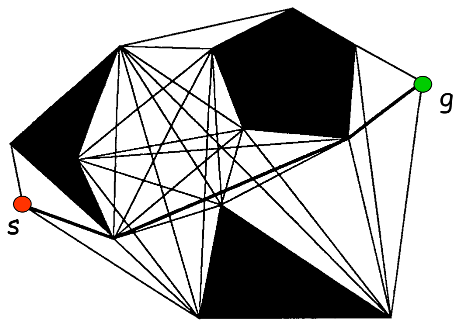
Motion Planning



Slide from Latombe, CS326A

Roadmap Methods

- **Visibility graph**
  Introduced in the Shakey project at SRI in the late 60s.
  Can produce shortest paths in 2-D configuration spaces

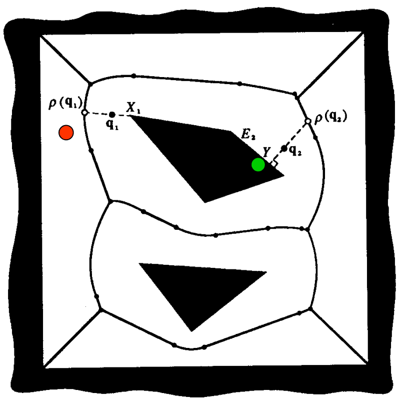


Slide from Latombe, CS326A

Roadmap Methods

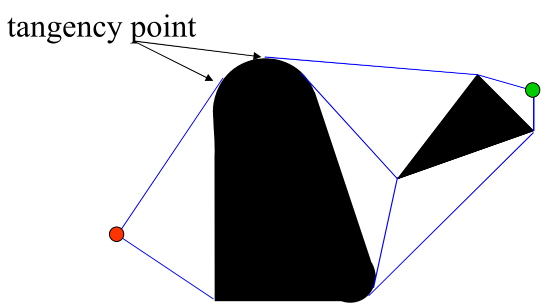
- Voronoi diagram
  Introduced by
  Computational
  Geometry researchers.
  Generate paths that
  maximizes clearance.

  O(n log n) time
  O(n) space



Slide from Latombe, CS326A

Generalized (Reduced) -- Visibility Graph

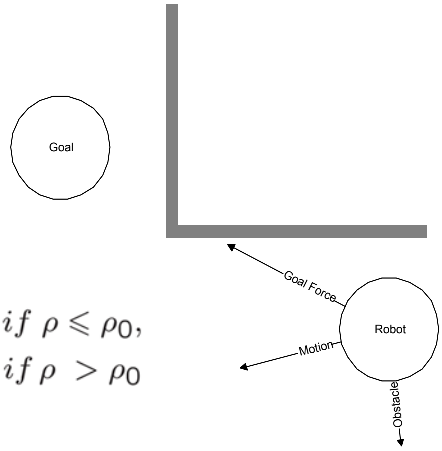


Slide from Latombe, CS326A

Potential Field Methods

- Approach initially proposed for real-time collision avoidance [Khatib, 86]

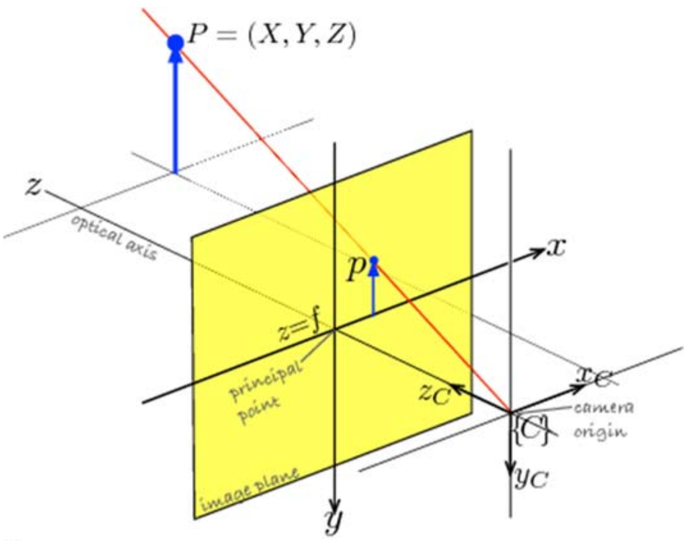
$$F_{Goal} = -k_p(x - x_{Goal})$$



$$F_{Obstacle} = \begin{cases} \eta \left(\frac{1}{\rho} - \frac{1}{\rho_0}\right) \frac{1}{\rho^2} \frac{\partial \rho}{\partial x} & if \ \rho \leqslant \rho_0, \\ 0 & if \ \rho \ > \rho_0 \end{cases}$$

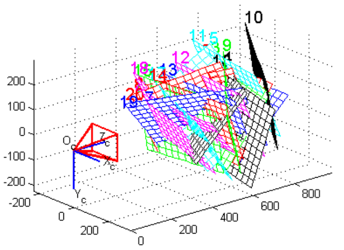
Slide based on Latombe, CS326A

Image Formation



Corke, Ch. 11

Camera calibration



- Five intrinsic parameters:
  - (f_x, f_y) : Focal length
  - (o_x, o_y) : Image center
  - α: skew
- Extrinsic Parameters:
  - Tx, Ty, Tz: Translation
  - ϕ, φ, ψ: Orientation

$$\begin{pmatrix} \hat{X}^C \\ \hat{Y}^C \\ \hat{Z}^C \end{pmatrix} = R \begin{pmatrix} X^W \\ Y^W \\ Z^W \end{pmatrix} + \begin{pmatrix} T_X \\ T_Y \\ T_Z \end{pmatrix}$$

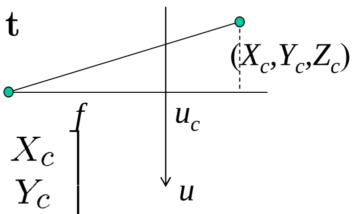
$$R = \begin{pmatrix} \cos\phi & \sin\phi & 0 \\ -\sin\phi & \cos\phi & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \cos\varphi & 0 & \sin\varphi \\ 0 & 1 & 0 \\ -\sin\varphi & 0 & \cos\varphi \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\psi & \sin\psi \\ 0 & -\sin\psi & \cos\psi \end{pmatrix}$$

$$x_{im} \quad f_x \frac{X^C}{Z^C} \quad o_x \quad 0.5$$

$$y_{im} \quad f_y \frac{Y^C}{Z^C} \quad o_y \quad 0.5$$

Image formation equations

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = \begin{bmatrix} \mathbf{R} \end{bmatrix}_{3\times 3} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \mathbf{t}$$



$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \sim \begin{bmatrix} U \\ V \\ W \end{bmatrix} = \begin{bmatrix} f & 0 & u_c \\ 0 & f & v_c \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix}$$

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \sim \begin{bmatrix} f & 0 & u_c \\ 0 & f & v_c \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = \mathbf{K}\ \mathbf{X}_c$$

$$\boxed{\mathbf{K} = \begin{bmatrix} fa & s & u_c \\ 0 & f & v_c \\ 0 & 0 & 1 \end{bmatrix}}$$

From Szeliski, Computer Vision: Algorithms and Applications

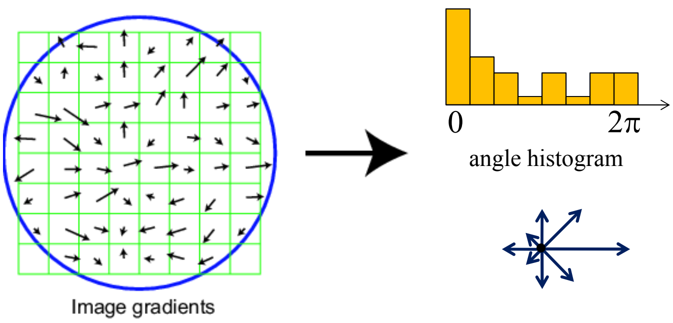
Harris Corner Detectors

Adopted from S. Lazebnik, Gang Hua (CS 558)

Scale Invariant Feature Transform

Basic idea:

- Take 16x16 square window around detected feature
- Compute edge orientation (angle of the gradient - 90°) for each pixel
- Throw out weak edges (threshold gradient magnitude)
- Create histogram of surviving edge orientations



Adapted from slide by David Lowe

SIFT descriptor

Full version

- Divide the 16x16 window into a 4x4 grid of cells (2x2 case shown below)
- Compute an orientation histogram for each cell
- 16 cells * 8 orientations = 128 dimensional descriptor

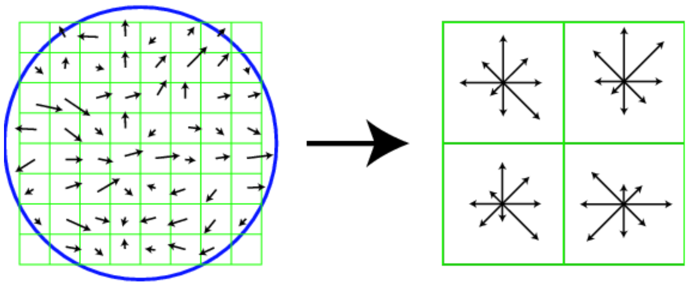


Image gradients — Keypoint descriptor

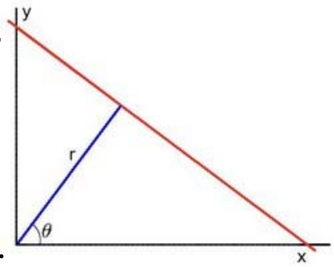
Adapted from slide by David Lowe

Line Detection – Hough Lines [1]

- A line in an image can be expressed as two variables:
  - **Cartesian coordinate system:** m,b
  - **Polar coordinate system:** r, θ
    - ➔ **<u>avoids problems with vert. lines</u>**

y=**m**x+**b** ➔

$$y = \left(-\frac{\cos\theta}{\sin\theta}\right)x + \left(\frac{r}{\sin\theta}\right)$$



- For each point (x_1, y_1) we can write:

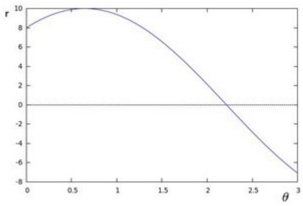
$$r = x_1 \cos\theta + y_1 \sin\theta$$

- Each pair (r,θ) represents a line that passes through (x_1, y_1)

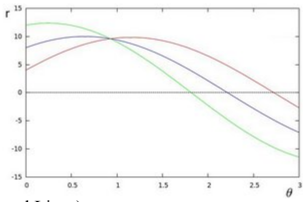
See also OpenCV documentation (cv::HoughLines)

Line Detection – Hough Lines [2]

- Thus a given point gives a sinusoid



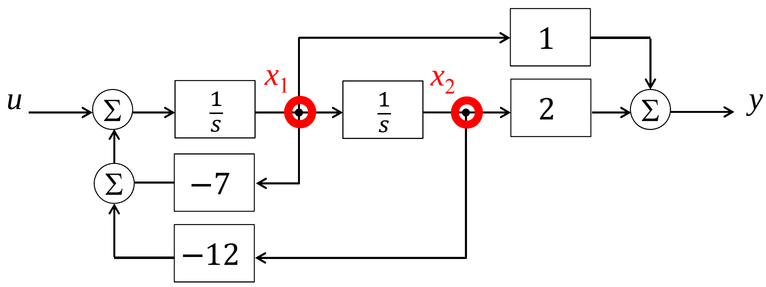
- Repeating for all points on the image



See also OpenCV documentation (cv::HoughLines)

Finding states

- We can identify nodes of a dynamic system which contain the integrated time-history values of the system response
  - We call them "states"



Linear system equations

- We can represent the dynamic relationship between the states with a linear system:

$$
\begin{aligned}
\dot{x_1} &= -7x_1 - 12x_2 + u \\
\dot{x_2} &= x_1 + 0x_2 + 0u \\
y &= x_1 + 2x_2 + 0u
\end{aligned}
$$

State-space representation

- We can write linear systems in matrix form:

$$
\begin{aligned}
\dot{\boldsymbol{x}} &= \begin{bmatrix} -7 & 12 \\ 1 & 0 \end{bmatrix} \boldsymbol{x} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u \\
\boldsymbol{y} &= \begin{bmatrix} 1 & 2 \end{bmatrix} \boldsymbol{x} + 0u
\end{aligned}
$$

Or, more generally:

$$
\left.\begin{aligned}
\dot{\boldsymbol{x}} &= \mathbf{A}\boldsymbol{x} + \mathbf{B}u \\
y &= \mathbf{C}\boldsymbol{x} + Du
\end{aligned}\right\} \text{"State-space equations"}
$$

State-space representation

- State-space matrices are not necessarily unique representations of a dynamic system
  - There are several common forms (here's two)
- <u>Control canonical form</u>
  - Each node – each entry in $\boldsymbol{x}$ – represents a state of the system (each order of s maps to a state)
- <u>Modal form</u>
  - Diagonals of the state matrix **A** are the poles ("modes") of the transfer function

Control canonical form

- CCF matrix representations have the following structure:

$$\mathbf{A} = \begin{bmatrix} -a_1 & -a_1 & \cdots & -a_{n-2} & -a_{n-1} & -a_n \\ 1 & 0 & & 0 & 0 & 0 \\ 0 & 1 & & & & \\ \vdots & & \ddots & & & \vdots \\ & & & 1 & 0 & 0 \\ 0 & 0 & \cdots & 0 & 1 & 0 \end{bmatrix}$$

Pretty diagonal!

Modal form

- MF matrix representations have the following structure:

$$\mathbf{A} = \begin{bmatrix} -p_1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & -p_2 & & & & 0 \\ \vdots & & \ddots & & & \vdots \\ 0 & & & -p_{n-2} & & 0 \\ 0 & & & & -p_{n-1} & 0 \\ 0 & 0 & \cdots & 0 & 0 & -p_n \end{bmatrix}$$

Also pretty diagonal!

Controllability matrix

- To convert an arbitrary state representation in **F**, **G**, **H** and *J* to control canonical form **A**, **B**, **C** and *D*, the controllability matrix

$$\mathcal{C} = [\mathbf{G} \quad \mathbf{FG} \quad \mathbf{F}^2\mathbf{G} \quad \cdots \quad \mathbf{F}^{n-1}\mathbf{G}]$$

must be invertible (i.e. full rank).

- If you can write it in CCF, then the system equations must be linearly independent.
  - Thus, a invertible controllability matrix means $\boldsymbol{x}$ can be driven to any value.

State evolution

- Consider the system matrix relation:

$$\dot{\boldsymbol{x}} = \mathbf{F}\boldsymbol{x} + \mathbf{G}u$$
$$y = \mathbf{H}\boldsymbol{x} + Ju$$

The time solution of this system is:

$$\boldsymbol{x}(t) = e^{\mathbf{F}(t-t_0)}\,\boldsymbol{x}(t_0) + \int_{t_0}^{t} e^{\mathbf{F}(t-\tau)}\mathbf{G}u(\tau)d\tau$$

If you didn't know, the matrix exponential is:

$$e^{\mathbf{K}t} = \mathbf{I} + \mathbf{K}t + \frac{1}{2!}\mathbf{K}^2t^2 + \frac{1}{3!}\mathbf{K}^3t^3 + \cdots$$

System poles are the Eigenvalues of $\mathbf{F}$, (p_i)

Stability

- We can solve for the natural response to initial conditions $\boldsymbol{x}_0$:

$$\boldsymbol{x}(t) = e^{p_i t}\boldsymbol{x}_0$$
$$\therefore \dot{\boldsymbol{x}}(t) = p_i e^{p_i t}\boldsymbol{x}_0 = \boxed{\mathbf{F}e^{p_i t}\boldsymbol{x}_0}$$

homogenous response

A system will be stable provided $\mathrm{eig}(\mathbf{F}) < 0$

Characteristic polynomial

- From this, we can see $\mathbf{F}\boldsymbol{x}_0 = p_i \boldsymbol{x}_0$

$$\text{or, } (p_i \mathrm{I} - \mathbf{F})\boldsymbol{x}_0 = 0$$

$$\text{which is true only when } \det(p_i \mathrm{I} - \mathbf{F})\boldsymbol{x}_0 = 0$$

Aka. the characteristic equation!

- We can reconstruct the CP in s by writing:

$$\det(s\mathrm{I} - \mathbf{F})\boldsymbol{x}_0 = 0$$

Feedback control

- Given $\dot{\boldsymbol{x}} = \mathbf{F}\boldsymbol{x} + \mathbf{G}u$, if we know $\mathbf{F}$ and $\mathbf{G}$, we can design a controller $u = -\mathbf{K}\boldsymbol{x}$ such that

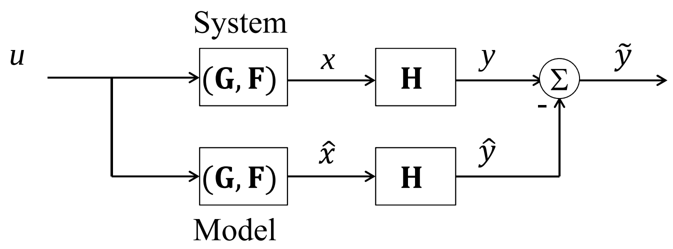
$$\mathrm{eig}(\mathbf{F} - \mathbf{GK}) < 0$$

- If we have full measurement and control of the states of $\boldsymbol{x}$, we can position poles of the closed-loop system in arbitrary locations
  - Modal form makes this straight forward:

$$\mathbf{F} - \mathbf{GK} = \begin{bmatrix} -p_1 - G \cdot K_{1j} & & \\ & -p_2 - G \cdot K_{2j} & \\ & & -p_3 - G \cdot K_{3j} \end{bmatrix}$$

Observers

- Observers (aka “estimators”) are used to infer the hidden states of a system from measured outputs.



A controller is designed using estimates in lieu of full measurements

Observers

- The state estimate can be treated like a control system itself

  – Dynamics to update the estimate:

$$\dot{\hat{x}} = \mathbf{F}\hat{x}+\mathbf{G}\mathrm{u}$$

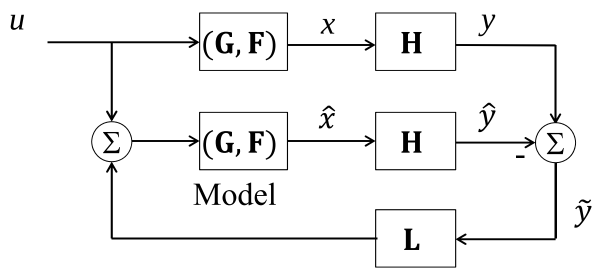
  – By measuring an ‘error signal’ from the difference between the real output measurement and the output estimate, $\tilde{x} = x - \hat{x}$, the state estimate can be shown to converge

Observers

- Just like you might expect:

$$\dot{\hat{x}} = \mathbf{F}\hat{x} + \mathbf{G}u + \mathbf{L}(y - \mathbf{H}\hat{x})$$
$$\therefore \dot{\tilde{x}} = (\mathbf{F} - \mathbf{L}\mathbf{H})\tilde{x}$$

Choose **L** to make $\tilde{x}$ converge to 0



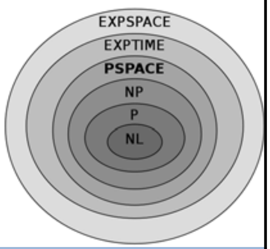
Observability

- The ability to infer these values is called "Observability"
  - This is the dual of controllability; a system that is observable is also controllable and vice versa.
  - Observability matrix:

$$\mathcal{O} = \begin{bmatrix} \mathbf{H} \\ \mathbf{HF} \\ \vdots \\ \mathbf{HF}^{n-1} \end{bmatrix}$$

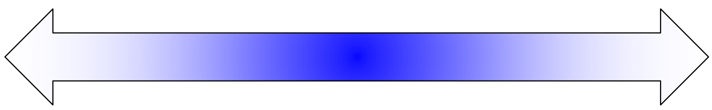
Motion planning under uncertainty

- Goal: Find a motion strategy to accomplish a given task, where there's one or more combination of:
  - Motion error.
  - Sensing error.
  - Model error: Partially known / imperfectly known / unknown operating environment.
- Motion planning under uncertainty in 3D C-space:
  - PSPACE-hard [Natarajan'86].
  - NEXPTIME-hard [Cany & Reif'87].



Several approaches

- Algorithms vs. heuristics.



Conservative
Overestimate risk.

Algorithms:
•Enlarging obstacles.
•Pre-image backchaining.

Probabilistic
Quantify uncertainty, to tradeoff risk w. achieving goal.

POMDP Algorithms.

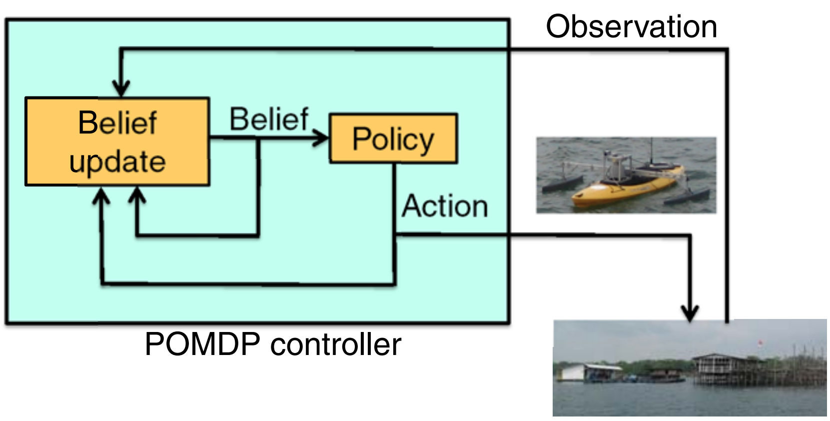
Reactive
Underestimate difficulty of achieving goal.

Reactive greedy heuristics, base on velocity obstacle.

Partially Observable Markov Decision Processes (POMDP)

- Defined as 6 tuples (S, A, Ω, T, Z, R):
  - State space (S). Not known
  - Action space (A).
  - Observation space (Ω).
  - Transition function T(s, a, s').
  - Observation function Z(s, a, o).
  - Reward function R(s, a).
- Belief: Distribution over states.
- Goal: Find an optimal policy.
  - Policy: Mapping from beliefs to actions.

POMDP policy: Usage



Computing an optimal policy

- Dynamic programming to construct optimal value function: Value iteration algorithm.
- Starts from computing the optimal value for 1 step.
- Subsequently compute the optimal value for step-2, step-3, …, step-n for all beliefs, using:

$$V_i^*(b) = HV_{i-1}(b)$$
$$= \max_a R(b,a) + \gamma \sum_o P(o \mid b,a) V_{i-1}^*(T(b,a,o))$$

where T(b, a, o) = b',

$$b'(s') = \frac{Z(s',a,o)\sum_{s\in S} T(s,a,s')b(s)}{P(o \mid a,b)}$$

Example: Tiger problem

A robot is sent to free kitty the kitten from an evil taxidermist. The taxidermist has two rooms for storing animals he is going to "preserve". One room contains kitty, while the other contains a ferocious tiger. The robot doesn't know which room kitty is in and needs to avoid opening the door containing the tiger. It can listen to improve its knowledge of where kitty is.
However, it can only localize the e
correctly 85% of the time. e
listening costs -1, opening
tiger's room costs -20, opening
room gives a reward of 10.
POMDP model ? What's the optimal
policy for 1 planning horizon?

SECaT Time! … Brought To You By the Number 5