MARYLAND
ROBOTICS CENTER
THE INSTITUTE FOR SYSTEMS RESEARCH

ENPM696: Planning for Autonomous Robots

Spaces, Geometric Modeling, and Rigid Body Transformations

Dr. Michael Otte

Slides by S. K. Gupta, as noted

Spring 2017
University of Maryland

Course Project

Still deciding what to do? here are some things to consider:

1. What do you find more interesting?
   A. manipulators (arms etc.)
   B. vehicles
2. What type of problems do you find most interesting?
   A. so hard that we are happy to find any solution.
   B. real-time
   C. optimize vs. some metric
3. Do you find any of these problem classes interesting?
   A. problems with multiple robots
   B. problems with moving obstacles
   C. problems with adversaries
   D. problems with environments that change

Course Project

Still deciding what to do? here are some things to consider:

4. Would you rather:
   A. Implement one or two methods in code?
      i. On hardware or in simulation?
   B. (OR) become an expert on the state-of-the-art for a particular class of algorithms/problems?
      i. Do you prefer taxonomy or compare and contrast?
5. What is your background (M.E., C.S, E.E., etc.)?
6. Why are you in the robotics program?
   1. (are you aiming for an engineering position? R&D? applied research, pure research? not sure?)

Course Project, Project Assignment #P1

Get LaTex Working and submit the PDF file online on Campus

Note: all group members do this assignment separately

Assigned: February 6 (today)

Due: February 13 at 11:59 PM (~ 1 week from today)

1. Download IEEE 2 column conference format
   - http://ras.papercept.net/conferences/support/tex.php
   - Also available on the course Campus page
2. Make yourself the author and create a new title
3. Using the \label{} and \ref{} commands, make inline text markup reference to the following 3 things:
   1. A section header.
   2. Your favorite equation using LaTex math within the \begin{equation} environment.
   3. Add a figure of a robot complete with caption.
4. Get BibTex bibliography working and \cite{} at least 3 references

Course Project, Project Assignment #P1

Resources:

- The not so short introduction to latex2E
  - Online and on course Campus page
- Latex math wikis
  - https://en.wikibooks.org/wiki/LaTeX/Mathematics
  - http://web.ift.uib.no/Teori/KURS/WRK/TeX/symALL.html
- Latex Compilers
  - On the cloud: https://www.overleaf.com/
    - **This is recommended unless you already have LaTex experience**
    - Edit online using a browser
    - Download the pdf
    - "good" error message support
    - This is super nice and easy for anybody that does not do latex often
  - Ubuntu: apt-get install texlive-full
  - Windows: https://miktex.org/
  - Apple: MacTeX

Course Project, Project Assignment #P1

Getting links and references to work if you compile yourself:

“your_paper.tex”

:> pdflatex your_paper

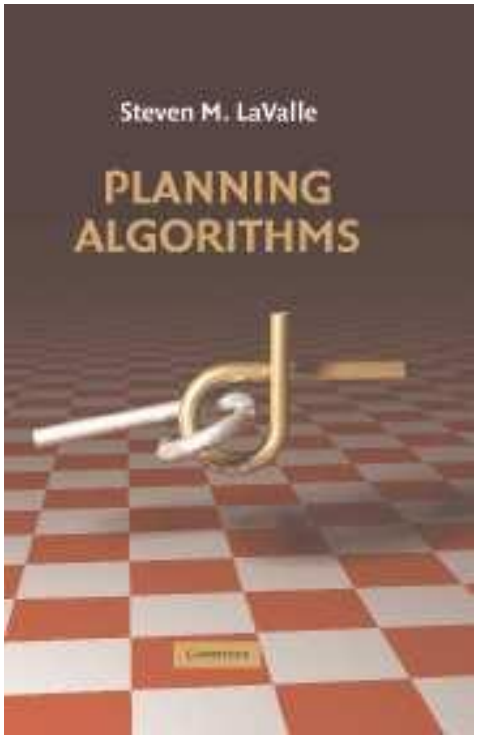
:> bibtex you_paper

:> pdflatex your_paper

:> pdflatex your_paper

This is necessary because pdftex is inherently a single-pass program, so we need to manually help it do multi-pass to make reference links and bibtex citations work.

Reading that Corresponds to today's lecture



(chapters based on website version)

Chapter 3, Geometric Representations and Transformations

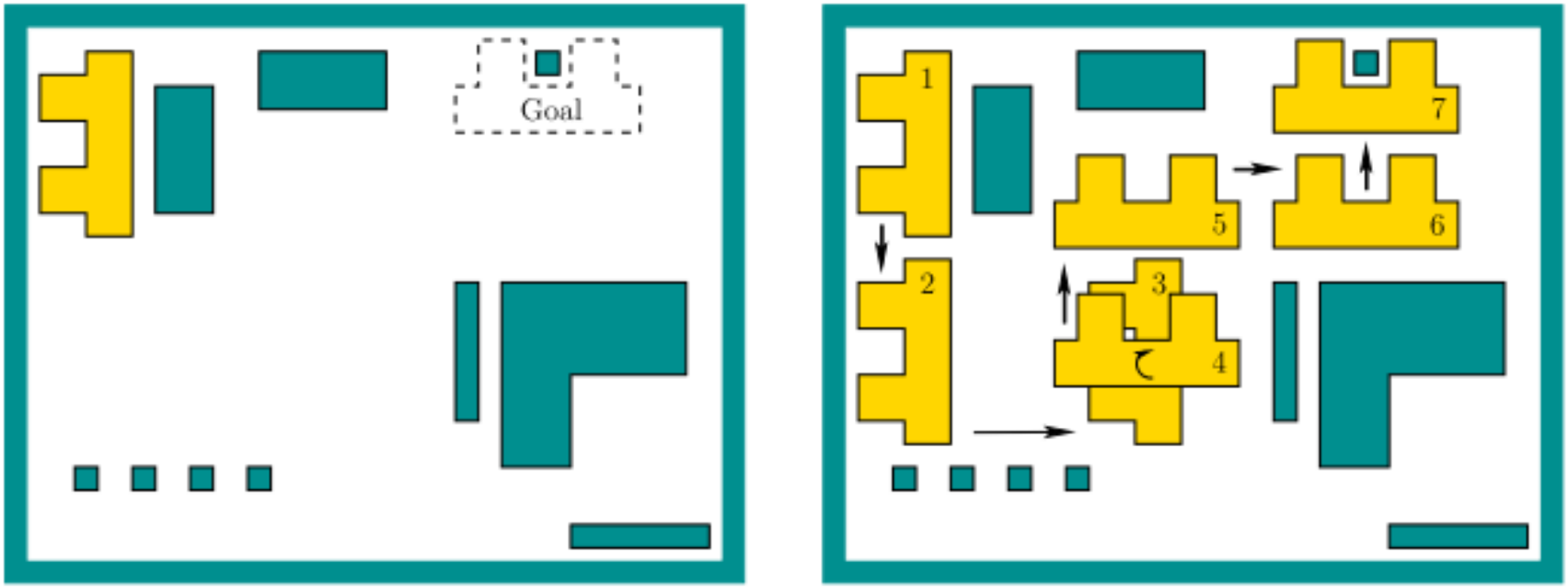
Chapter 4.2-4.4 (and 4.1 if you need it) Configuration Space

Chapter 15.3 Optimal Paths for Some Wheeled Vehicles

Switch to other slides (lecture 2A)

Need for Geometric Modeling

- Produce a motion plan that consists of robot's collision-free motions to the goal



(Source: LaValle, S. M. Motion Planning – The Essentials, 2011)

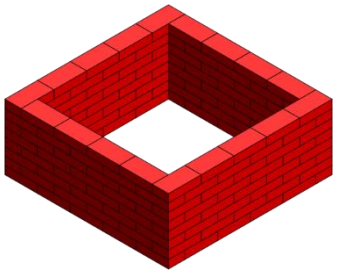
Left: Start (yellow) and goal (dashed)

Right: Workspace projection at 7 different times:

slide 80% by S. K. Gupta

Geometric Modeling

- Need to model
  - Robot
  - Obstacles
    - Walls, etc.
    - Negative obstacles (holes, rivers, etc.)
    - Other Robots, etc.
  - Motion: robot vs. environment and obstacles



- Need to identify points/trajectories where the robot can be placed without colliding with the obstacles.

Slide 60% by S. K. Gupta

Geometric Modeling (Cont.)

- Workspace W
  - Usually $W = \mathbb{R}^2$ or $W = \mathbb{R}^3$.
- Robot $A(q) \subseteq W$.
  - q is a robot's configuration/geometric transformation.
- Obstacles $O \subset W$.
- Robot and obstacles:
  - open or closed subsets of space.
  - (depending on a method's theoretical requirements).
- Constructed from geometric primitives.
  - In practice: should be computationally efficient and capture necessary details.

slide 80% by S. K. Gupta

Representing Geometric Objects

slide by S. K. Gupta

Where Models Come From?

- CAD
  - Mechanical engineering CAD tools
  - Industrial design CAD tools
  - Architecture CAD tools
- Scanners
  - Laser
  - Stereo camera

slide by S. K. Gupta

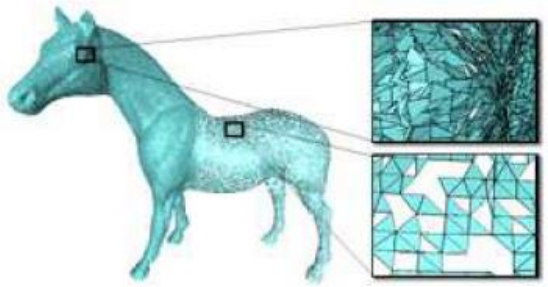
Modeling Options

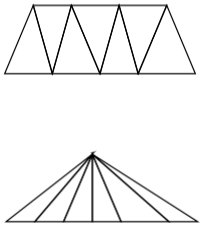
- Point clouds
- Surfaces
  - Polygonal Mesh
  - Non-uniform rational B-splines (NURBS)
- Solids
  - Pixels, Voxels, and Quadtrees
  - Shape Primitives
  - Solids using boundary representations
    - Half-plane representations

Slide 80% by S. K. Gupta

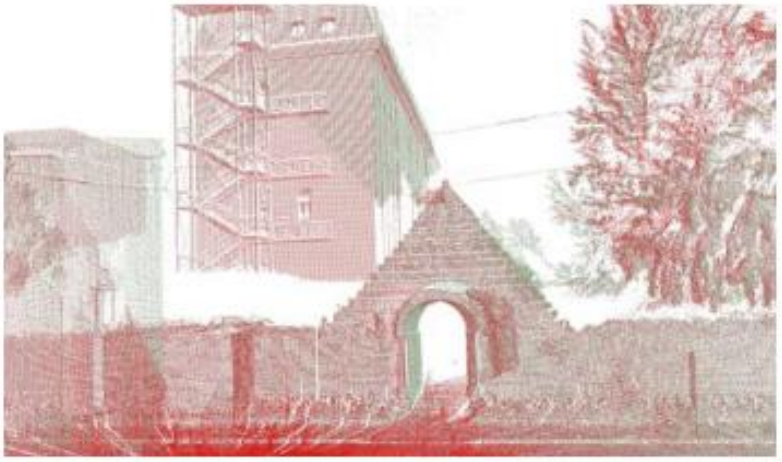
Popular Models

- 3D triangles
  - Polygon soup





- Point clouds



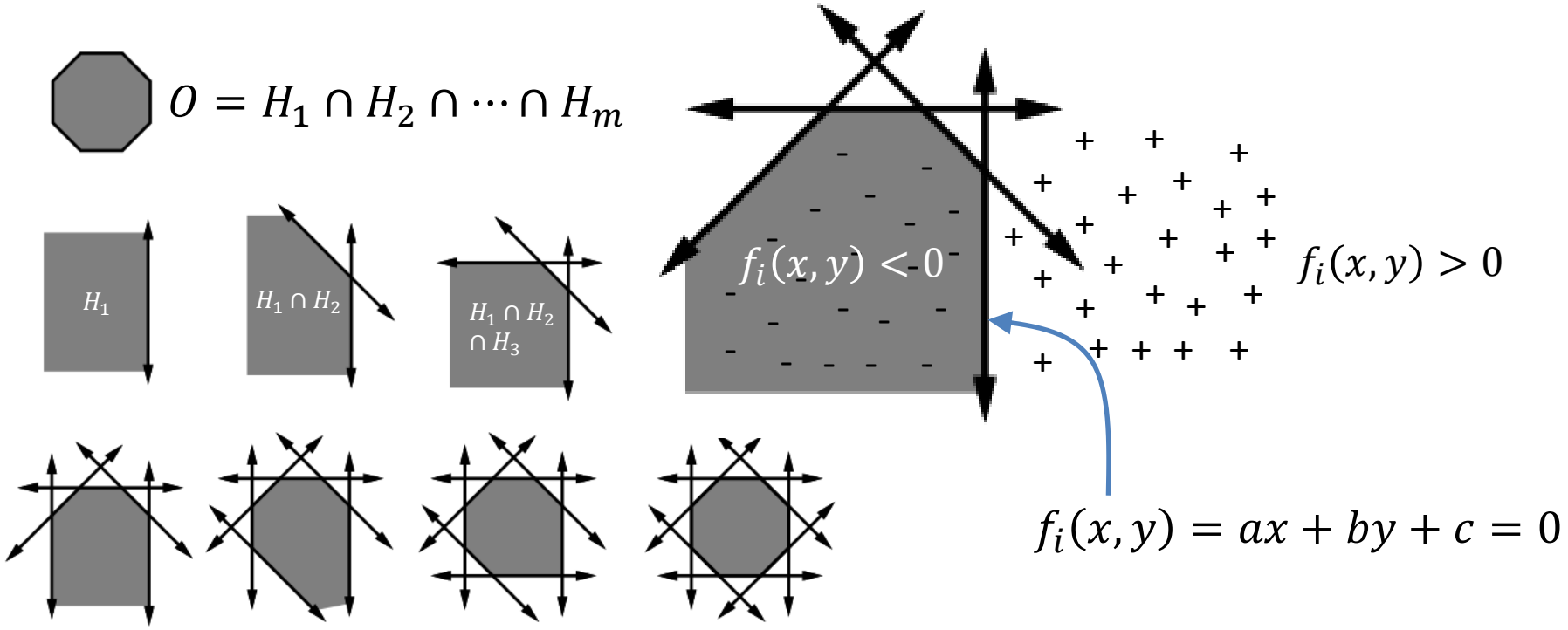
(Source: LaValle, S. M. ICRA 2012 Motion Planning Tutorial, 2012)

slide by S. K. Gupta

Polygonal Models

- Solid representation of convex polygon O as Boolean combination of half-plane primitives

$$H_i = \{(x, y) \in W | f_i(x, y) \leq 0\}$$

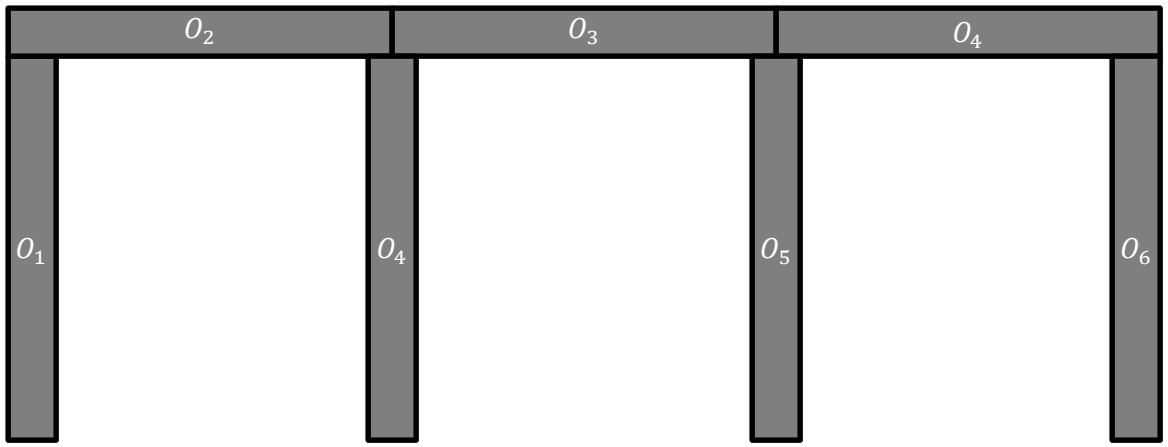


(Adapted from: LaValle, S. M. Planning Algorithms, 2006)

slide by S. K. Gupta

Polygonal Models (Cont.)

- Solid representation of non-convex polygon
  $O = O_1 \cup O_2 \cup \cdots \cup O_n$ as union of convex polygons O_i



slide by S. K. Gupta

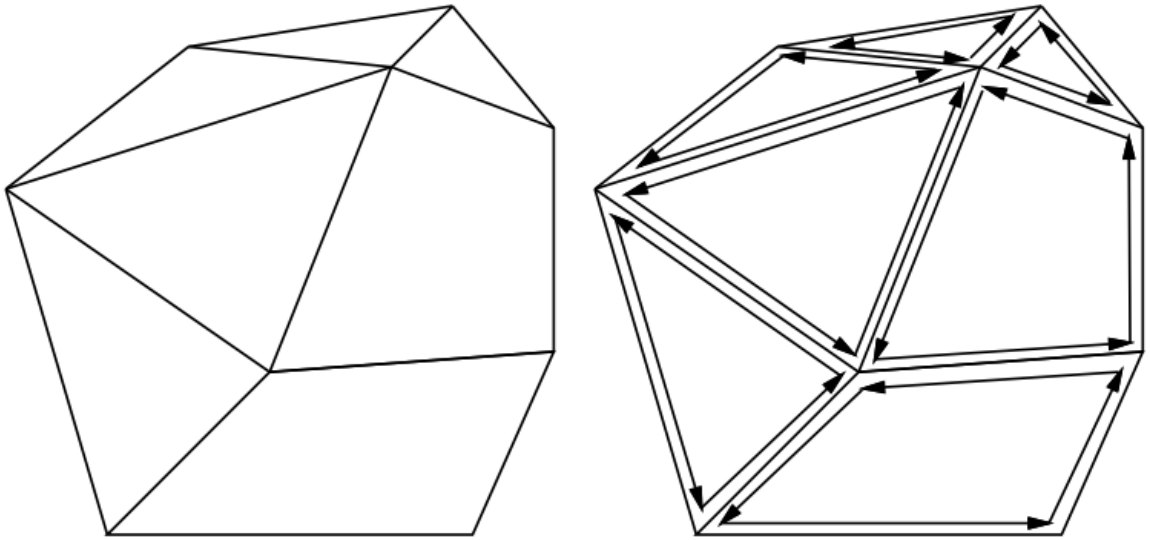
Polyhedral Models

- Solid representation of convex polyhedra O as a Boolean combination of half-space primitives

$$H_i = \{(x, y, z) \in W | f_i(x, y, z) \leq 0\}$$

  - Doubly connected edge list data structure

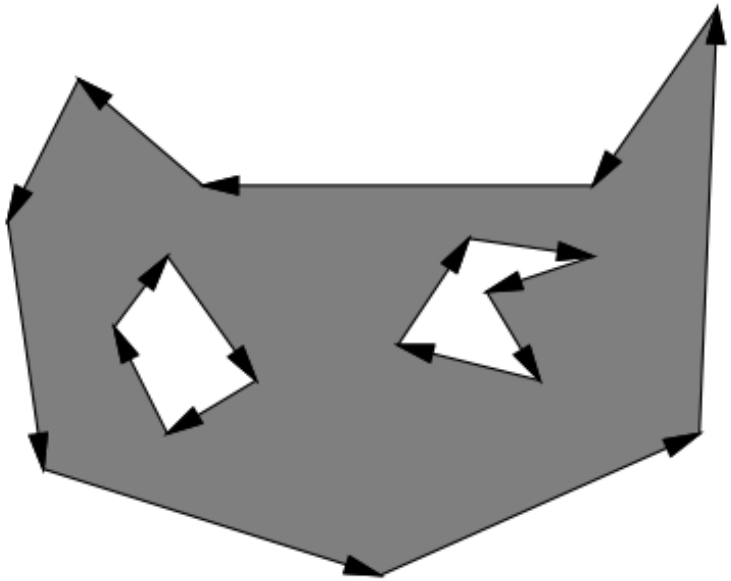
$$f_i(x, y, z) = ax + by + cz + d = 0$$

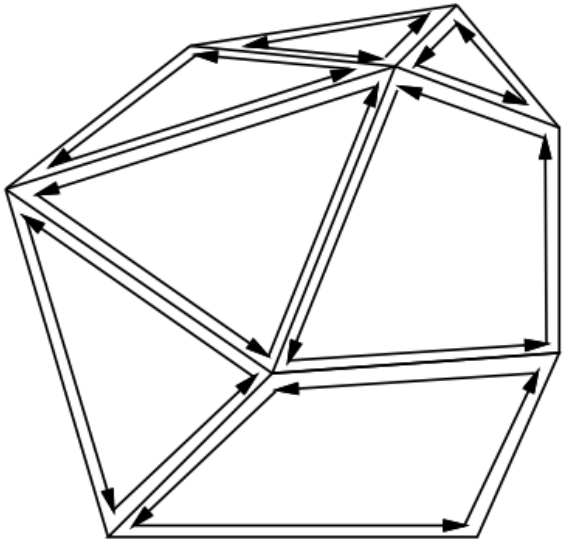


(Source: LaValle, S. M. Planning Algorithms, 2006)

slide by S. K. Gupta

Non-Convex Polygons and Polyhedra





(Source: LaValle, S. M. Motion Planning – The Essentials, 2011)

Semi-Algebraic Models

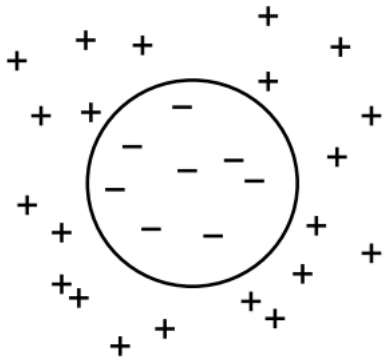
- Solid representation of semi-algebraic set O as Boolean combination of algebraic primitives

$$H_i = \{(x, y) \in W | f_i(x, y) \leq 0\}$$

  - Algebraic primitive defined by a polynomial primitive

$$f_1(x, y) = x^2 + y^2 - r_1^2$$
$$f_2(x, y) = -\left((x - x_2)^2 + (y - y_2)^2 - r_2^2\right)$$
$$f_3(x, y) = -\left((x - x_3)^2 + (y - y_3)^2 - r_3^2\right)$$
$$f_4(x, y) = -\left(\frac{x^2}{a^2} + \frac{(y - y_4)^2}{b^2} - 1\right)$$

$$f(x, y) = x^2 + y^2 - 4$$



$$O = H_1 \cap H_2 \cap H_3 \cap H_4$$

(Source: LaValle, S. M. Planning Algorithms, 2006)

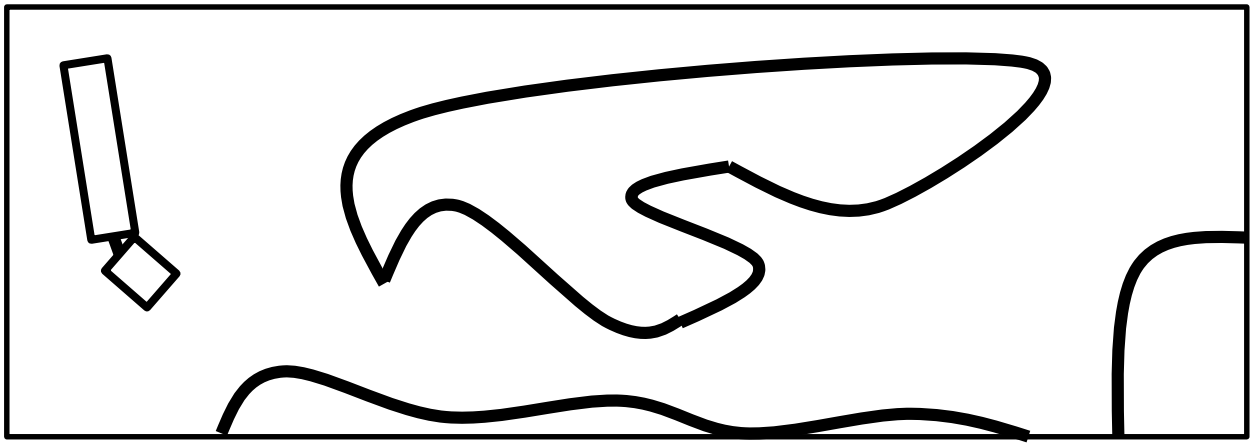
slide by S. K. Gupta

Representing Curves

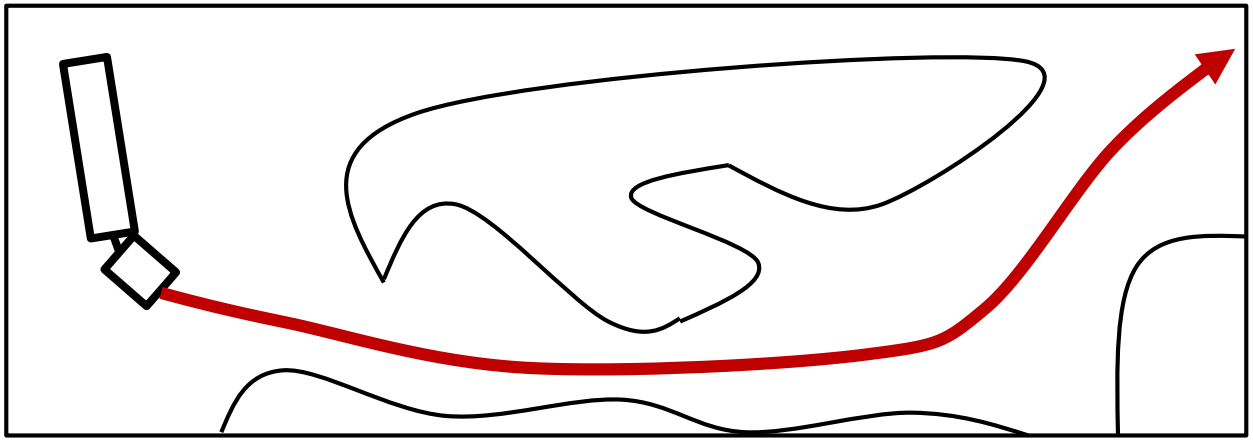
slide by S. K. Gupta

Representing Curves

- For Object Boundaries



- For Motion Primitives (or idealizations of them)



Representing Curves

- Parameterized polynomials
  - Bezier: Single curve parameterized
  - B-Splines: Bezier concatenation maintaining continuity
  - Hermite: Continuity + first derivative continuity
- Single curve segments
  - Single-integrator systems
  - Double-integrator systems
  - etc.
  - Full vehicle dynamics model
- Simple multi-segment idealizations
  - Dubins path
  - Reeds-Shepp path

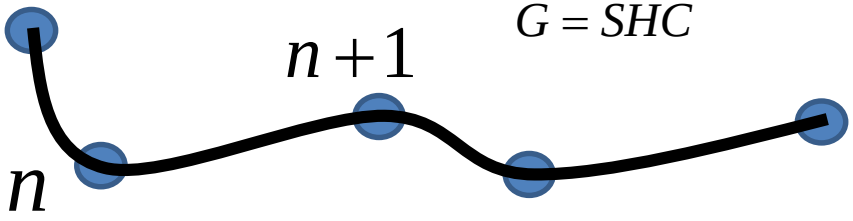
Slide 30% by S. K. Gupta

Cubic Hermite spline

- Multi-segment curve $g(x)$
- Each segment is a 3rd degree polynomial
  - At endpoints of the n-th segment we choose:
    - Locations: p_n and p_{n+1}
    - Tangent vectors: d_n and d_{n+1}
  - Interpolate over $s = 0 \ldots 1$

$$S = \begin{bmatrix} s^3 \\ s^2 \\ s^1 \\ 1 \end{bmatrix} \quad C = \begin{bmatrix} p_n \\ p_{n+1} \\ d_n \\ d_{n+1} \end{bmatrix} \quad H = \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

$$G = SHC$$



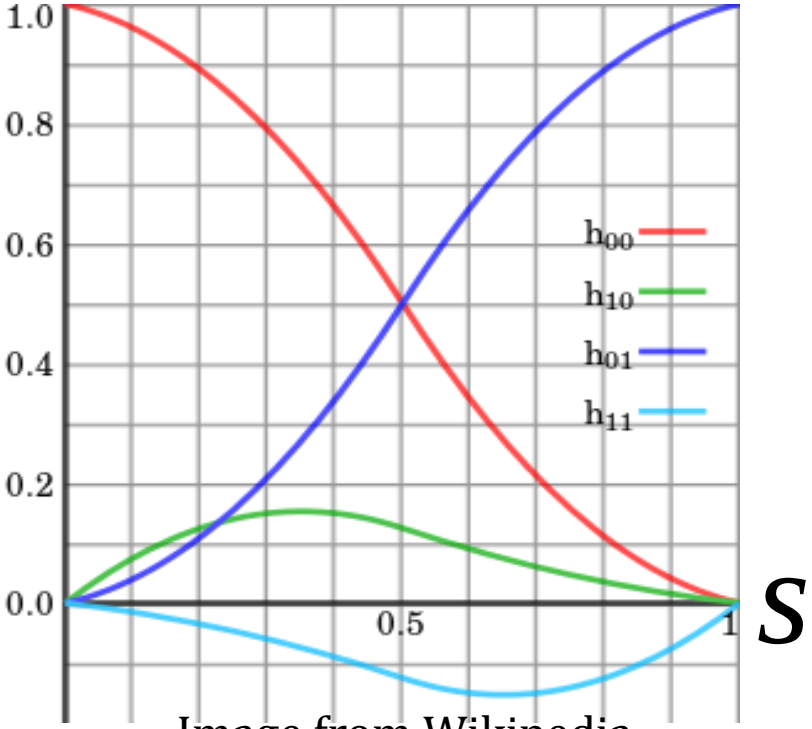


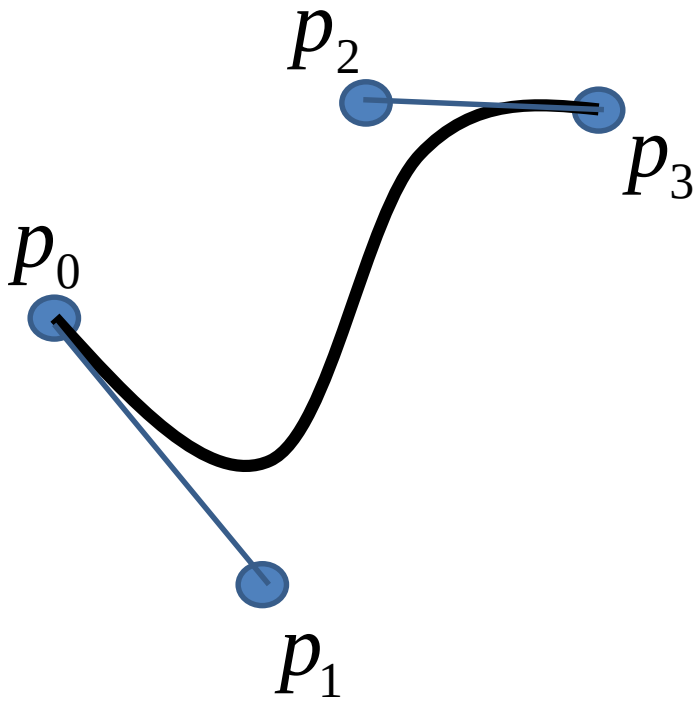
Image from Wikipedia

Bezier Curves

- Single-segment curve $g(x)$
- A 3rd degree polynomial
  - end points: p_0 p_3
  - control points: p_1 p_2
  - "Linear Combination of Basis Polynomials"

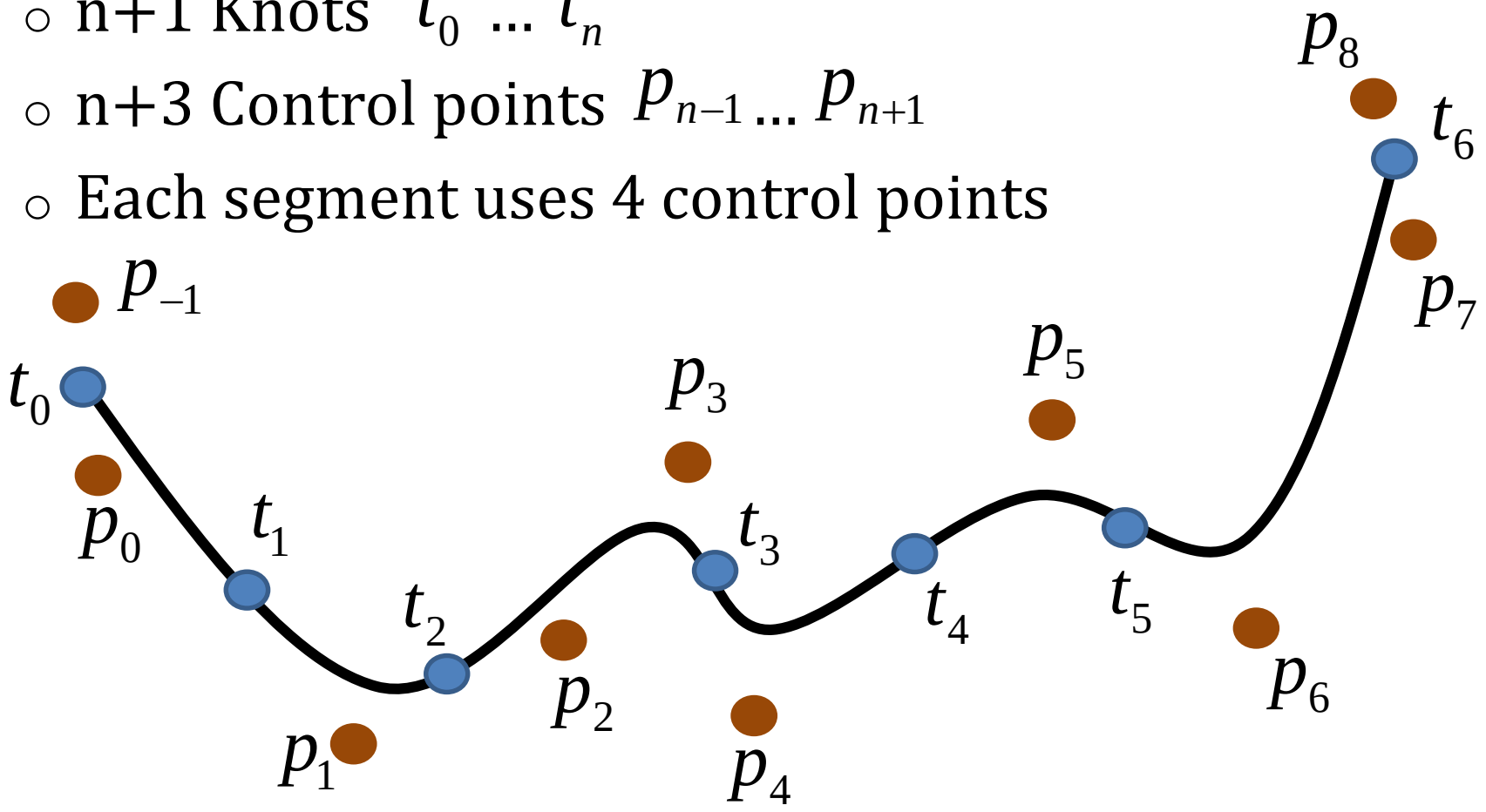
$$S = \begin{bmatrix} s^3 \\ s^2 \\ s^1 \\ 1 \end{bmatrix} \quad C = \begin{bmatrix} p_0 \\ p_1 \\ p_2 \\ p_3 \end{bmatrix} \quad B = \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

$$G = SBC$$



B-Spline (B=Basis)

- Multi-segment curve $g(x)$
- Bezier curve segments (n segments)
  - n+1 Knots $t_0 \ldots t_n$
  - n+3 Control points $p_{n-1} \ldots p_{n+1}$
  - Each segment uses 4 control points



NURBS

- Non-uniform rational B-splines
  - Add weights to B-splines
  - Weight of 1 gives B-spline
  - "Like B-spline but more control points influence each segment"

  - $C(u) = \frac{\sum_{i=0}^{n} \omega_i P_i N_{i,k}(u)}{\sum_{i=0}^{n} \omega_i N_{i,k}(u)}$

    $\omega_i \in \mathbb{R}$ are weights,
    P_i are control points, and
    $N_{i,k}$ are normalized basis functions of degree k

  - $N_{i,k}(u) = \left(\frac{u - t_i}{t_{i+k} - t_i}\right) N_{i,k-1}(u) + \left(\frac{t_{i+k+1} - u}{t_{i+k+1} - t_{i+1}}\right) N_{i+1,k-1}(u)$

$$N_{i,0}(u) = \begin{cases} 1 \text{ if } t_i \le u < t_{i+1} \\ 0 \quad \text{otherwise} \end{cases}$$

where t_i belongs to a knot vector $\{t_o, t_1, \ldots, t_m\}$

Slide 90% by S. K. Gupta

Dubins Curves

- "car-like" but can only move forward so… "airplane-like"
- Velocity (constant) v
- Position x, y
- Heading θ
- Control (bounded) u

- Vehicle Model:

$$\dot{x} = v\cos(\theta)$$
$$\dot{y} = v\sin(\theta)$$
$$\dot{\theta} = u$$

- 6 types of turns
  - RSL = Right-Straight-Left
  - RSR, RSL, LSR, LSL, RLR, LRL
  - Radius is constant function of $\dot{\theta} = u$

RSL Turn

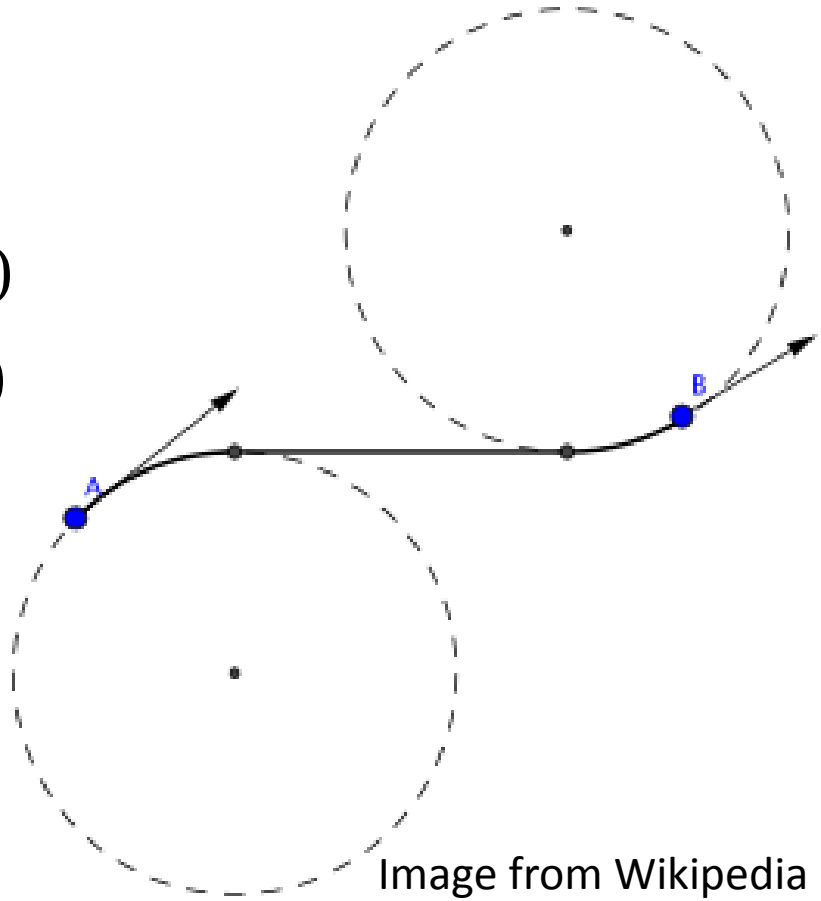
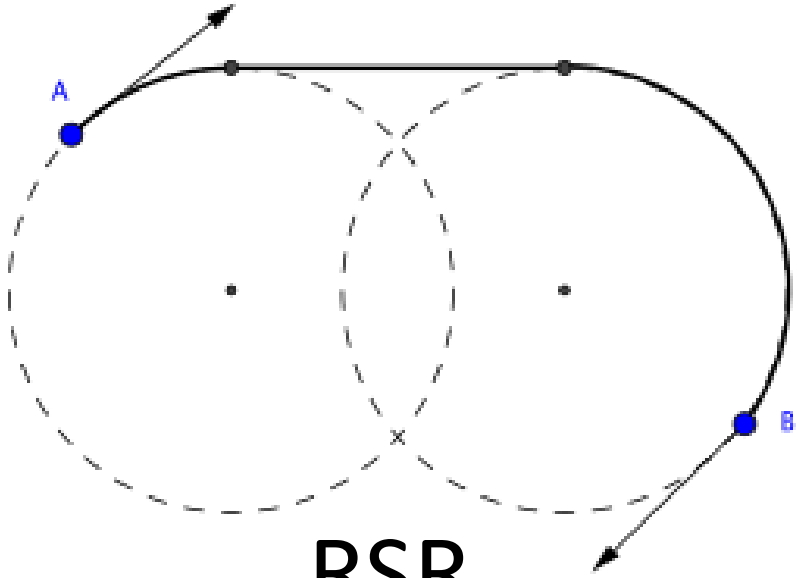
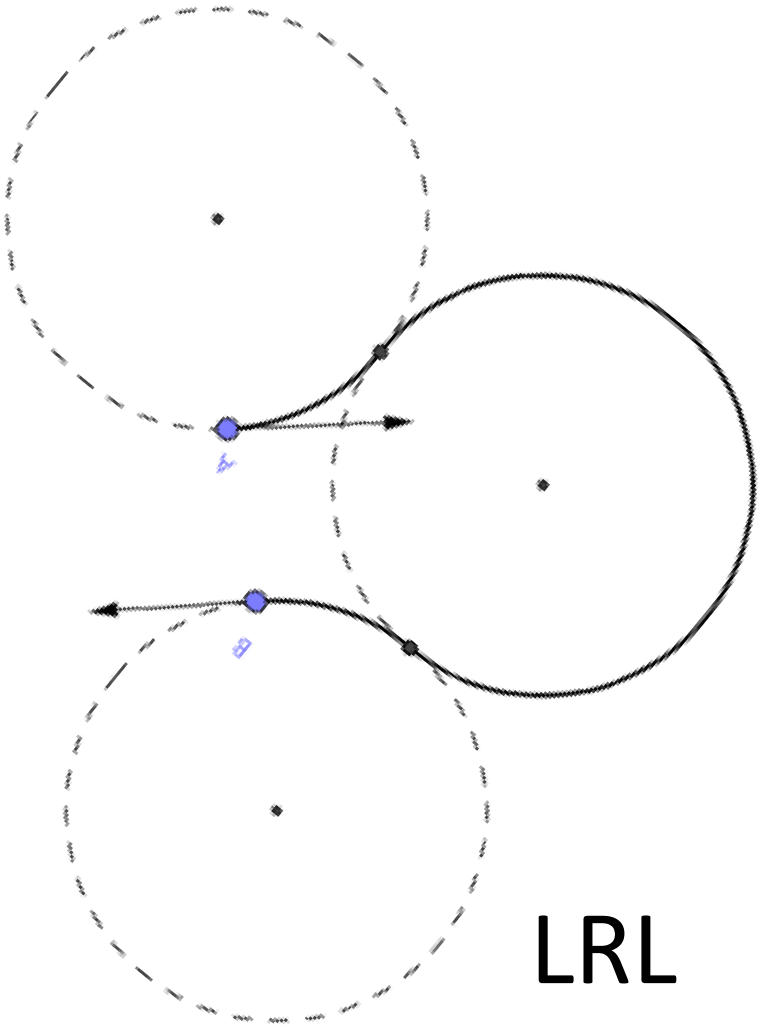


Image from Wikipedia

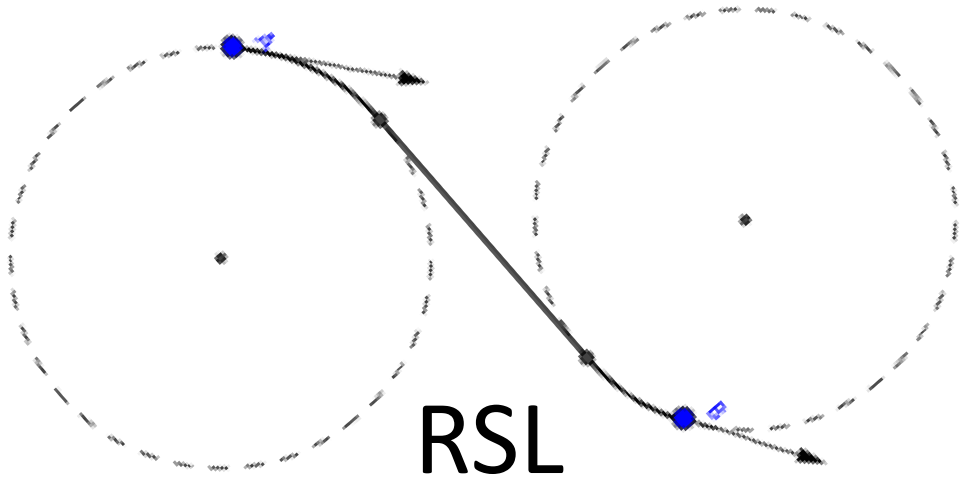
Dubins Curves



RSR



LRL



RSL

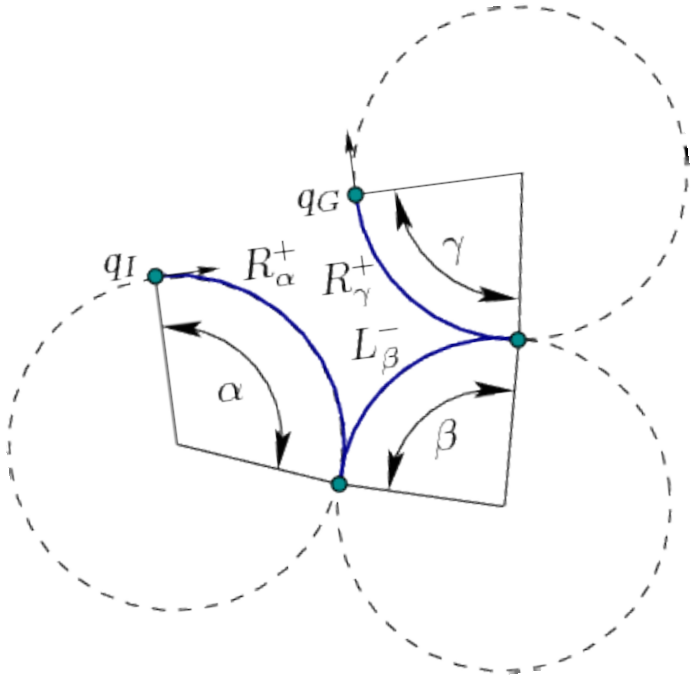
Images from Wikipedia

Reeds-Shepp Curves

- "car-like" but can only move forward and backward
- Velocity (constant) $\{v, -v\}$
- Position x, y
- Heading θ
- Control (bounded) u

- Vehicle Model:

$$\dot{x} = v\cos(\theta)$$
$$\dot{y} = v\sin(\theta)$$
$$\dot{\theta} = u$$



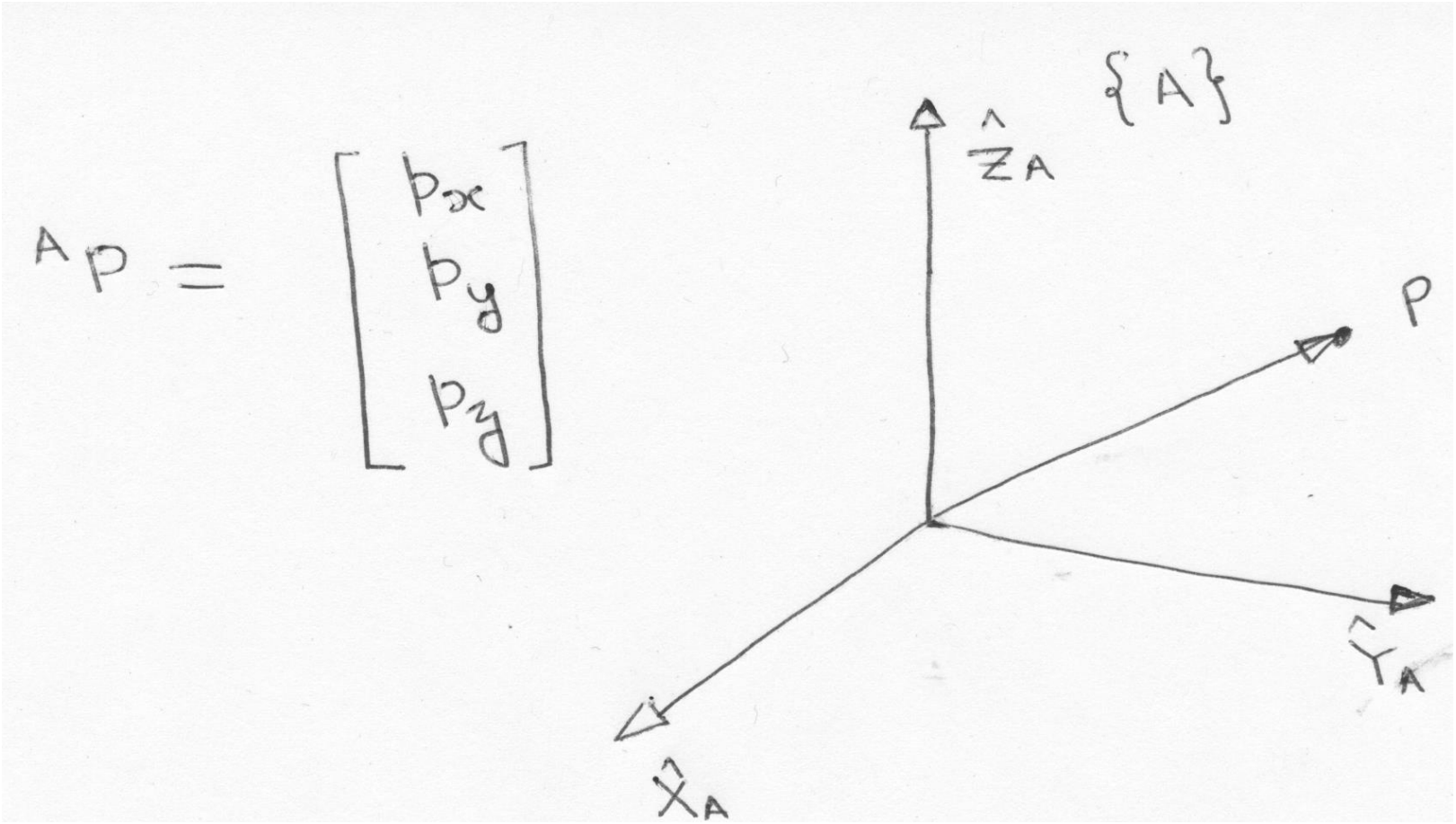
- 48 types of turns
- 46 types of turns actually needed

Image from LaValle's website (planning algorithms)

Robot-Body Transformations

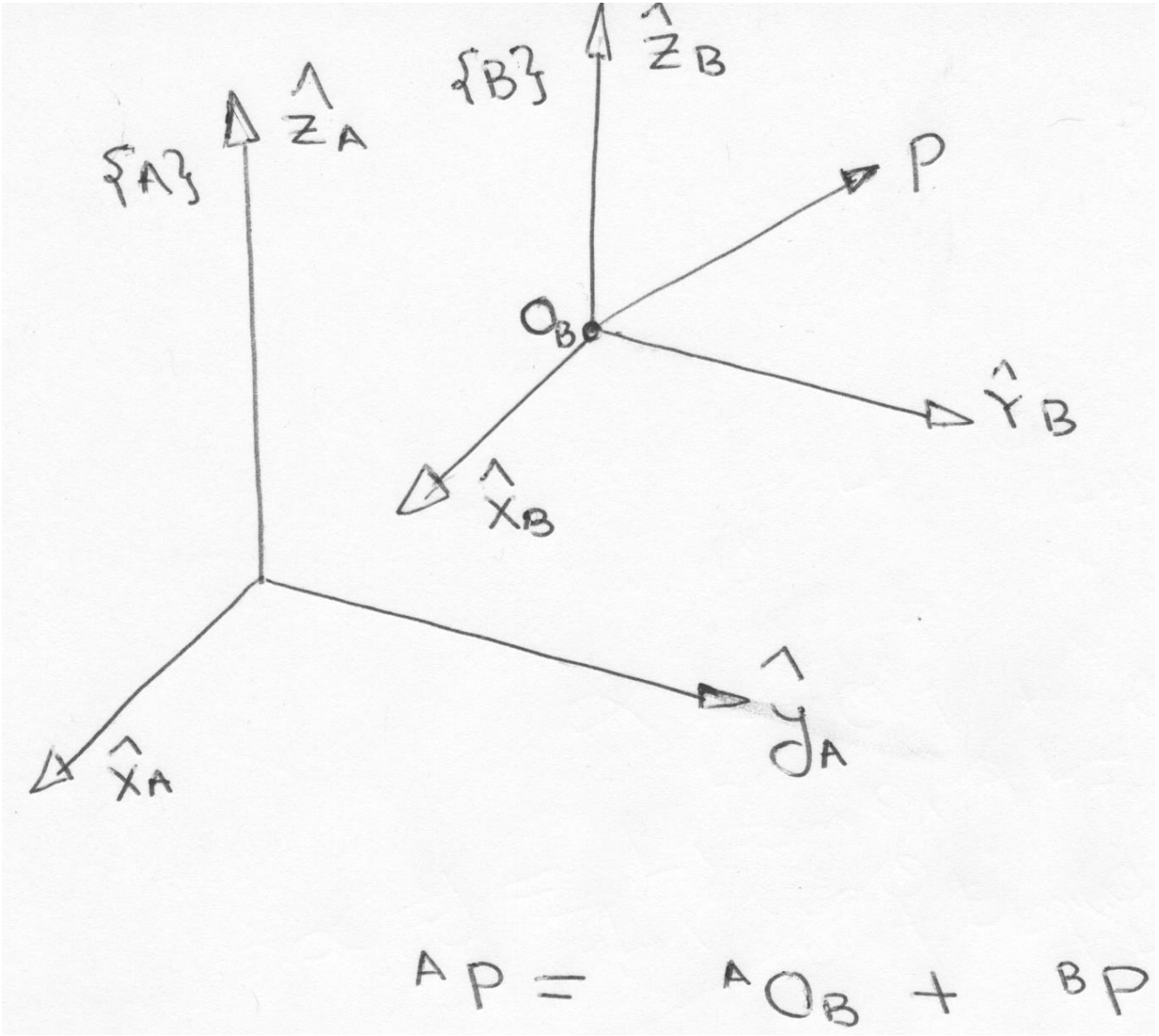
slide by S. K. Gupta

Representing Position Vectors



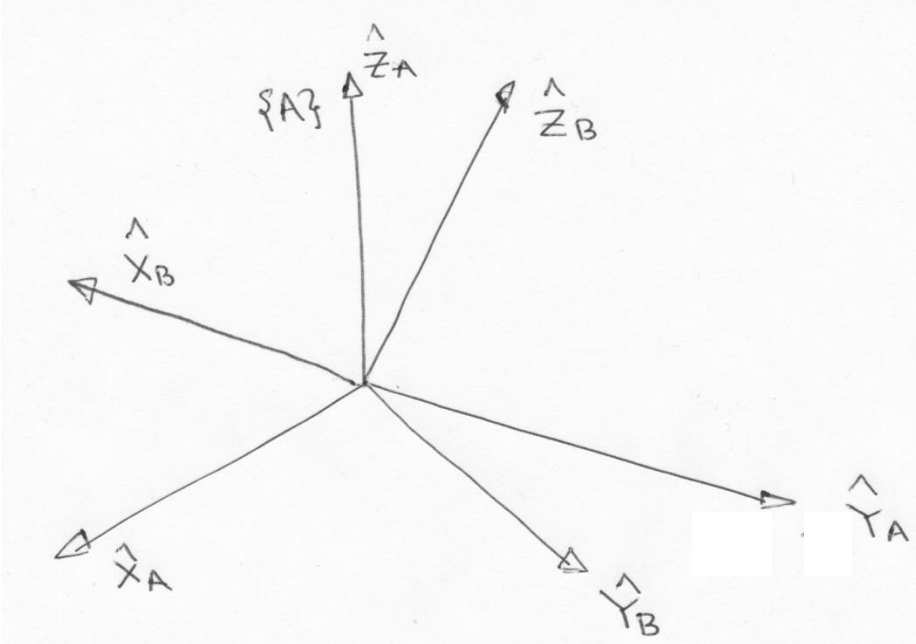
slide by S. K. Gupta

Translated Frame



slide by S. K. Gupta

Rotated Frame



$$ {}^{A}_{B}R = \begin{bmatrix} {}^{A}\hat{X}_B & {}^{A}\hat{Y}_B & {}^{A}\hat{Z}_B \end{bmatrix} $$

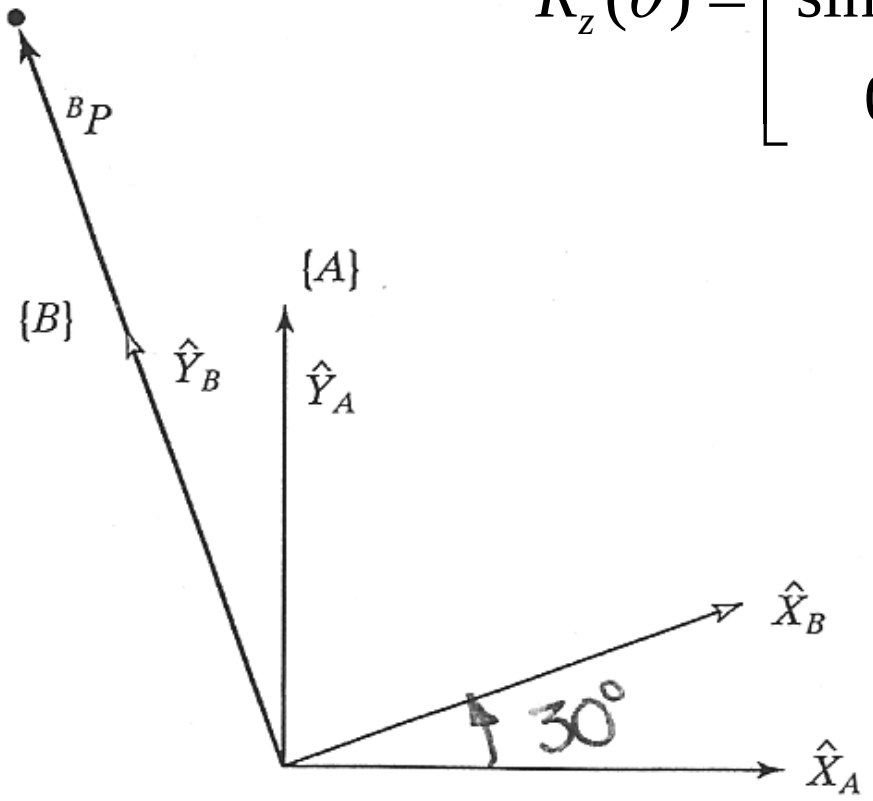
$$ = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} $$

slide by S. K. Gupta

Rotation Matrix Example

matrix for pure rotation around z axis

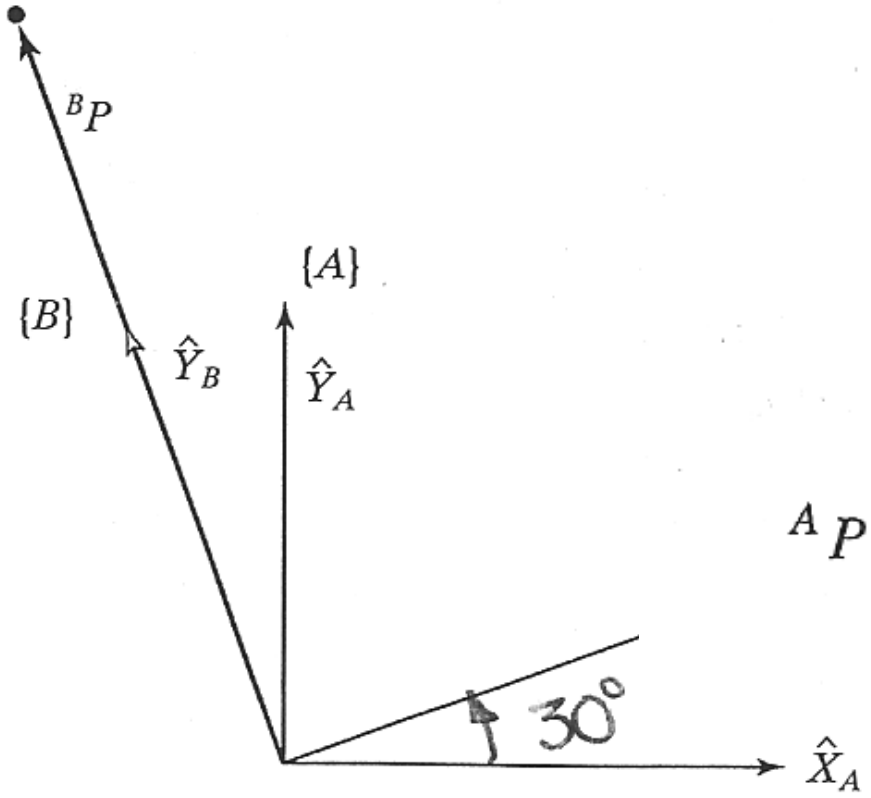
$$R_z(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



slide 50% by S. K. Gupta

Example (Cont.)

$$ {}^{A}_{B}R = \begin{bmatrix} 0.866 & -0.500 & 0.000 \\ 0.500 & 0.866 & 0.000 \\ 0.000 & 0.000 & 1.000 \end{bmatrix}. $$

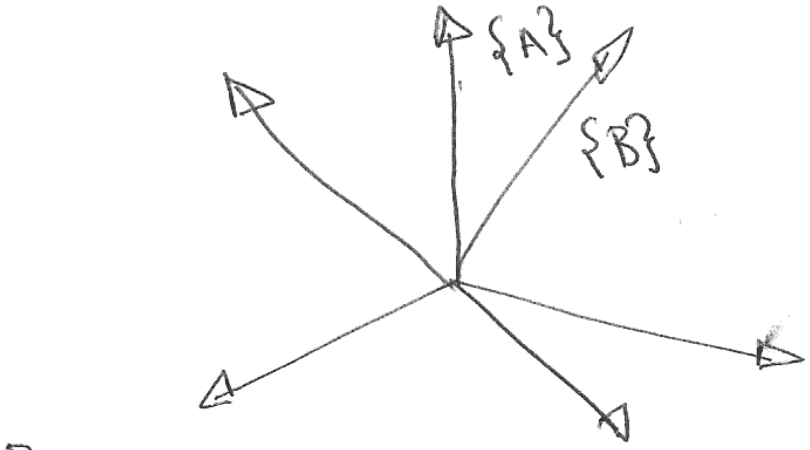


$$ {}^{B}P = \begin{bmatrix} 0.0 \\ 2.0 \\ 0.0 \end{bmatrix}, $$

$$ {}^{A}P = {}^{A}_{B}R\,{}^{B}P = \begin{bmatrix} -1.000 \\ 1.732 \\ 0.000 \end{bmatrix}. $$

Slide by S. K. Gupta

Inverting a Rotation Matrix

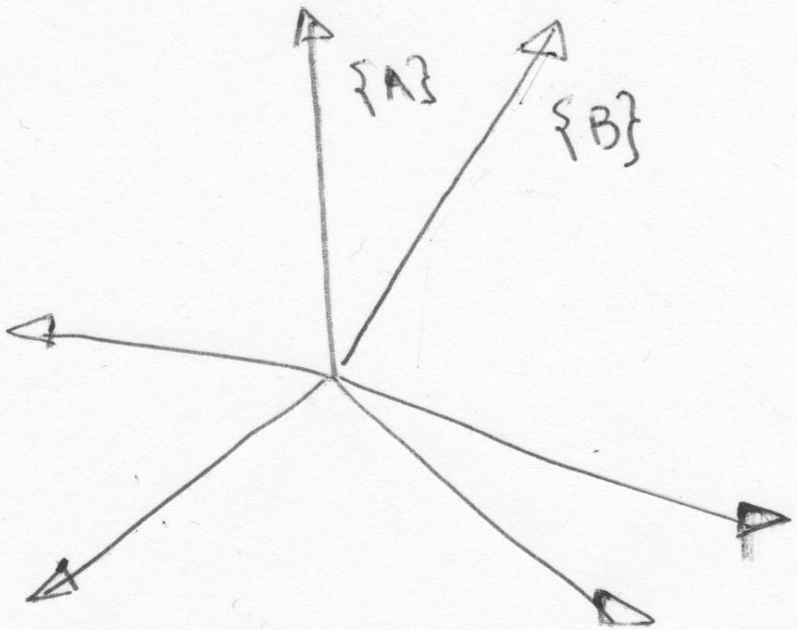


$$ {}^{B}_{A}R = {}^{A}_{B}R^{T} = {}^{A}_{B}R^{-1} $$

slide by S. K. Gupta

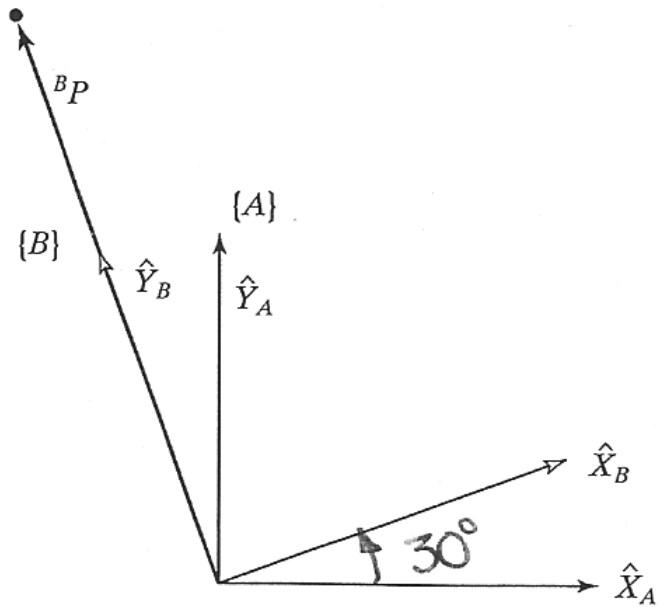
Effect of Frame Rotation

$${}^{A}p = {}^{A}_{B}R \; {}^{B}p$$



slide by S. K. Gupta

Example



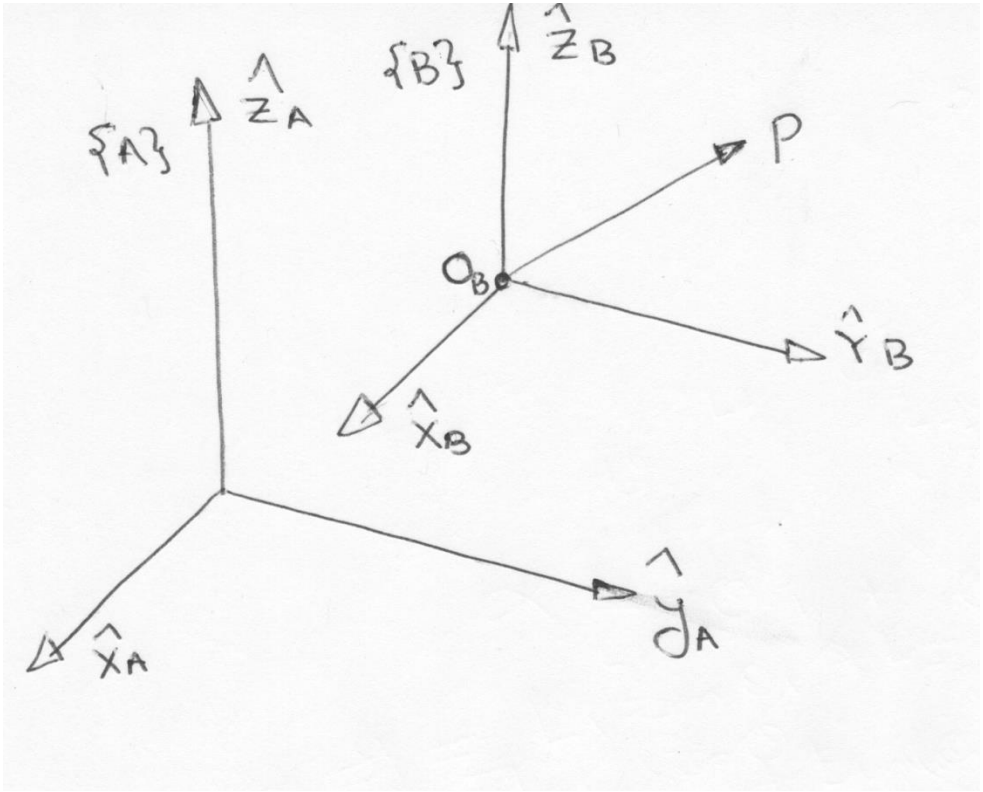
$$ {}^{A}_{B}R = \begin{bmatrix} 0.866 & -0.500 & 0.000 \\ 0.500 & 0.866 & 0.000 \\ 0.000 & 0.000 & 1.000 \end{bmatrix}. $$

$$ {}^{B}P = \begin{bmatrix} 0.0 \\ 2.0 \\ 0.0 \end{bmatrix}, $$

$$ {}^{A}P = {}^{A}_{B}R\,{}^{B}P = \begin{bmatrix} -1.000 \\ 1.732 \\ 0.000 \end{bmatrix}. $$

slide by S. K. Gupta

Expressing a Frame in Terms of Other Frame



$$\{B\} = \left\{ {}^{A}_{B}R, \quad {}^{A}O_B \right\}$$

slide by S. K. Gupta

Comparing Translations and Rotations

$$^{A}p = {}^{A}O_{B} + {}^{B}p$$

$$^{A}p = {}^{A}_{B}R\ {}^{B}p$$

We would like,

$$^{A}p = {}^{A}_{B}T\ {}^{B}p$$

slide by S. K. Gupta

Homogenous Transformations (Translation)

Take home message:
By choosing to use a 4x4 matrix for representation (requires inserting an extra "1" row in point vectors) **we are able to use same form/operation for rotation and translation**

$$ {}^{A}P = \begin{bmatrix} {}^{A}x_P \\ {}^{A}y_P \\ {}^{A}z_P \\ 1 \end{bmatrix} $$

$$ {}^{B}P = \begin{bmatrix} {}^{B}x_P \\ {}^{B}y_P \\ {}^{B}z_P \\ 1 \end{bmatrix} $$

$$ {}^{A}x_P = {}^{B}x_P + {}^{A}x_{OB} $$

$$ {}^{A}y_P = {}^{B}y_P + {}^{A}y_{OB} $$

$$ {}^{A}z_P = {}^{B}z_P + {}^{A}z_{OB} $$

$$ {}^{A}P = \begin{bmatrix} 1 & 0 & 0 & {}^{A}x_{OB} \\ 0 & 1 & 0 & {}^{A}y_{OB} \\ 0 & 0 & 1 & {}^{A}z_{OB} \\ 0 & 0 & 0 & 1 \end{bmatrix} {}^{B}P $$

$$ {}^{A}P = {}^{A}_{B}T \, {}^{B}P $$

Slide 70% by S. K. Gupta

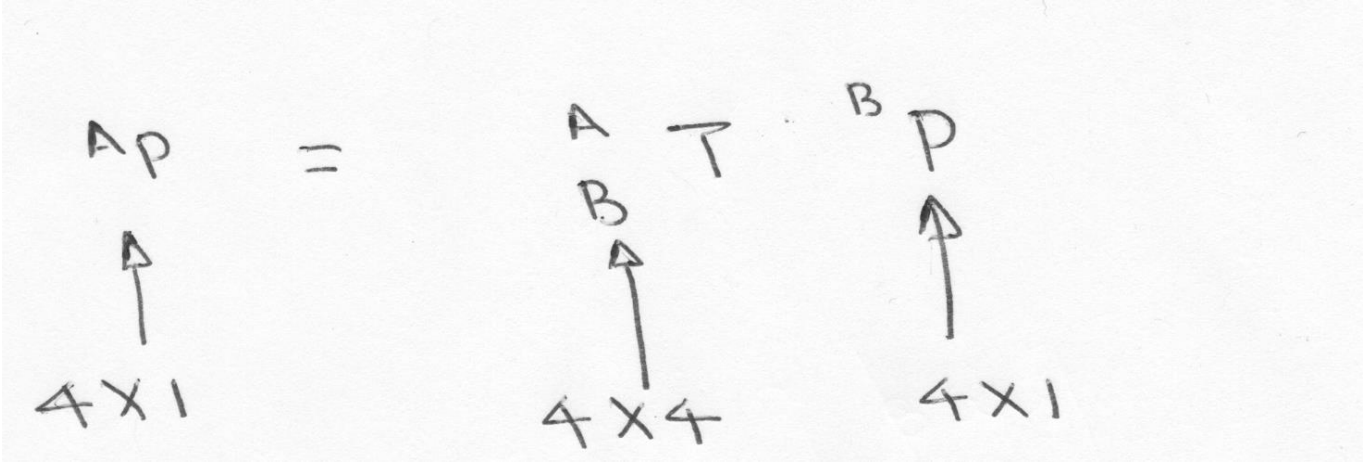
Homogenous Transformation (rotation)

$$ {}^{A}p = \begin{bmatrix} {}^{A}_{B}R & \begin{matrix} 0 \\ 0 \\ 0 \end{matrix} \\ \hline \begin{matrix} 0 & 0 & 0 \end{matrix} & 1 \end{bmatrix} {}^{B}p $$

slide by S. K. Gupta

Homogeneous Coordinates and Transforms

- Both translation and rotation are expressed using the same framework

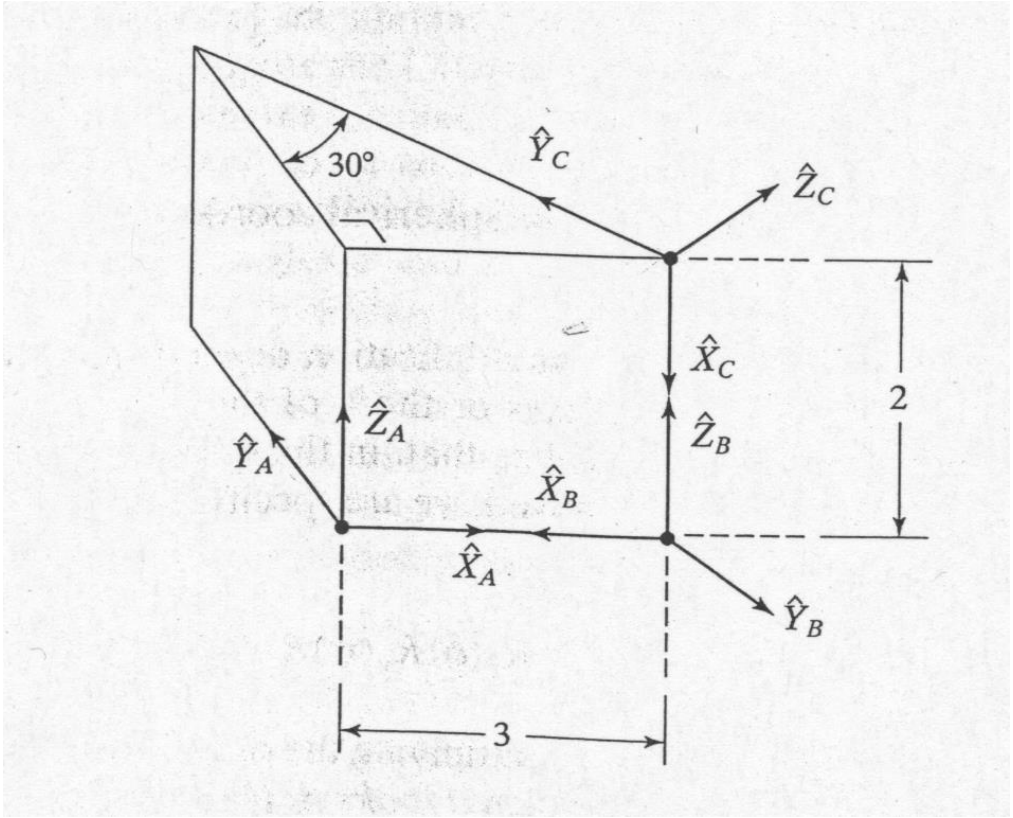


This makes them easy to chain together…

$$^{A}p = {}^{A}_{B}T \cdot {}^{B}_{C}T \cdot {}^{C}_{D}T \cdot {}^{D}p = {}^{A}_{D}T \cdot {}^{D}p$$

slide 70% by S. K. Gupta

Example



A

B

T = ?

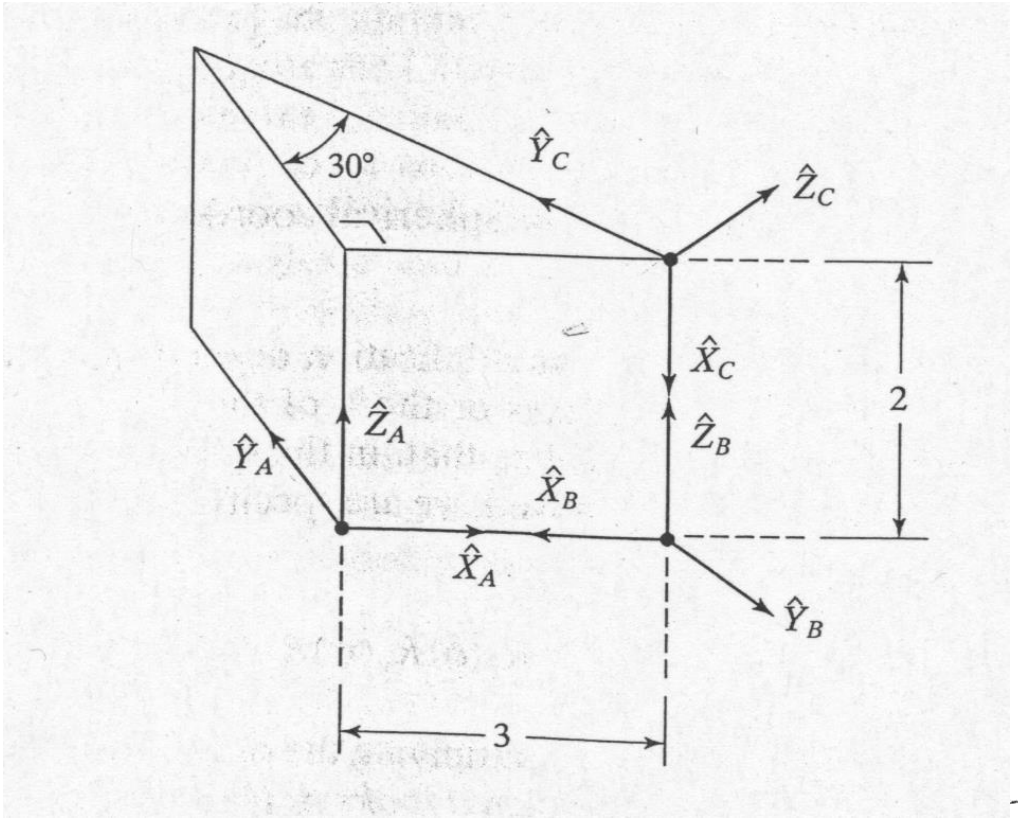
slide by S. K. Gupta

Summary of how to changes two coordinate frames

- Consider two fixed coordinate frames {A} and {B}
- How to transform points ${}^{B}P$ modeled in frame {B} into frame {A}
  - Determine position vector of the origin of {B} in terms of {A}
  - Determine rotation matrix associated with frame {B}
  - Determine homogeneous transformation
    - Note: you can always figure out the homogeneous pure rotation matrix R and the homogenous pure translation matrix Q and then multiply them to get the overall homogeneous transform.
    - Order of operations: rotate then translate

$$Tp = QRp$$

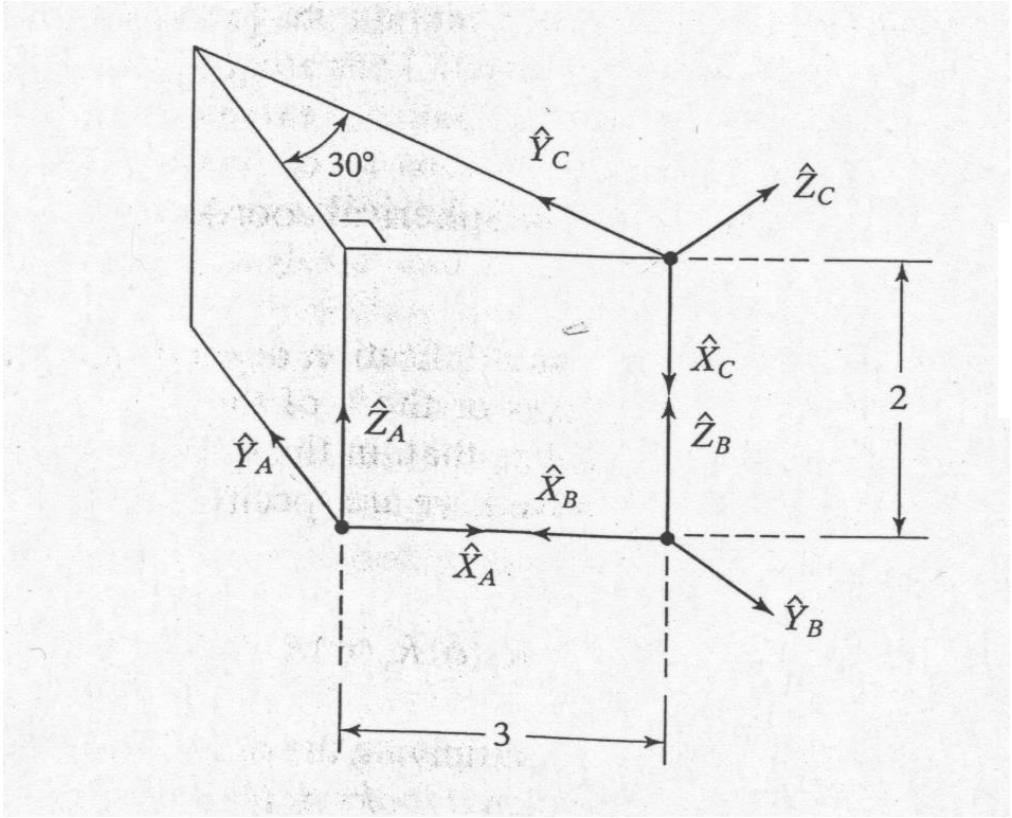
slide 70% by S. K. Gupta



$$ {}^{A}_{B}T = \begin{bmatrix} -1 & 0 & 0 & 3 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

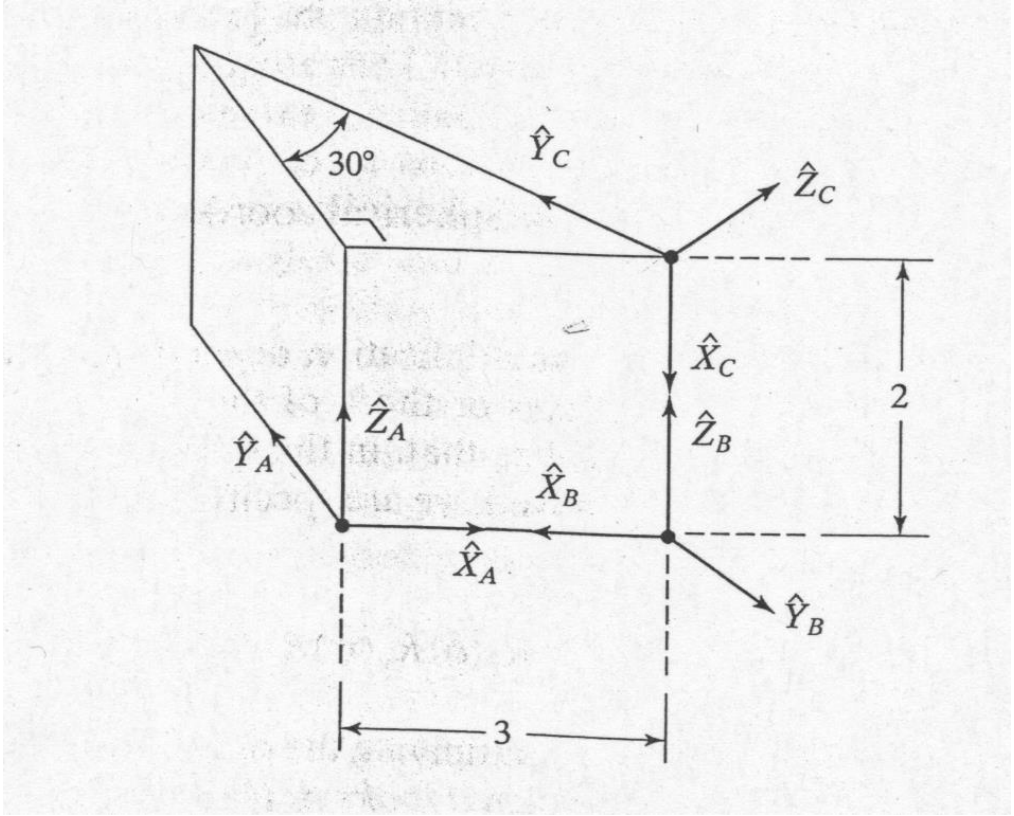
slide by S. K. Gupta

Another Example



A
C $T = ?$

slide by S. K. Gupta



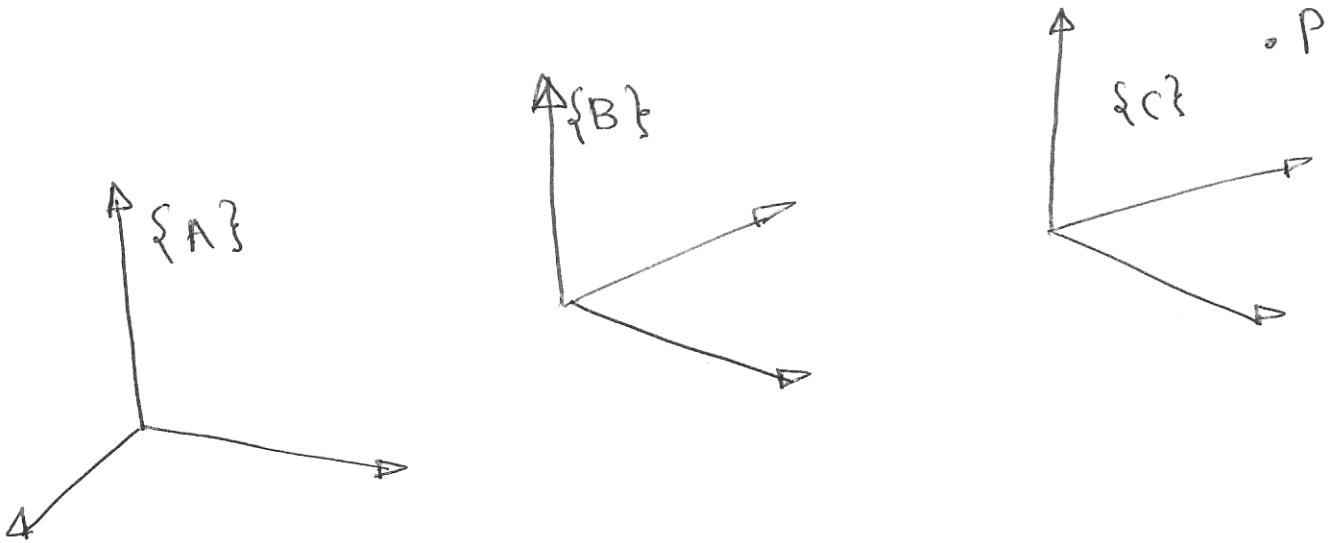
$$ {}^{A}_{C}T = \begin{bmatrix} 0 & -0.5 & 0.866 & 3 \\ 0 & 0.866 & 0.5 & 0 \\ -1 & 0 & 0 & 2 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

slide by S. K. Gupta

Inverting a Homogeneous Transformation

$$ {}^{B}_{A}T = {}^{A}_{B}T^{-1} $$

Composition of Transformations



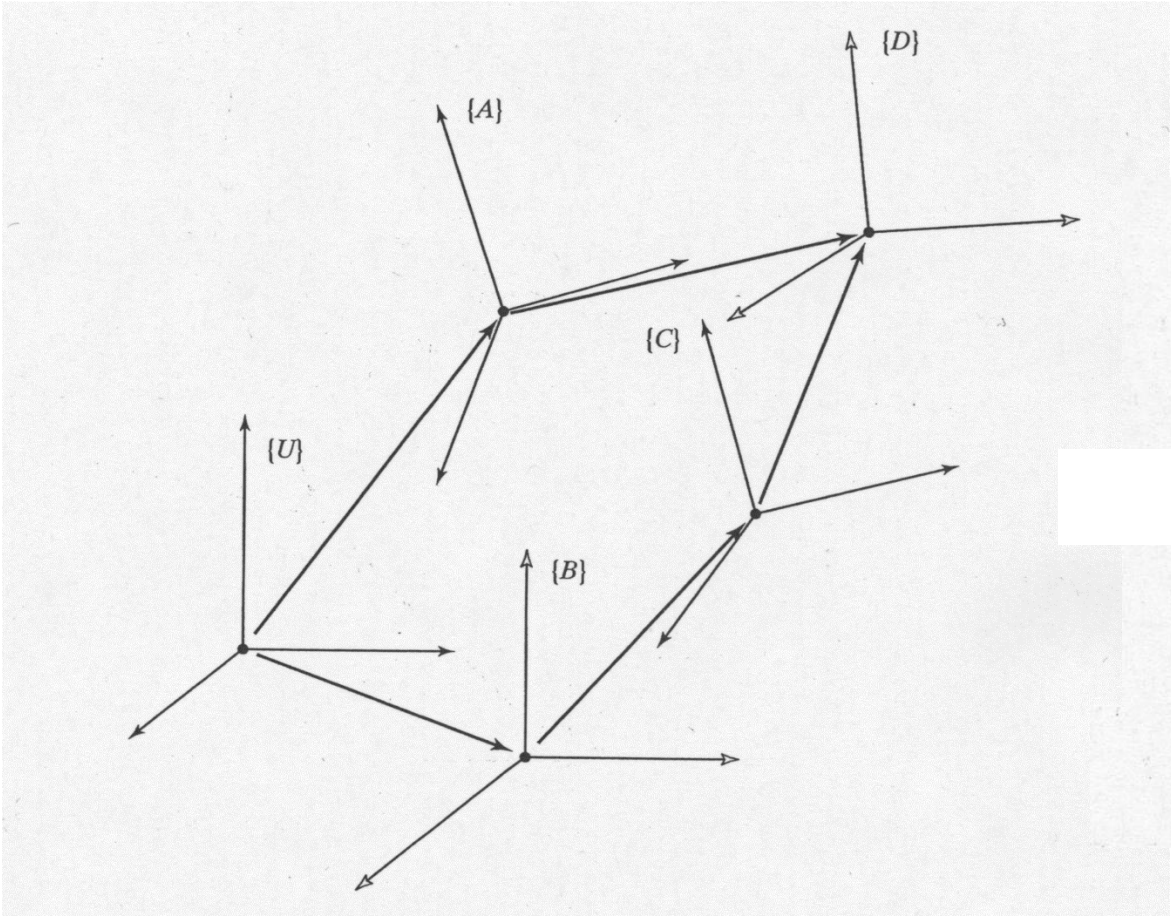
$$ {}^{B}p = {}^{B}_{C}T \; {}^{C}p $$

$$ {}^{A}p = {}^{A}_{B}T \; {}^{B}p $$

$$ = {}^{A}_{B}T \; {}^{B}_{C}T \; {}^{C}p $$

slide by S. K. Gupta

Transformation Equations



$$ {}^{U}_{D}T = {}^{U}_{A}T\,{}^{A}_{D}T; $$

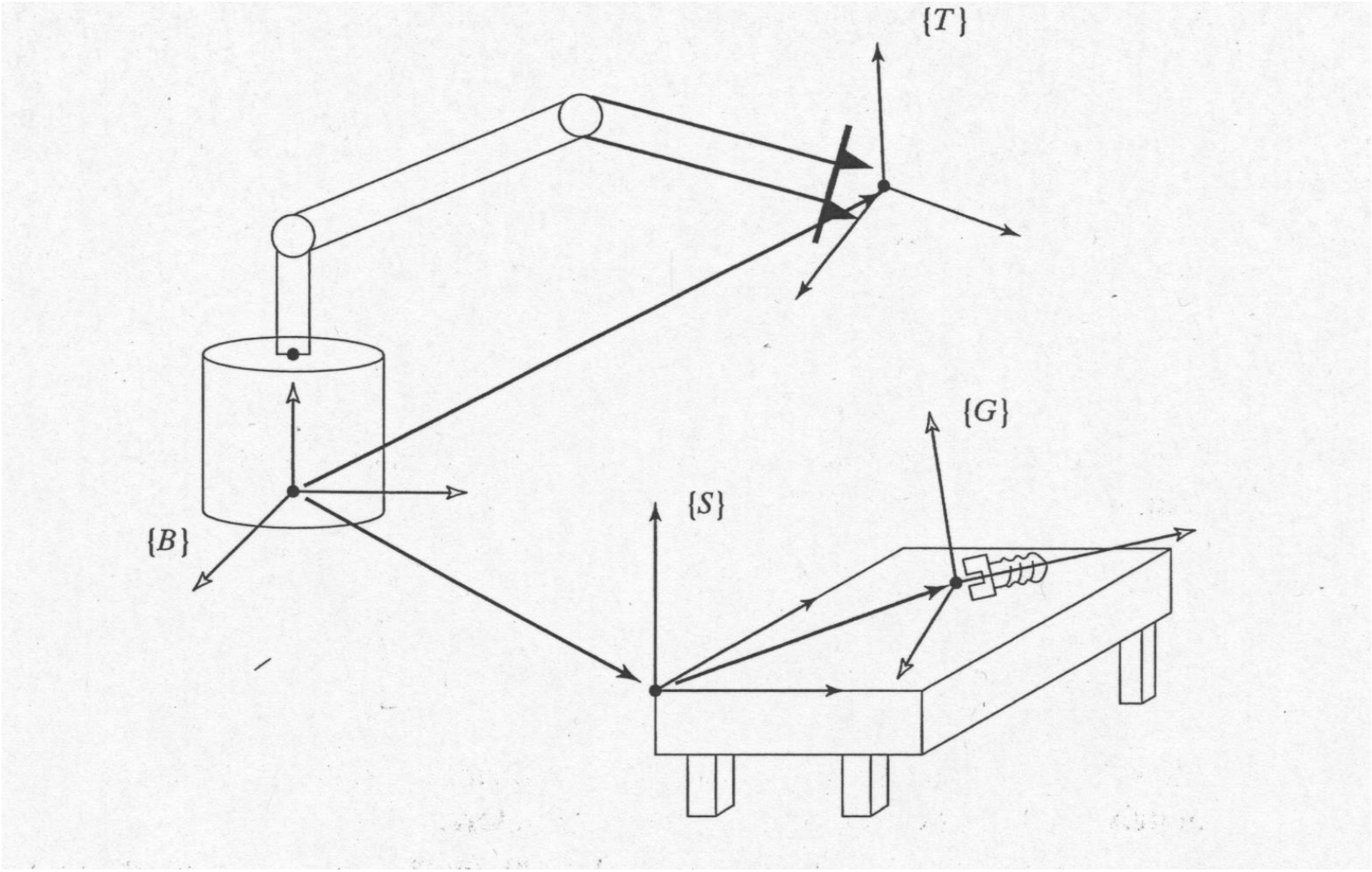
$$ {}^{U}_{D}T = {}^{U}_{B}T\,{}^{B}_{C}T\,{}^{C}_{D}T. $$

$$ {}^{U}_{A}T\,{}^{A}_{D}T = {}^{U}_{B}T\,{}^{B}_{C}T\,{}^{C}_{D}T. $$

$$ {}^{B}_{C}T = {}^{U}_{B}T^{-1}\,{}^{U}_{A}T\,{}^{A}_{D}T\,{}^{C}_{D}T^{-1}. $$

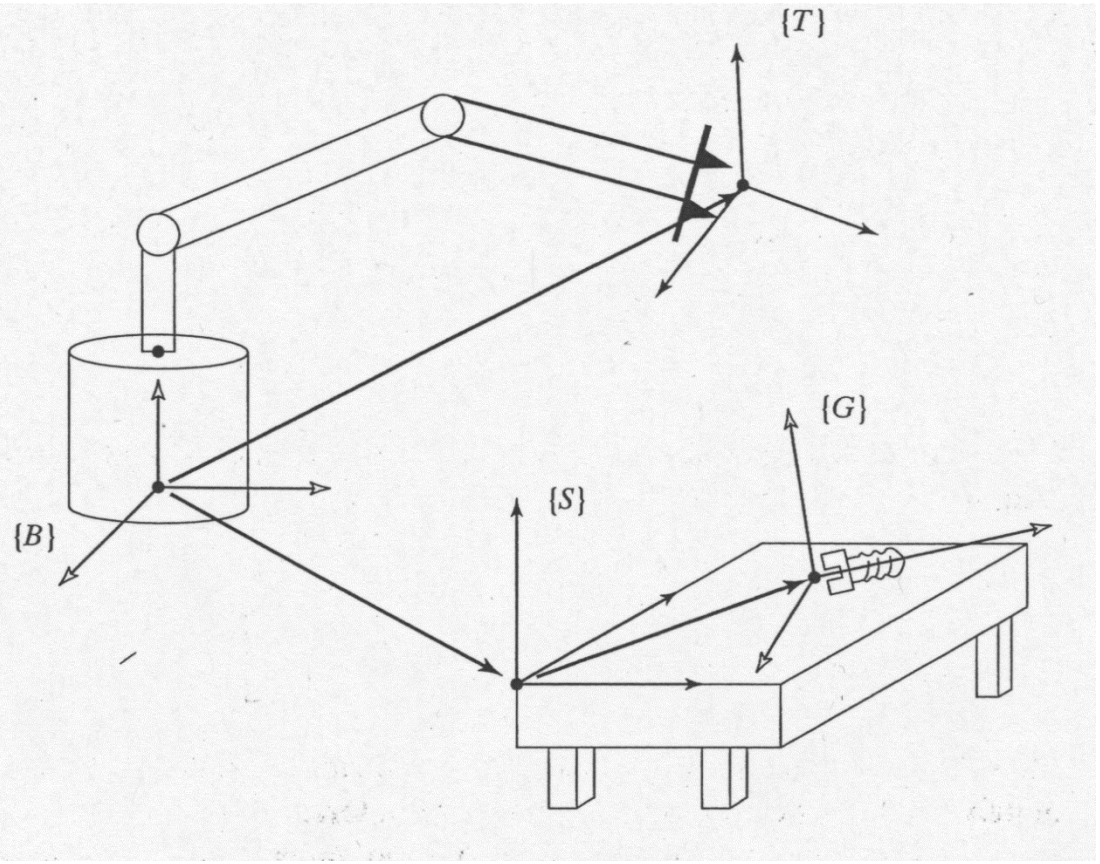
slide by S. K. Gupta

An Example



slide by S. K. Gupta

Example



Known

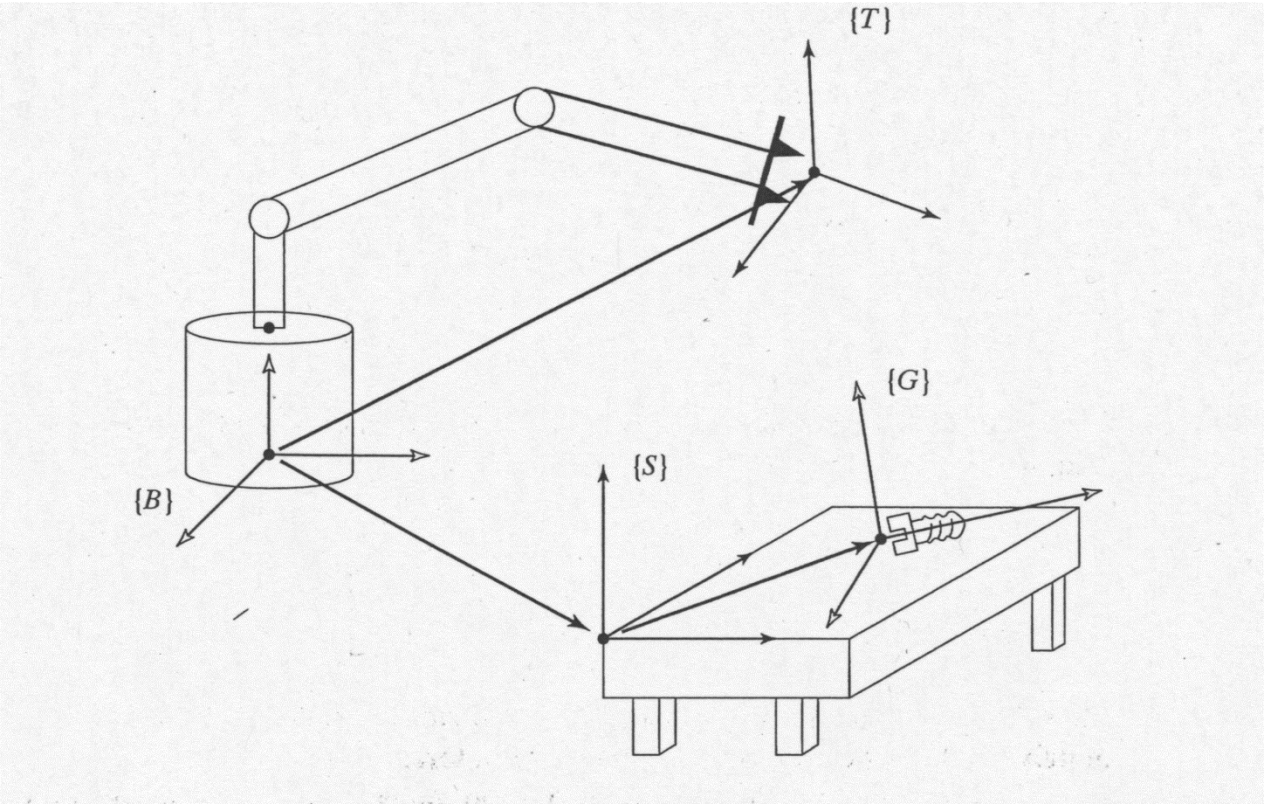
$${}^{B}_{T}T \quad , \quad {}^{B}_{S}T \quad , \quad {}^{S}_{G}T$$

Need to find

$${}^{T}_{G}T = ?$$

slide by S. K. Gupta

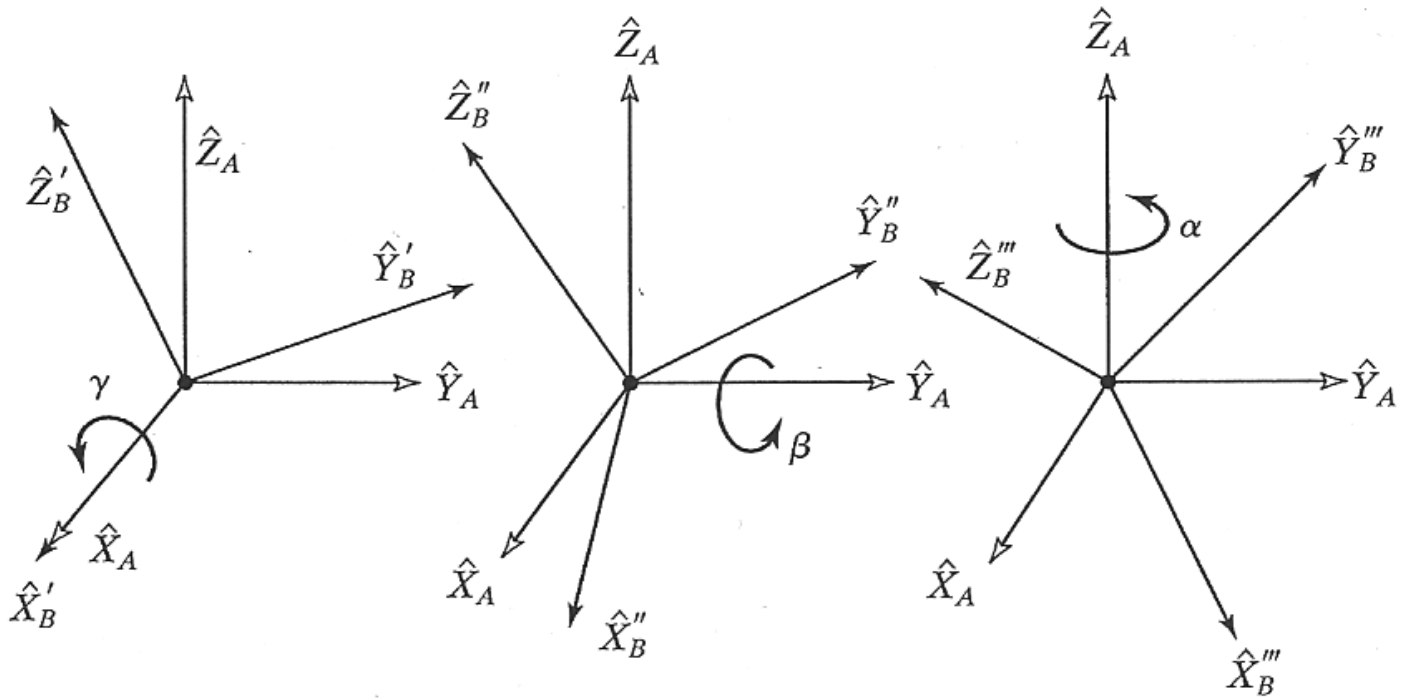
Answer



$$ {}^{T}_{G}T = {}^{B}_{T}T^{-1}\,{}^{B}_{S}T\,{}^{S}_{G}T. $$

slide by S. K. Gupta

X-Y-Z Fixed Angle Rotation



slide by S. K. Gupta

X-Y-Z Fixed Angle Rotation (Cont.)

$$
\begin{aligned}
{}^{A}_{B}R_{XYZ}(\gamma,\beta,\alpha) &= R_Z(\alpha)R_Y(\beta)R_X(\gamma) \\
&= \begin{bmatrix} c\alpha & -s\alpha & 0 \\ s\alpha & c\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}
\begin{bmatrix} c\beta & 0 & s\beta \\ 0 & 1 & 0 \\ -s\beta & 0 & c\beta \end{bmatrix}
\begin{bmatrix} 1 & 0 & 0 \\ 0 & c\gamma & -s\gamma \\ 0 & s\gamma & c\gamma \end{bmatrix},
\end{aligned}
$$

where $c\alpha$ is shorthand for $\cos\alpha$, $s\alpha$ for $\sin\alpha$, and so on.

$$
{}^{A}_{B}R_{XYZ}(\gamma,\beta,\alpha) = \begin{bmatrix} c\alpha c\beta & c\alpha s\beta s\gamma - s\alpha c\gamma & c\alpha s\beta c\gamma + s\alpha s\gamma \\ s\alpha c\beta & s\alpha s\beta s\gamma + c\alpha c\gamma & s\alpha s\beta c\gamma - c\alpha s\gamma \\ -s\beta & c\beta s\gamma & c\beta c\gamma \end{bmatrix}.
$$

slide by S. K. Gupta

Final Matrix

$$ {}^{A}_{B}R_{XYZ}(\gamma, \beta, \alpha) = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}. $$

slide by S. K. Gupta

Recovering Angles from Final Matrix

$$\beta = \text{Atan2}(-r_{31}, \sqrt{r_{11}^2 + r_{21}^2}),$$

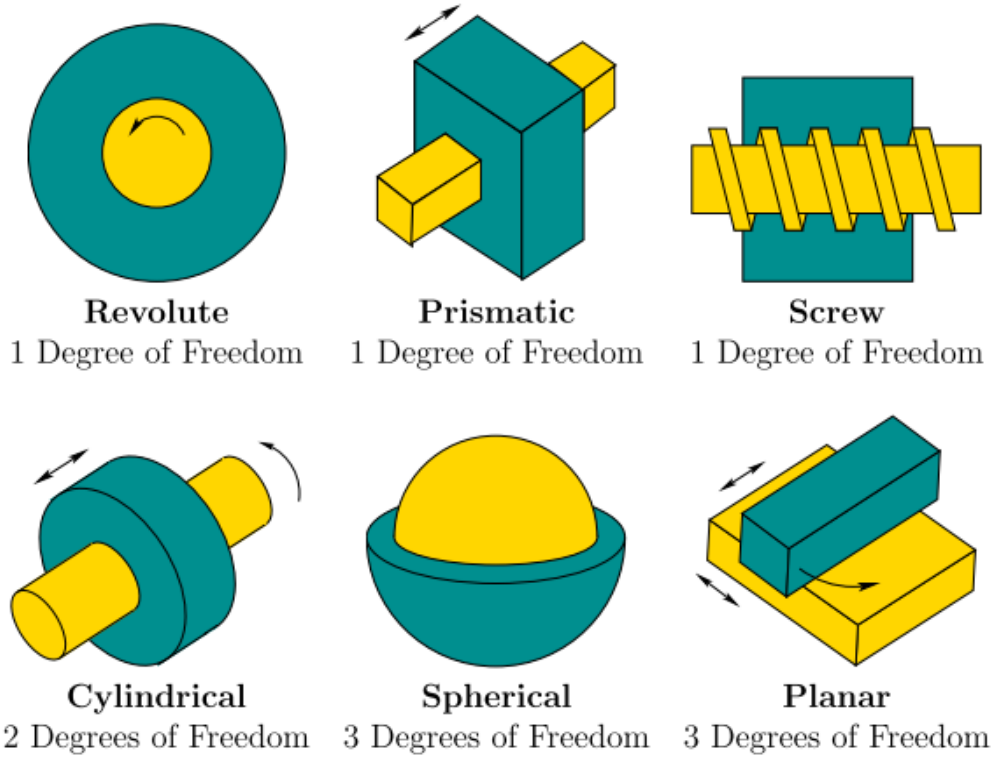
$$\alpha = \text{Atan2}(r_{21}/c\beta, r_{11}/c\beta),$$

$$\gamma = \text{Atan2}(r_{32}/c\beta, r_{33}/c\beta),$$

slide by S. K. Gupta

MARYLAND ROBOTICS CENTER
THE INSTITUTE FOR SYSTEMS RESEARCH

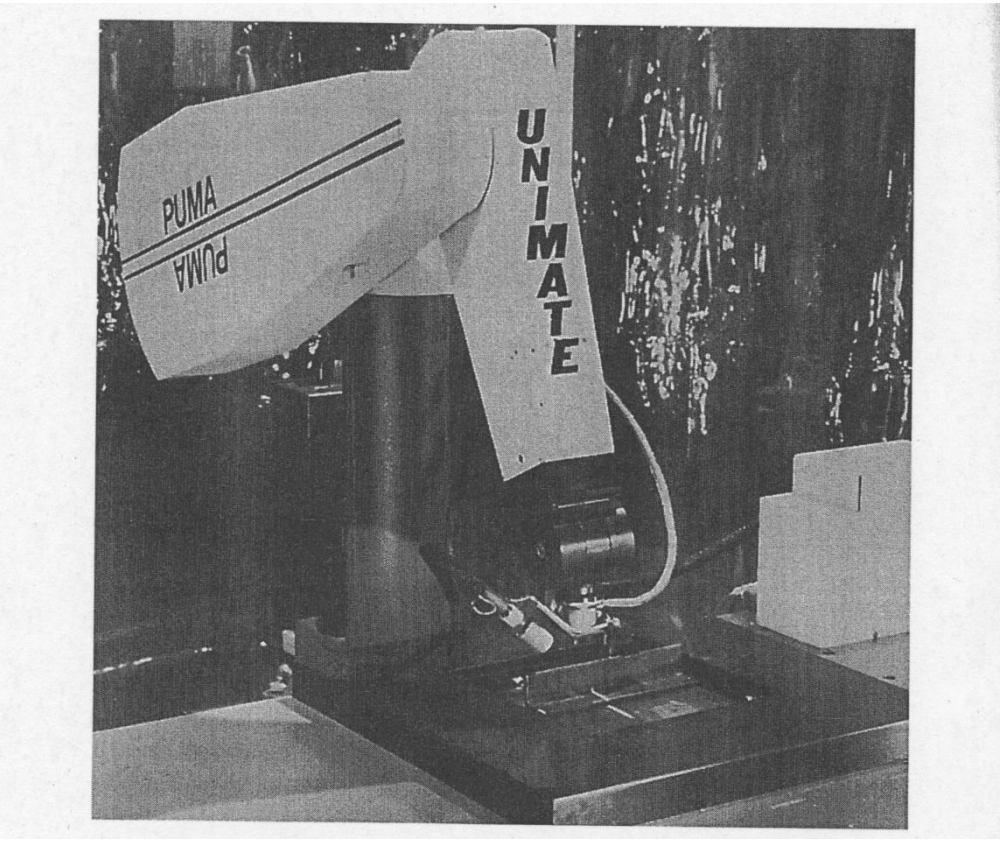
Transformations of 3D Kinematic Chains



(Source: LaValle, S. M. Planning Algorithms, 2006)

slide by S. K. Gupta

Motivation



- **Goal:** Describe end effector frame in terms of base frame
- Location and orientation of end effector frame depends upon the joint values

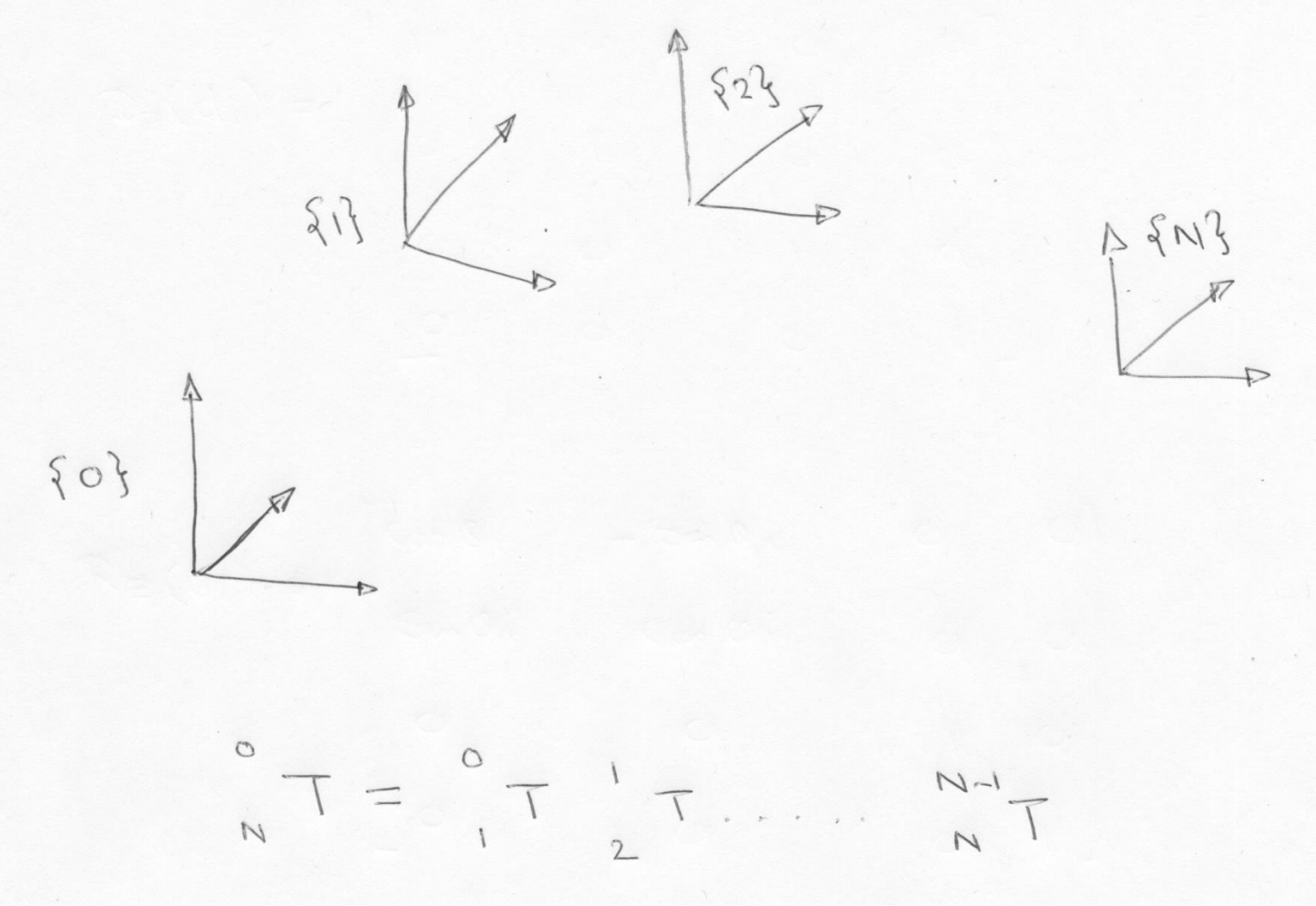
slide by S. K. Gupta

Strategy

- Attach frames to links
- Develop frame to frame transforms
  - In terms of joint parameters and link parameters
- Compute transform to express end effector frame in terms of base frame
  - Use composition rules

slide by S. K. Gupta

Basic Idea



slide by S. K. Gupta

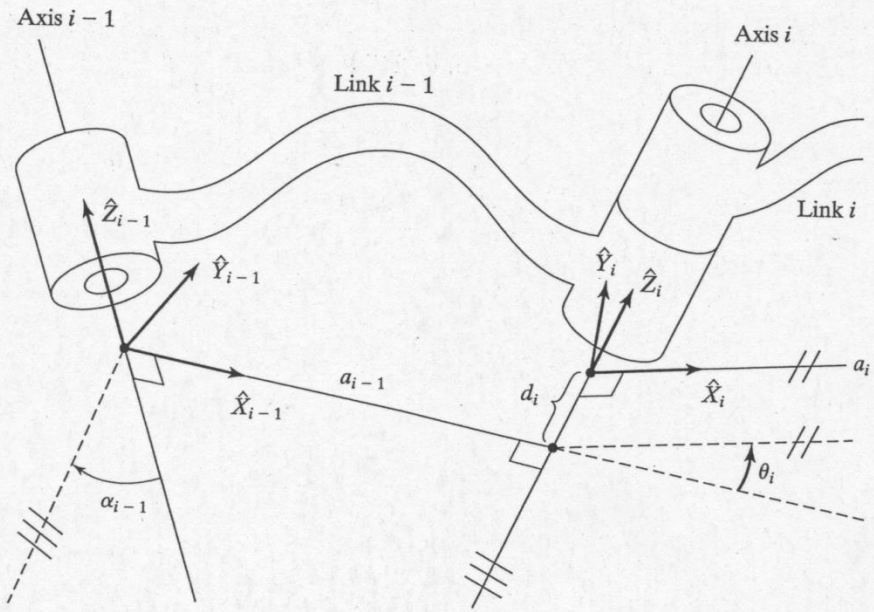
Summary of Denavit–Hartenberg Parameters

a_i = *the distance from* $\hat{Z}_i$ *to* $\hat{Z}_{i+1}$ *measured along* $\hat{X}_i$;

α_i = *the angle from* $\hat{Z}_i$ *to* $\hat{Z}_{i+1}$ *measured about* $\hat{X}_i$;

d_i = *the distance from* $\hat{X}_{i-1}$ *to* $\hat{X}_i$ *measured along* $\hat{Z}_i$; and

θ_i = *the angle from* $\hat{X}_{i-1}$ *to* $\hat{X}_i$ *measured about* $\hat{Z}_i$.



- $a_0 = 0$; $\alpha_0 = 0$
- $a_n = 0$; $\alpha_n = 0$

slide by S. K. Gupta

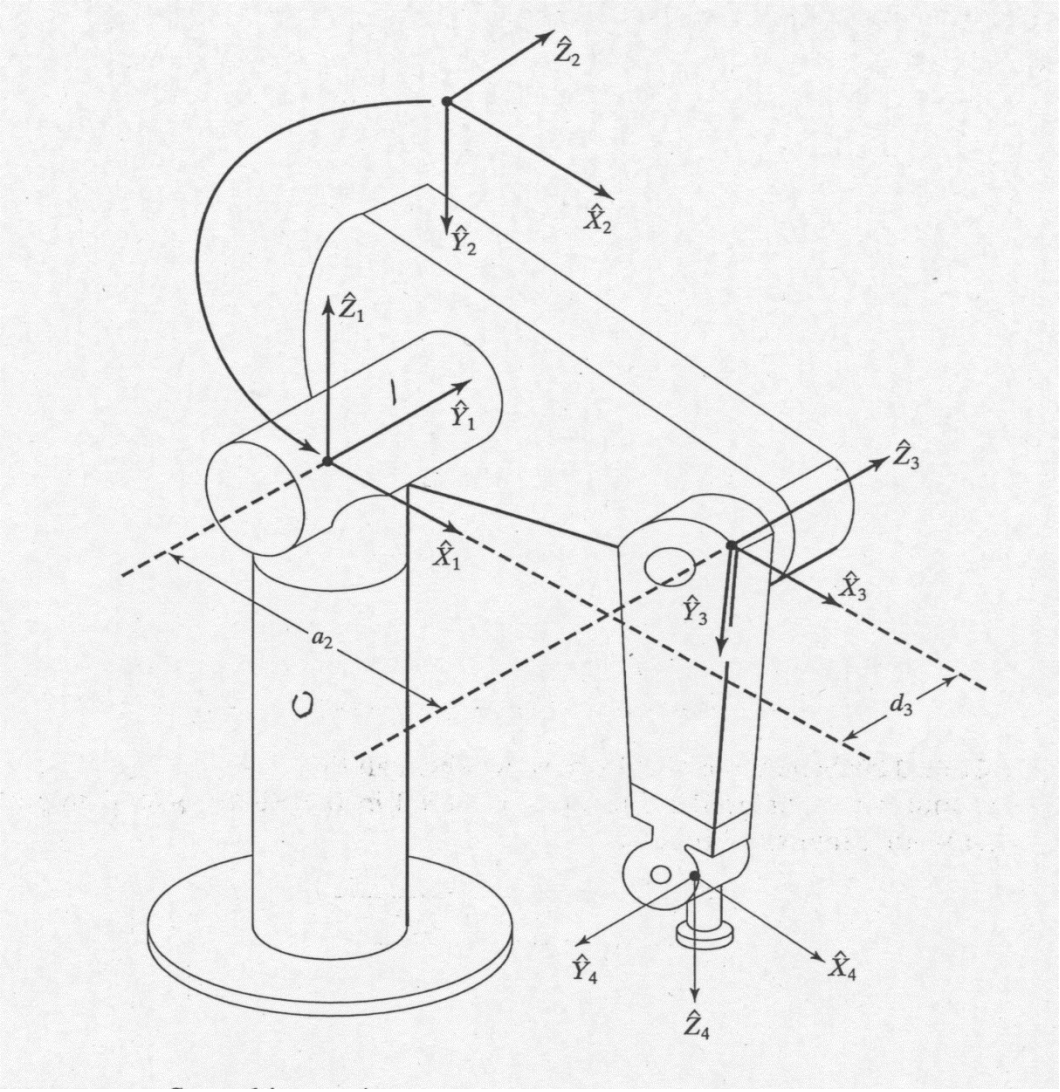
Final Transformations

$$
{}^{i-1}_{i}T = \begin{bmatrix} c\theta_i & -s\theta_i & 0 & a_{i-1} \\ s\theta_i c\alpha_{i-1} & c\theta_i c\alpha_{i-1} & -s\alpha_{i-1} & -s\alpha_{i-1} d_i \\ s\theta_i s\alpha_{i-1} & c\theta_i s\alpha_{i-1} & c\alpha_{i-1} & c\alpha_{i-1} d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}.
$$

slide by S. K. Gupta

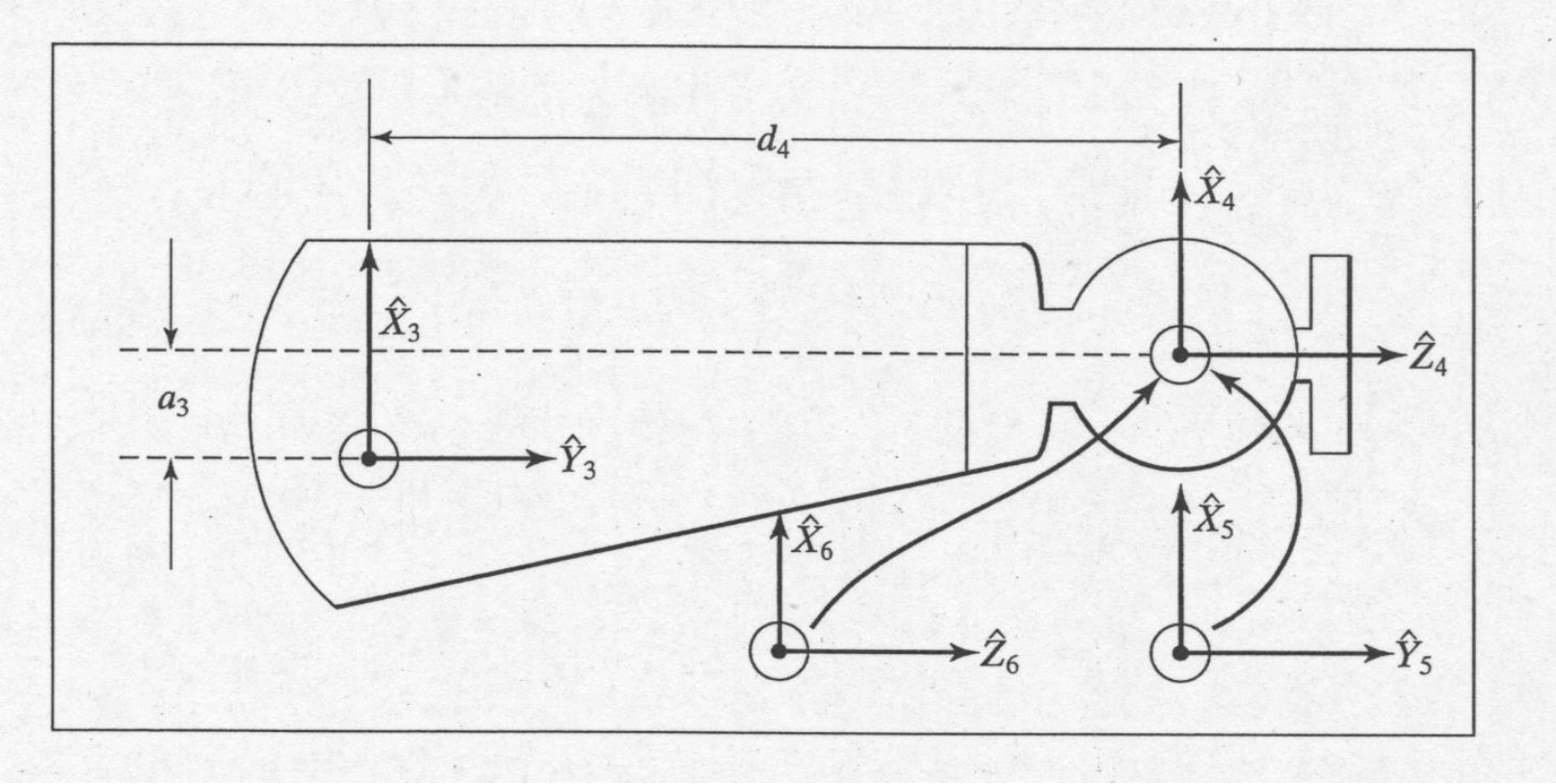
MARYLAND ROBOTICS CENTER
THE INSTITUTE FOR SYSTEMS RESEARCH

PUMA Robot with Assigned Frames



slide by S. K. Gupta

PUMA Robot with Assigned Frames (Cont.)



slide by S. K. Gupta

Link Parameters

i	α_{i-1}	a_{i-1}	d_i	θi
1	0	0	0	θ_1
2	$-90°$	0	0	θ_2
3	0	a_2	d_3	θ_3
4	$-90°$	a_3	d_4	θ_4
5	$90°$	0	0	θ_5
6	$-90°$	0	0	θ_6

slide by S. K. Gupta

Transforms for PUMA Robot

$$
{}^{0}_{1}T = \begin{bmatrix} c\theta_1 & -s\theta_1 & 0 & 0 \\ s\theta_1 & c\theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},
$$

$$
{}^{1}_{2}T = \begin{bmatrix} c\theta_2 & -s\theta_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s\theta_2 & -c\theta_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},
$$

$$
{}^{2}_{3}T = \begin{bmatrix} c\theta_3 & -s\theta_3 & 0 & a_2 \\ s\theta_3 & c\theta_3 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix},
$$

$$
{}^{3}_{4}T = \begin{bmatrix} c\theta_4 & -s\theta_4 & 0 & a_3 \\ 0 & 0 & 1 & d_4 \\ -s\theta_4 & -c\theta_4 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},
$$

$$
{}^{4}_{5}T = \begin{bmatrix} c\theta_5 & -s\theta_5 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s\theta_5 & c\theta_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},
$$

$$
{}^{5}_{6}T = \begin{bmatrix} c\theta_6 & -s\theta_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s\theta_6 & -c\theta_6 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.
$$

slide by S. K. Gupta

Final Transform from Base to Final Link

$$ {}^{0}_{6}T = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}. $$

slide by S. K. Gupta

Final Transform from Base to Final Link (Cont.)

$$
\begin{aligned}
r_{11} &= c_1[c_{23}(c_4c_5c_6 - s_4s_5) - s_{23}s_5c_5] + s_1(s_4c_5c_6 + c_4s_6),\\
r_{21} &= s_1[c_{23}(c_4c_5c_6 - s_4s_6) - s_{23}s_5c_6 - c_1(s_4c_5c_6 + c_4s_6),\\
r_{31} &= -s_{23}(c_4c_5c_6 - s_4s_6) - c_{23}s_5c_6,
\end{aligned}
$$

$$
\begin{aligned}
r_{12} &= c_1[c_{23}(-c_4c_5s_6 - s_4c_6) + s_{23}s_5s_6] + s_1(c_4c_6 - s_4c_5s_6),\\
r_{22} &= s_1[c_{23}(-c_4c_5s_6 - s_4c_6) + s_{23}s_5s_6] - c_1(c_4c_6 - s_4c_5s_6),\\
r_{32} &= -s_{23}(-c_4c_5s_6 - s_4c_6) + c_{23}s_5s_6,
\end{aligned}
$$

$$
\begin{aligned}
r_{13} &= -c_1(c_{23}c_4s_5 + s_{23}c_5) - s_1s_4s_5,\\
r_{23} &= -s_1(c_{23}c_4s_5 + s_{23}c_5) + c_1s_4s_5,\\
r_{33} &= s_{23}c_4s_5 - c_{23}c_5,
\end{aligned}
$$

$$
\begin{aligned}
p_x &= c_1[a_2c_2 + a_3c_{23} - d_4s_{23}] - d_3s_1,\\
p_y &= s_1[a_2c_2 + a_3c_{23} - d_4s_{23}] + d_3c_1,\\
p_z &= -a_3s_{23} - a_2s_2 - d_4c_{23}.
\end{aligned}
$$

slide by S. K. Gupta

Bibliography for this Lecture

- LaValle, S. M. Planning Algorithms, 2006
- Craig, J.J. Introduction to Robotics: Mechanics and Control, 2005

slide by S. K. Gupta