ROBOTICS: ADVANCED CONCEPTS & ANALYSIS

MODULE 3 - KINEMATICS OF SERIAL ROBOTS

Ashitava Ghosal[1]

[1]Department of Mechanical Engineering
&
Centre for Product Design and Manufacture
Indian Institute of Science
Bangalore 560 012, India
Email: asitava@mecheng.iisc.ernet.in

NPTEL, 2010

OUTLINE

Introduction

Review

- Serial manipulators: One end fixed $\rightarrow$ links and joints $\rightarrow$ free end with end-effector.
- Kinematics $\rightarrow$ motion of (rigid) links without considering force and torques.
- Kinematics $\rightarrow$ study of "geometry" of motion.
- Serial manipulators modeled using Denavit-Hartenberg parameters (see Module 2).
- Two main problems: Direct Kinematics and Inverse Kinematics

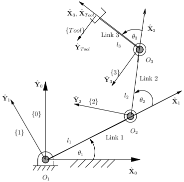
- Serial manipulators: One end fixed $\rightarrow$ links and joints $\rightarrow$ free end with end-effector.
- Kinematics $\rightarrow$ motion of (rigid) links without considering force and torques.
- Kinematics $\rightarrow$ study of "geometry" of motion.
- Serial manipulators modeled using Denavit-Hartenberg parameters (see Module 2).
- Two main problems: Direct Kinematics and Inverse Kinematics

- Serial manipulators: One end fixed $\rightarrow$ links and joints $\rightarrow$ free end with end-effector.
- Kinematics $\rightarrow$ motion of (rigid) links without considering force and torques.
- Kinematics $\rightarrow$ study of "geometry" of motion.
- Serial manipulators modeled using Denavit-Hartenberg parameters (see Module 2).
- Two main problems: Direct Kinematics and Inverse Kinematics

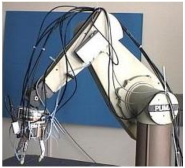
- Serial manipulators: One end fixed $\rightarrow$ links and joints $\rightarrow$ free end with end-effector.
- Kinematics $\rightarrow$ motion of (rigid) links without considering force and torques.
- Kinematics $\rightarrow$ study of "geometry" of motion.
- Serial manipulators modeled using Denavit-Hartenberg parameters (see Module 2).
- Two main problems: Direct Kinematics and Inverse Kinematics

Introduction

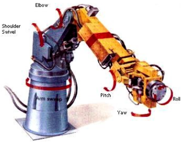
Examples of Serial Robots



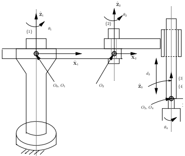
(a) 3 DOF planar Robot



(b) Six DOF PUMA Industrial Robot



(c) Six DOF T3 Robot



(d) 4 DOF SCARA Robot

- Grübler-Kutzbach's criterion

$$DOF = \lambda(N - J - 1) + \sum_{i=1}^{J} F_i \tag{1}$$

N – total number of links including the fixed link (or base),
J – total number of joints connecting *only* two links (if joint connects three links then it must be counted as two joints),
F_i – degrees of freedom at the i^{th} joint, and $\lambda = 6$ for spatial, 3 for planar manipulators and mechanisms.

- PUMA 560 – $N = 7$, $J = 6$, $F_1 = 1$, $\lambda = 6 \rightarrow$ $DOF = 6$
- Grübler criterion *does not* work for *over-constrained mechanisms* (see Mavroidas and Roth (1995), Gan and Pellegrino(2003), review paper by Gogu(2007)).

- Grübler-Kutzbach's criterion

$$DOF = \lambda(N - J - 1) + \sum_{i=1}^{J} F_i \qquad (1)$$

N – total number of links including the fixed link (or base),
J – total number of joints connecting *only* two links (if joint connects three links then it must be counted as two joints),
F_i – degrees of freedom at the i^{th} joint, and $\lambda = 6$ for spatial, 3 for planar manipulators and mechanisms.

- PUMA 560 – $N = 7,\ J = 6,\ F_1 = 1,\ \lambda = 6 \rightarrow\ DOF = 6$
- Grübler criterion *does not* work for *over-constrained mechanisms* (see Mavroidas and Roth (1995), Gan and Pellegrino(2003), review paper by Gogu(2007)).

- Grübler-Kutzbach's criterion

$$DOF = \lambda(N - J - 1) + \sum_{i=1}^{J} F_i \qquad (1)$$

N – total number of links including the fixed link (or base),
J – total number of joints connecting *only* two links (if joint connects three links then it must be counted as two joints),
F_i – degrees of freedom at the i^{th} joint, and $\lambda = 6$ for spatial, 3 for planar manipulators and mechanisms.

- PUMA 560 – $N = 7,\ J = 6,\ F_1 = 1,\ \lambda = 6 \rightarrow\ DOF = 6$
- Grübler criterion *does not* work for *over-constrained mechanisms* (see Mavroidas and Roth (1995), Gan and Pellegrino(2003), review paper by Gogu(2007)).

- *DOF* – the number of independent actuators.
- *DOF* – capability of a manipulator with respect to λ.
  1. $DOF = \lambda \rightarrow$ End-effector can be positioned and oriented arbitrarily.
  2. $DOF < \lambda \rightarrow \lambda - DOF$ relationships containing the position and orientation variables.
  3. $DOF > \lambda \rightarrow$ Position and orientation of the end-effector in ∞ ways – *redundant manipulators*.
- Serial manipulators with a fixed base, a free end-effector and two links connected by a joint – $N = J + 1$ and $DOF = \sum_{i=1}^{J} F_i$.
- All actuated joints are one- degree-of-freedom joints $\rightarrow J = DOF$.
- $J > DOF$ (in parallel manipulators) $\rightarrow J - DOF$ joints are *passive*.
- $J < DOF \rightarrow$ one or more of the actuated joints are multi-degree-of-freedom joints – rare in mechanical manipulators but common in biological joints actuated with muscles.

INTRODUCTION

DEGREES OF FREEDOM (CONTD.)

- *DOF* – the number of independent actuators.
- *DOF* – capability of a manipulator with respect to λ.
  1. $DOF = \lambda \rightarrow$ End-effector can be positioned and oriented arbitrarily.
  2. $DOF < \lambda \rightarrow \lambda - DOF$ relationships containing the position and orientation variables.
  3. $DOF > \lambda \rightarrow$ Position and orientation of the end-effector in ∞ ways – *redundant manipulators*.
- Serial manipulators with a fixed base, a free end-effector and two links connected by a joint – $N = J + 1$ and $DOF = \sum_{i=1}^{J} F_i$.
- All actuated joints are one- degree-of-freedom joints $\rightarrow J = DOF$.
- $J > DOF$ (in parallel manipulators) $\rightarrow J - DOF$ joints are *passive*.
- $J < DOF \rightarrow$ one or more of the actuated joints are multi-degree-of-freedom joints – rare in mechanical manipulators but common in biological joints actuated with muscles.

- *DOF* – the number of independent actuators.
- *DOF* – capability of a manipulator with respect to λ.
  1. $DOF = \lambda \rightarrow$ End-effector can be positioned and oriented arbitrarily.
  2. $DOF < \lambda \rightarrow \lambda - DOF$ relationships containing the position and orientation variables.
  3. $DOF > \lambda \rightarrow$ Position and orientation of the end-effector in ∞ ways – *redundant manipulators*.
- Serial manipulators with a fixed base, a free end-effector and two links connected by a joint – $N = J + 1$ and $DOF = \sum_{i=1}^{J} F_i$.
- All actuated joints are one- degree-of-freedom joints $\rightarrow J = DOF$.
- $J > DOF$ (in parallel manipulators) $\rightarrow J - DOF$ joints are *passive*.
- $J < DOF \rightarrow$ one or more of the actuated joints are multi-degree-of-freedom joints – rare in mechanical manipulators but common in biological joints actuated with muscles.

Introduction

Degrees of Freedom (contd.)

- J joint variables – θ_i's or d_i's form the *joint* space.
- Position and orientation variables form the *task* space.
- For planar motion, $\lambda = 3$ – Task space (x, y, ϕ).
- For spatial motion, $\lambda = 6$ – Task space $(x, y, z; [R])$
- *Actuator space*: due to mechanical linkages, gears, etc. between actuators and joints, joint variables are *not identical* to actuator variables.
- Dimension of actuator space is more than λ – manipulator is redundant.
- Dimension of actuator space is less than λ – manipulator is under-actuated.
- Kinematics – functional relationship between *joint space* and *task space*.

INTRODUCTION

DEGREES OF FREEDOM (CONTD.)

- J joint variables – θ_i's or d_i's form the *joint* space.
- Position and orientation variables form the *task* space.
- For planar motion, $\lambda = 3$ – Task space (x, y, ϕ).
- For spatial motion, $\lambda = 6$ – Task space $(x, y, z; [R])$
- *Actuator space*: due to mechanical linkages, gears, etc. between actuators and joints, joint variables are *not identical* to actuator variables.
- Dimension of actuator space is more than λ – manipulator is redundant.
- Dimension of actuator space is less than λ – manipulator is under-actuated.
- Kinematics – functional relationship between *joint space* and *task space*.

Introduction

Degrees of Freedom (contd.)

- J joint variables – θ_i's or d_i's form the *joint* space.
- Position and orientation variables form the *task* space.
- For planar motion, $\lambda = 3$ – Task space (x, y, ϕ).
- For spatial motion, $\lambda = 6$ – Task space $(x, y, z; [R])$
- *Actuator space*: due to mechanical linkages, gears, etc. between actuators and joints, joint variables are *not identical* to actuator variables.
- Dimension of actuator space is more than λ – manipulator is redundant.
- Dimension of actuator space is less than λ – manipulator is under-actuated.
- Kinematics – functional relationship between *joint space* and *task space*.

- J joint variables – θ_i's or d_i's form the *joint* space.
- Position and orientation variables form the *task* space.
- For planar motion, $\lambda = 3$ – Task space (x, y, ϕ).
- For spatial motion, $\lambda = 6$ – Task space $(x, y, z; [R])$
- *Actuator space*: due to mechanical linkages, gears, etc. between actuators and joints, joint variables are *not identical* to actuator variables.
- Dimension of actuator space is more than λ – manipulator is redundant.
- Dimension of actuator space is less than λ – manipulator is under-actuated.
- Kinematics – functional relationship between *joint space* and *task space*.

- **Direct Kinematics Problem**:
Given the constant D-H link parameters and the joint variable, a_{i-1}, α_{i-1}, d_i, and θ_i $i = 1, 2, ..n$, find the position and orientation of the last link in a fixed or reference coordinate system.
    - Most basic problem in serial manipulator kinematics.
    - Required to be solved for computer visualization of motion and in off-line programming systems.
    - Used in advanced control schemes.
- **Inverse Kinematics Problem**:
Given the constant D-H link parameters and the position and orientation of the end-effector ($\{n\}$) with respect to the fixed frame $\{0\}$, find the joint variables.
    - Harder than the direct kinematics problem.
    - Leads to the notion of *workspace* of a robot.
    - Required for computer visualization of motion and used in advanced control schemes.

- **Direct Kinematics Problem**:
  Given the constant D-H link parameters and the joint variable, a_{i-1}, α_{i-1}, d_i, and θ_i $i = 1, 2, ..n$, find the position and orientation of the last link in a fixed or reference coordinate system.
  - Most basic problem in serial manipulator kinematics.
  - Required to be solved for computer visualization of motion and in off-line programming systems.
  - Used in advanced control schemes.
- **Inverse Kinematics Problem**:
  Given the constant D-H link parameters and the position and orientation of the end-effector ($\{n\}$) with respect to the fixed frame $\{0\}$, find the joint variables.
  - Harder than the direct kinematics problem.
  - Leads to the notion of *workspace* of a robot.
  - Required for computer visualization of motion and used in advanced control schemes.

OUTLINE

1. CONTENTS
2. LECTURE 1
   - Introduction
   - Direct Kinematics of Serial Robots
3. LECTURE 2
   - Inverse Kinematics of Serial Robots
4. LECTURE 3
   - Inverse Kinematics of Serial Robots with $n < 6$
   - Inverse Kinematics of Serial Robots with $n > 6$
5. LECTURE 4*
   - Elimination Theory & Solution of Non-linear Equations
   - Inverse Kinematics of a General 6R Robot
6. MODULE 3 – ADDITIONAL MATERIAL
   - Problems, References and Suggested Reading

Direct Kinematics Problem

- Since all D-H parameters are known $\rightarrow$ All 4×4 link transforms ${}^{i-1}_{i}[T],\ i = 1,..,n$ are known (see Module 2, Lectures 2 and 3).
- With respect to $\{0\}$, the position and orientation of $\{n\}$ is ${}^{0}_{n}[T] = {}^{0}_{1}[T]^{1}_{2}[T]......^{n-1}_{n}[T]$.
- For another reference $\{Base\}$, ${}^{Base}_{n}[T] = {}^{Base}_{0}[T]^{0}_{n}[T]$. Note: ${}^{Base}_{0}[T]$ must be known.
- As in Module 2, the end-effector geometry *does not* appear in ${}^{0}_{n}[T]$ – ${}^{Base}_{Tool}[T] = {}^{Base}_{0}[T]^{0}_{n}[T]^{n}_{Tool}[T]$; ${}^{n}_{Tool}[T]$ is known.
- *One advantage of the used D-H convention*: Manipulator transform ${}^{0}_{n}[T]$ can be computed *only once* and need not be changed if location of $\{Base\}$ or the geometry of end-effector ${}^{n}_{Tool}[T]$ changes – Recall a robot can have a variety of end-effectors!

Direct Kinematics Problem

- Since all D-H parameters are known $\rightarrow$ All 4×4 link transforms ${}^{i-1}_{i}[T]$, $i = 1, .., n$ are known (see Module 2, Lectures 2 and 3).
- With respect to $\{0\}$, the position and orientation of $\{n\}$ is ${}^{0}_{n}[T] = {}^{0}_{1}[T]{}^{1}_{2}[T] \ldots\ldots {}^{n-1}_{n}[T]$.
- For another reference $\{Base\}$, ${}^{Base}_{n}[T] = {}^{Base}_{0}[T]{}^{0}_{n}[T]$. Note: ${}^{Base}_{0}[T]$ must be known.
- As in Module 2, the end-effector geometry *does not* appear in ${}^{0}_{n}[T]$ – ${}^{Base}_{Tool}[T] = {}^{Base}_{0}[T]{}^{0}_{n}[T]{}^{n}_{Tool}[T]$; ${}^{n}_{Tool}[T]$ is known.
- *One advantage of the used D-H convention*: Manipulator transform ${}^{0}_{n}[T]$ can be computed *only once* and need not be changed if location of $\{Base\}$ or the geometry of end-effector ${}^{n}_{Tool}[T]$ changes – Recall a robot can have a variety of end-effectors!

Direct Kinematics Problem

- Since all D-H parameters are known $\rightarrow$ All 4×4 link transforms ${}^{i-1}_{i}[T],\; i = 1, .., n$ are known (see Module 2, Lectures 2 and 3).
- With respect to $\{0\}$, the position and orientation of $\{n\}$ is ${}^{0}_{n}[T] = {}^{0}_{1}[T]{}^{1}_{2}[T] ...... {}^{n-1}_{n}[T]$.
- For another reference $\{Base\}$, ${}^{Base}_{n}[T] = {}^{Base}_{0}[T]{}^{0}_{n}[T]$. Note: ${}^{Base}_{0}[T]$ must be known.
- As in Module 2, the end-effector geometry *does not* appear in ${}^{0}_{n}[T]$ – ${}^{Base}_{Tool}[T] = {}^{Base}_{0}[T]{}^{0}_{n}[T]{}^{n}_{Tool}[T]$; ${}^{n}_{Tool}[T]$ is known.
- *One advantage of the used D-H convention*: Manipulator transform ${}^{0}_{n}[T]$ can be computed *only once* and need not be changed if location of $\{Base\}$ or the geometry of end-effector ${}^{n}_{Tool}[T]$ changes – Recall a robot can have a variety of end-effectors!

Direct Kinematics Problem

- Since all D-H parameters are known $\rightarrow$ All 4×4 link transforms ${}^{i-1}_{i}[T], \ i = 1, .., n$ are known (see Module 2, Lectures 2 and 3).
- With respect to $\{0\}$, the position and orientation of $\{n\}$ is ${}^{0}_{n}[T] = {}^{0}_{1}[T]{}^{1}_{2}[T] \ldots\ldots {}^{n-1}_{n}[T]$.
- For another reference $\{Base\}$, ${}^{Base}_{n}[T] = {}^{Base}_{0}[T]{}^{0}_{n}[T]$. Note: ${}^{Base}_{0}[T]$ must be known.
- As in Module 2, the end-effector geometry *does not* appear in ${}^{0}_{n}[T]$ – ${}^{Base}_{Tool}[T] = {}^{Base}_{0}[T]{}^{0}_{n}[T]{}^{n}_{Tool}[T]$; ${}^{n}_{Tool}[T]$ is known.
- *One advantage of the used D-H convention*: Manipulator transform ${}^{0}_{n}[T]$ can be computed *only once* and need not be changed if location of $\{Base\}$ or the geometry of end-effector ${}^{n}_{Tool}[T]$ changes – Recall a robot can have a variety of end-effectors!

Direct Kinematics Problem (Contd.)

- The *direct kinematics* problem can be *always* solved for *any* serial manipulator!
- The solution procedure is simple – involves only multiplication of matrices.

Examples 1: A planar 3R manipulator (see Lecture 3, Module 2).

- ϕ represents orientation of the tool.
- From ${}^{0}_{Tool}[T]$

$$
\begin{aligned}
x &= l_1 c_1 + l_2 c_{12} + l_3 c_{123} \\
y &= l_1 s_1 + l_2 s_{12} + l_3 s_{123} \\
\phi &= \theta_1 + \theta_2 + \theta_3
\end{aligned}
\tag{2}
$$

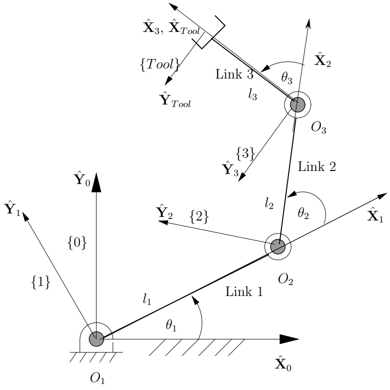


Figure 1: The planar 3R manipulator

Direct Kinematics Problem (Contd.)

Example 2 – A SCARA manipulator

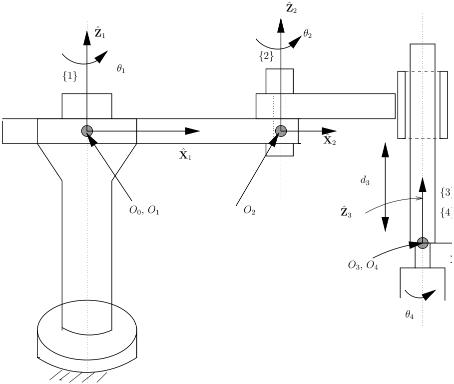


Figure 2: A SCARA manipulator

- ϕ represents orientation of the $\{4\}$.
- The position, (x, y, z), and orientation of $\{4\}$ is

$$\begin{aligned} x &= a_1 c_1 + a_2 c_{12} \\ y &= a_1 s_1 + a_2 s_{12} \\ z &= -d_3 \\ \phi &= \theta_1 + \theta_2 + \theta_4 \end{aligned} \tag{3}$$

Direct Kinematics Problem (Contd.)

Example 3 – The PUMA 560 manipulator

- ${}^0_6[T] = {}^0_3[T]\,{}^3_6[T]$ (see Lecture 3, Module 2)
- Orientation and position of ${}^0_6[T]$

$$
\begin{aligned}
r_{11} &= c_1\{c_{23}(c_4c_5c_6 - s_4s_6) - s_{23}s_5c_6\} + s_1(s_4c_5c_6 + c_4s_6)\\
r_{21} &= s_1\{c_{23}(c_4c_5c_6 - s_4s_6) - s_{23}s_5c_6\} - c_1(s_4c_5c_6 + c_4s_6)\\
r_{31} &= -s_{23}(c_4c_5c_6 - s_4s_6) - c_{23}s_5c_6\\
r_{12} &= c_1\{c_{23}(-c_4c_5s_6 - s_4c_6) + s_{23}s_5s_6\} + s_1(-s_4c_5s_6 + c_4c_6)\\
r_{22} &= s_1\{c_{23}(-c_4c_5s_6 - s_4c_6) + s_{23}s_5s_6\} - c_1(-s_4c_5s_6 + c_4c_6)\\
r_{32} &= -s_{23}(c_4c_5s_6 - s_4c_6) + c_{23}s_5s_6\\
r_{13} &= -c_1(c_{23}c_4s_5 + s_{23}c_5) - s_1s_4s_5\\
r_{23} &= -s_1(c_{23}c_4s_5 + s_{23}c_5) + c_1s_4s_5\\
r_{33} &= s_{23}c_4s_5 - c_{23}c_5\\
O_{6x} &= x = c_1(a_2c_2 + a_3c_{23} - d_4s_{23}) - d_3s_1\\
O_{6y} &= y = s_1(a_2c_2 + a_3c_{23} - d_4s_{23}) + d_3c_1\\
O_{6z} &= z = -a_2s_2 - a_3s_{23} - d_4c_{23}
\end{aligned}
\tag{4}
$$

Direct Kinematics Problem (Contd.)

Example 3 – The PUMA 560 manipulator

- ${}^0_6[T] = {}^0_3[T]{}^3_6[T]$ (see Lecture 3, Module 2)
- Orientation and position of ${}^0_6[T]$

$$
\begin{aligned}
r_{11} &= c_1\{c_{23}(c_4c_5c_6 - s_4s_6) - s_{23}s_5c_6\} + s_1(s_4c_5c_6 + c_4s_6) \\
r_{21} &= s_1\{c_{23}(c_4c_5c_6 - s_4s_6) - s_{23}s_5c_6\} - c_1(s_4c_5c_6 + c_4s_6) \\
r_{31} &= -s_{23}(c_4c_5c_6 - s_4s_6) - c_{23}s_5c_6 \\
r_{12} &= c_1\{c_{23}(-c_4c_5s_6 - s_4c_6) + s_{23}s_5s_6\} + s_1(-s_4c_5s_6 + c_4c_6) \\
r_{22} &= s_1\{c_{23}(-c_4c_5s_6 - s_4c_6) + s_{23}s_5s_6\} - c_1(-s_4c_5s_6 + c_4c_6) \\
r_{32} &= -s_{23}(c_4c_5s_6 - s_4c_6) + c_{23}s_5s_6 \\
r_{13} &= -c_1(c_{23}c_4s_5 + s_{23}c_5) - s_1s_4s_5 \\
r_{23} &= -s_1(c_{23}c_4s_5 + s_{23}c_5) + c_1s_4s_5 \\
r_{33} &= s_{23}c_4s_5 - c_{23}c_5 \\
O_{6x} &= x = c_1(a_2c_2 + a_3c_{23} - d_4s_{23}) - d_3s_1 \\
O_{6y} &= y = s_1(a_2c_2 + a_3c_{23} - d_4s_{23}) + d_3c_1 \\
O_{6z} &= z = -a_2s_2 - a_3s_{23} - d_4c_{23}
\end{aligned}
\tag{4}
$$

OUTLINE

1. CONTENTS
2. LECTURE 1
   - Introduction
   - Direct Kinematics of Serial Robots
3. LECTURE 2
   - Inverse Kinematics of Serial Robots
4. LECTURE 3
   - Inverse Kinematics of Serial Robots with $n < 6$
   - Inverse Kinematics of Serial Robots with $n > 6$
5. LECTURE 4*
   - Elimination Theory & Solution of Non-linear Equations
   - Inverse Kinematics of a General 6R Robot
6. MODULE 3 – ADDITIONAL MATERIAL
   - Problems, References and Suggested Reading

INTRODUCTION

- *Inverse Kinematics Problem* (restated): Given the constant D-H link parameters and ${}^0_n[T]$, find the joint variables $\theta_i,\ i = 1,..,n$.
  - For 3D motion, 6 task space variables – 3 position + 3 orientation – in ${}^0_n[T]$
  - For planar motion, 3 task space variables – 2 position + 1 orientation – in ${}^0_n[T]$
- Following cases possible:
  1. $n = 6$ for 3D motion or $n = 3$ for planar motion $\rightarrow$ same number of equations as unknowns.
  2. $n < 6$ for 3D motion or $n < 3$ for planar motion $\rightarrow$ number of task space variables larger than number of equations and hence there must be $6 - n$ ($3 - n$ for planar) relationships involving the task space variables.
  3. $n > 6$ for 3D motion or $n > 3$ for planar motion $\rightarrow$ more unknowns than equations and hence infinite number of solutions – redundant manipulators.
- Start with the simplest case of planar 3R manipulator.

INTRODUCTION

- *Inverse Kinematics Problem* (restated): Given the constant D-H link parameters and ${}^0_n[T]$, find the joint variables $\theta_i, \; i = 1,..,n$.
  - For 3D motion, 6 task space variables – 3 position + 3 orientation – in ${}^0_n[T]$
  - For planar motion, 3 task space variables – 2 position + 1 orientation – in ${}^0_n[T]$
- Following cases possible:
  1. $n = 6$ for 3D motion or $n = 3$ for planar motion $\rightarrow$ same number of equations as unknowns.
  2. $n < 6$ for 3D motion or $n < 3$ for planar motion $\rightarrow$ number of task space variables larger than number of equations and hence there must be $6 - n$ ($3 - n$ for planar) relationships involving the task space variables.
  3. $n > 6$ for 3D motion or $n > 3$ for planar motion $\rightarrow$ more unknowns than equations and hence infinite number of solutions – redundant manipulators.
- Start with the simplest case of planar 3R manipulator.

Introduction

- *Inverse Kinematics Problem* (restated): Given the constant D-H link parameters and ${}^0_n[T]$, find the joint variables $\theta_i, \; i = 1,..,n$.
  - For 3D motion, 6 task space variables – 3 position + 3 orientation – in ${}^0_n[T]$
  - For planar motion, 3 task space variables – 2 position + 1 orientation – in ${}^0_n[T]$
- Following cases possible:
  1. $n = 6$ for 3D motion or $n = 3$ for planar motion $\rightarrow$ same number of equations as unknowns.
  2. $n < 6$ for 3D motion or $n < 3$ for planar motion $\rightarrow$ number of task space variables larger than number of equations and hence there must be $6 - n$ $(3 - n$ for planar) relationships involving the task space variables.
  3. $n > 6$ for 3D motion or $n > 3$ for planar motion $\rightarrow$ more unknowns than equations and hence infinite number of solutions – redundant manipulators.
- Start with the simplest case of planar 3R manipulator.

Planar 3R Manipulator
Review

- Direct kinematics equations

$$
\begin{aligned}
x &= l_1 c_1 + l_2 c_{12} + l_3 c_{123} \\
y &= l_1 s_1 + l_2 s_{12} + l_3 s_{123} \\
\phi &= \theta_1 + \theta_2 + \theta_3
\end{aligned}
$$

- Inverse Kinematics: Given (x, y, ϕ) obtain θ_1, θ_2 and θ_3.
- Solution of system of *non-linear* transcendental equations.
- No general methods (as in linear equations) exists – solution procedure depends on problem.

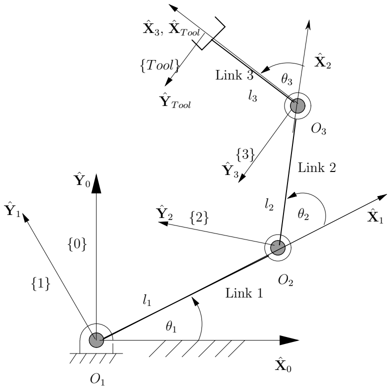


Figure 3: The planar 3R manipulator

Planar 3R Manipulator

Inverse Kinematics Algorithm

- Define $X = x - l_3 c_\phi$ and $Y = y - l_3 s_\phi$ – X and Y are known since x, y, ϕ and l_3 are known.
- Squaring and adding

$$X^2 + Y^2 = l_1^2 + l_2^2 + 2 l_1 l_2 c_2 \tag{5}$$

- From equation (5)

$$\theta_2 = \pm \cos^{-1}\left(\frac{X^2 + Y^2 - l_1^2 - l_2^2}{2 l_1 l_2}\right) \tag{6}$$

- Once θ_2 is known

$$\theta_1 = \mathrm{Atan2}(Y, X) - \mathrm{Atan2}(k_2, k_1) \tag{7}$$

where $k_2 = l_2 s_2$ and $k_1 = l_1 + l_2 c_2$. Note: $\mathrm{Atan2}(y, x)$ is the four quadrant arc-tangent function and $\theta_1 \in [0, 2\pi]$

- Finally, θ_3 is obtained from

$$\theta_3 = \phi - \theta_1 - \theta_2 \tag{8}$$

- Define $X = x - l_3 c_\phi$ and $Y = y - l_3 s_\phi$ – X and Y are known since x, y, ϕ and l_3 are known.
- Squaring and adding

$$X^2 + Y^2 = l_1^2 + l_2^2 + 2 l_1 l_2 c_2 \tag{5}$$

- From equation (5)

$$\theta_2 = \pm \cos^{-1}\left(\frac{X^2 + Y^2 - l_1^2 - l_2^2}{2 l_1 l_2}\right) \tag{6}$$

- Once θ_2 is known

$$\theta_1 = \mathrm{Atan2}(Y, X) - \mathrm{Atan2}(k_2, k_1) \tag{7}$$

where $k_2 = l_2 s_2$ and $k_1 = l_1 + l_2 c_2$. Note: $\mathrm{Atan2}(y, x)$ is the four quadrant arc-tangent function and $\theta_1 \in [0, 2\pi]$

- Finally, θ_3 is obtained from

$$\theta_3 = \phi - \theta_1 - \theta_2 \tag{8}$$

- Define $X = x - l_3 c_\phi$ and $Y = y - l_3 s_\phi$ – X and Y are known since x, y, ϕ and l_3 are known.
- Squaring and adding

$$X^2 + Y^2 = l_1^2 + l_2^2 + 2 l_1 l_2 c_2 \tag{5}$$

- From equation (5)

$$\theta_2 = \pm \cos^{-1} \left(\frac{X^2 + Y^2 - l_1^2 - l_2^2}{2 l_1 l_2} \right) \tag{6}$$

- Once θ_2 is known

$$\theta_1 = \mathrm{Atan2}(Y, X) - \mathrm{Atan2}(k_2, k_1) \tag{7}$$

where $k_2 = l_2 s_2$ and $k_1 = l_1 + l_2 c_2$. Note: $\mathrm{Atan2}(y, x)$ is the four quadrant arc-tangent function and $\theta_1 \in [0, 2\pi]$

- Finally, θ_3 is obtained from

$$\theta_3 = \phi - \theta_1 - \theta_2 \tag{8}$$

- Define $X = x - l_3 c_\phi$ and $Y = y - l_3 s_\phi$ – X and Y are known since x, y, ϕ and l_3 are known.
- Squaring and adding

$$X^2 + Y^2 = l_1^2 + l_2^2 + 2 l_1 l_2 c_2 \qquad (5)$$

- From equation (5)

$$\theta_2 = \pm \cos^{-1}\left(\frac{X^2 + Y^2 - l_1^2 - l_2^2}{2 l_1 l_2}\right) \qquad (6)$$

- Once θ_2 is known

$$\theta_1 = \mathrm{Atan2}(Y, X) - \mathrm{Atan2}(k_2, k_1) \qquad (7)$$

where $k_2 = l_2 s_2$ and $k_1 = l_1 + l_2 c_2$. Note: $\mathrm{Atan2}(y, x)$ is the four quadrant arc-tangent function and $\theta_1 \in [0, 2\pi]$

- Finally, θ_3 is obtained from

$$\theta_3 = \phi - \theta_1 - \theta_2 \qquad (8)$$

- Define $X = x - l_3 c_\phi$ and $Y = y - l_3 s_\phi$ – X and Y are known since x, y, ϕ and l_3 are known.
- Squaring and adding

$$X^2 + Y^2 = l_1^2 + l_2^2 + 2 l_1 l_2 c_2 \tag{5}$$

- From equation (5)

$$\theta_2 = \pm \cos^{-1}\left(\frac{X^2 + Y^2 - l_1^2 - l_2^2}{2 l_1 l_2}\right) \tag{6}$$

- Once θ_2 is known

$$\theta_1 = \mathrm{Atan2}(Y, X) - \mathrm{Atan2}(k_2, k_1) \tag{7}$$

where $k_2 = l_2 s_2$ and $k_1 = l_1 + l_2 c_2$. Note: $\mathrm{Atan2}(y, x)$ is the four quadrant arc-tangent function and $\theta_1 \in [0, 2\pi]$

- Finally, θ_3 is obtained from

$$\theta_3 = \phi - \theta_1 - \theta_2 \tag{8}$$

- Workspace: All (x, y, ϕ) such that inverse kinematics solution *exists*.
- From equation (6) $-1 \leq (\frac{X^2+Y^2-l_1^2-l_2^2}{2l_1l_2}) \leq +1$ or

$$(l_1 - l_2)^2 \leq (X^2 + Y^2) \leq (l_1 + l_2)^2 \quad (9)$$

where $X = x - l_3 c_\phi$ and $Y = y - l_3 s_\phi$.

- Figure 4 shows the region in $\{x, y, \phi\}$ space where the above inequalities are satisfied and the inverse kinematics solution *exists*.

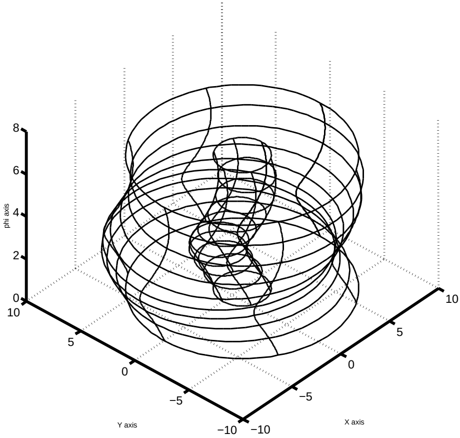


Figure 4: Workspace of a planar 3R robot

- Workspace: All (x, y, ϕ) such that inverse kinematics solution *exists*.
- From equation (6) $-1 \leq (\frac{X^2+Y^2-l_1^2-l_2^2}{2l_1 l_2}) \leq +1$ or

$$(l_1 - l_2)^2 \leq (X^2 + Y^2) \leq (l_1 + l_2)^2 \quad (9)$$

where $X = x - l_3 c_\phi$ and $Y = y - l_3 s_\phi$.

- Figure 4 shows the region in $\{x, y, \phi\}$ space where the above inequalities are satisfied and the inverse kinematics solution *exists*.

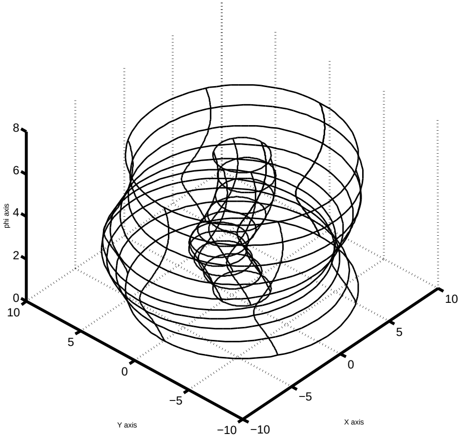


Figure 4: Workspace of a planar 3R robot

Planar 3R Manipulator

Existence of IK Solution – Workspace

- Workspace: All (x, y, ϕ) such that inverse kinematics solution *exists*.
- From equation (6)
  $-1 \leq (\frac{X^2+Y^2-l_1^2-l_2^2}{2l_1l_2}) \leq +1$ or

  $$(l_1-l_2)^2 \leq (X^2+Y^2) \leq (l_1+l_2)^2 \qquad (9)$$

  where $X = x - l_3 c_\phi$ and $Y = y - l_3 s_\phi$.
- Figure 4 shows the region in $\{x, y, \phi\}$ space where the above inequalities are satisfied and the inverse kinematics solution *exists*.

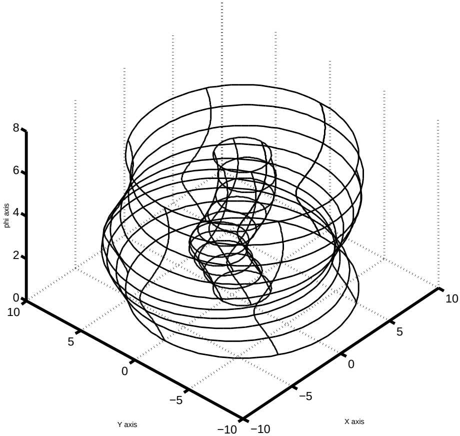


Figure 4: Workspace of a planar 3R robot

Planar 3R Manipulator

Workspace (contd.)

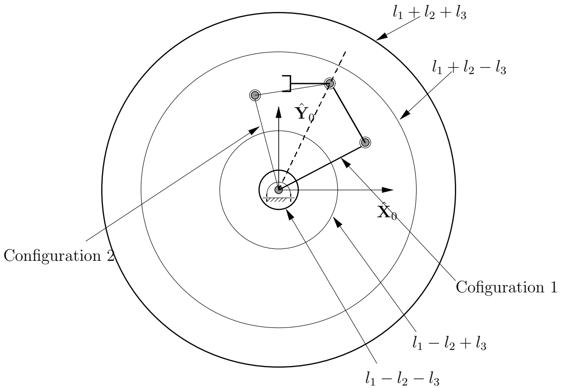


Figure 5: Projection of workspace of a planar 3R robot

- Projection of the workspace on $\hat{X}_0 - \hat{Y}_0$ plane for $l_1 > l_2 > l_3$ – four circles of radii $l_1 + l_2 + l_3$, $l_1 + l_2 - l_3$, $l_1 - l_2 + l_3$ and $l_1 - l_2 - l_3$.

- *Reachable Workspace*: All (x, y) between maximum reach $(l_1 + l_2 + l_3)$ and minimum reach $(l_1 - l_2 - l_3)$.
- *Dexterous Workspace*: All (x, y) between maximum reach $(l_1 + l_2 - l_3)$ and minimum reach $(l_1 - l_2 + l_3)$.
- All points *inside* dexterous workspace can be reached with any ϕ (Kumar and Waldron, 1980).
- As size of end-effector l_3 increases, reachable workspace increases and dexterous workspace decreases!
- Intuitively correct – with a long stick one can reach far but with less freedom in orientation.

- *Reachable Workspace*: All (x, y) between maximum reach $(l_1 + l_2 + l_3)$ and minimum reach $(l_1 - l_2 - l_3)$.
- *Dexterous Workspace*: All (x, y) between maximum reach $(l_1 + l_2 - l_3)$ and minimum reach $(l_1 - l_2 + l_3)$.
- All points *inside* dexterous workspace can be reached with any ϕ (Kumar and Waldron, 1980).
- As size of end-effector l_3 increases, reachable workspace increases and dexterous workspace decreases!
- Intuitively correct – with a long stick one can reach far but with less freedom in orientation.

Planar 3R Manipulator

Workspace (Contd.)

- *Reachable Workspace*: All (x, y) between maximum reach $(l_1 + l_2 + l_3)$ and minimum reach $(l_1 - l_2 - l_3)$.
- *Dexterous Workspace*: All (x, y) between maximum reach $(l_1 + l_2 - l_3)$ and minimum reach $(l_1 - l_2 + l_3)$.
- All points *inside* dexterous workspace can be reached with any ϕ (Kumar and Waldron, 1980).
- As size of end-effector l_3 increases, reachable workspace increases and dexterous workspace decreases!
- Intuitively correct – with a long stick one can reach far but with less freedom in orientation.

Planar 3R Manipulator

Workspace (Contd.)

- *Reachable Workspace*: All (x, y) between maximum reach $(l_1 + l_2 + l_3)$ and minimum reach $(l_1 - l_2 - l_3)$.
- *Dexterous Workspace*: All (x, y) between maximum reach $(l_1 + l_2 - l_3)$ and minimum reach $(l_1 - l_2 + l_3)$.
- All points *inside* dexterous workspace can be reached with any ϕ (Kumar and Waldron, 1980).
- As size of end-effector l_3 increases, reachable workspace increases and dexterous workspace decreases!
- Intuitively correct – with a long stick one can reach far but with less freedom in orientation.

PLANAR 3R MANIPULATOR

UNIQUENESS OF INVERSE KINEMATICS SOLUTION

- Equation (6) revisited:

$$\theta_2 = \pm \cos^{-1}\left(\frac{X^2 + Y^2 - l_1^2 - l_2^2}{2 l_1 l_2}\right)$$

  For any (X, Y) two values of θ_2.
- A given (X, Y) can be achieved by two configurations – shown in figure 5.
- For planar 3R manipulator – (x, y, ϕ) yields two sets of values of $\theta_i, \ i = 1, 2, 3$
- Inverse kinematics problem *does not* give unique solution – compare with direct kinematics!
- Existence and uniqueness issues important and non-trivial in solutions of *all* non-linear equations.

- Equation (6) revisited:

$$\theta_2 = \pm \cos^{-1}\left(\frac{X^2 + Y^2 - l_1^2 - l_2^2}{2 l_1 l_2}\right)$$

  For any (X, Y) two values of θ_2.

- A given (X, Y) can be achieved by two configurations – shown in figure 5.
- For planar 3R manipulator – (x, y, ϕ) yields two sets of values of $\theta_i, \; i = 1, 2, 3$
- Inverse kinematics problem *does not* give unique solution – compare with direct kinematics!
- Existence and uniqueness issues important and non-trivial in solutions of *all* non-linear equations.

- Equation (6) revisited:

$$\theta_2 = \pm \cos^{-1}\left(\frac{X^2 + Y^2 - l_1^2 - l_2^2}{2 l_1 l_2}\right)$$

  For any (X, Y) two values of θ_2.

- A given (X, Y) can be achieved by two configurations – shown in figure 5.
- For planar 3R manipulator – (x, y, ϕ) yields two sets of values of $\theta_i, \; i = 1, 2, 3$
- Inverse kinematics problem *does not* give unique solution – compare with direct kinematics!
- Existence and uniqueness issues important and non-trivial in solutions of *all* non-linear equations.

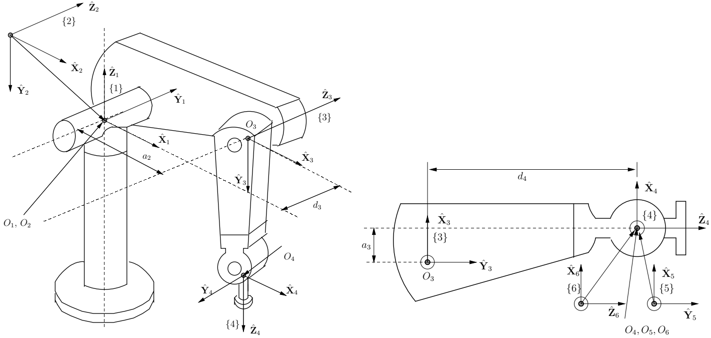
- Equation (6) revisited:

$$\theta_2 = \pm \cos^{-1}\left(\frac{X^2 + Y^2 - l_1^2 - l_2^2}{2 l_1 l_2}\right)$$

  For any (X, Y) two values of θ_2.
- A given (X, Y) can be achieved by two configurations – shown in figure 5.
- For planar 3R manipulator – (x, y, ϕ) yields two sets of values of $\theta_i, \; i = 1, 2, 3$
- Inverse kinematics problem *does not* give unique solution – compare with direct kinematics!
- Existence and uniqueness issues important and non-trivial in solutions of *all* non-linear equations.

PUMA 560 MANIPULATOR

REVIEW



(a) The PUMA 560 manipulator *(b) PUMA 560 - forearm and wrist*

Figure 6: The PUMA 560 manipulator

- Origins of {4}, {5} and {6} are *coincident* – *wrist* point.

- Position vector ${}^0\mathbf{O}_6$ of the wrist point is only a function of θ_1, θ_2 and θ_3.
- From equation (4),

$$
\begin{aligned}
x &= c_1(a_2c_2 + a_3c_{23} - d_4s_{23}) - d_3s_1 \\
y &= s_1(a_2c_2 + a_3c_{23} - d_4s_{23}) + d_3c_1 \\
z &= -a_2s_2 - a_3s_{23} - d_4c_{23}
\end{aligned} \tag{10}
$$

  where c_i, s_i denote $\cos(\theta_i)$, $\sin(\theta_i)$, respectively and a_2,a_3, d_3, d_4 denote constant D-H parameters.
- From the first two equations: $-s_1x + c_1y = d_3$
- Transcendental equation in sine and cosine of θ_1 – can be solved to obtain two sets of values for θ_1.

PUMA 560 MANIPULATOR

INVERSE KINEMATICS ALGORITHM

- Position vector ${}^0\mathbf{O}_6$ of the wrist point is only a function of θ_1, θ_2 and θ_3.
- From equation (4),

$$
\begin{aligned}
x &= c_1(a_2c_2 + a_3c_{23} - d_4s_{23}) - d_3s_1 \\
y &= s_1(a_2c_2 + a_3c_{23} - d_4s_{23}) + d_3c_1 \\
z &= -a_2s_2 - a_3s_{23} - d_4c_{23}
\end{aligned}
\tag{10}
$$

where c_i, s_i denote $\cos(\theta_i)$, $\sin(\theta_i)$, respectively and a_2,a_3, d_3, d_4 denote constant D-H parameters.

- From the first two equations: $-s_1x + c_1y = d_3$
- Transcendental equation in sine and cosine of θ_1 – can be solved to obtain two sets of values for θ_1.

- Position vector ${}^0\mathbf{O}_6$ of the wrist point is only a function of θ_1, θ_2 and θ_3.
- From equation (4),

$$\begin{aligned} x &= c_1(a_2c_2 + a_3c_{23} - d_4s_{23}) - d_3s_1 \\ y &= s_1(a_2c_2 + a_3c_{23} - d_4s_{23}) + d_3c_1 \\ z &= -a_2s_2 - a_3s_{23} - d_4c_{23} \end{aligned} \tag{10}$$

  where c_i, s_i denote $\cos(\theta_i)$, $\sin(\theta_i)$, respectively and a_2,a_3, d_3, d_4 denote constant D-H parameters.
- From the first two equations: $-s_1x + c_1y = d_3$
- Transcendental equation in sine and cosine of θ_1 – can be solved to obtain two sets of values for θ_1.

- Position vector ${}^0\mathbf{O}_6$ of the wrist point is only a function of θ_1, θ_2 and θ_3.
- From equation (4),

$$
\begin{aligned}
x &= c_1(a_2c_2 + a_3c_{23} - d_4s_{23}) - d_3s_1 \\
y &= s_1(a_2c_2 + a_3c_{23} - d_4s_{23}) + d_3c_1 \\
z &= -a_2s_2 - a_3s_{23} - d_4c_{23}
\end{aligned}
\tag{10}
$$

  where c_i, s_i denote $\cos(\theta_i)$, $\sin(\theta_i)$, respectively and a_2,a_3, d_3, d_4 denote constant D-H parameters.
- From the first two equations: $-s_1x + c_1y = d_3$
- Transcendental equation in sine and cosine of θ_1 – can be solved to obtain two sets of values for θ_1.

- Substitute *tangent half-angle* formulas from trigonometry

$$x_1 = \tan\frac{\theta_1}{2}, \quad c_1 = \frac{1 - x_1^2}{1 + x_1^2}, \quad s_1 = \frac{2x_1}{1 + x_1^2} \tag{11}$$

in $\quad -s_1 x + c_1 y = d_3$
gives $x_1^2(d_3 + y) + (2x)x_1 + (d_3 - y) = 0$

- Solve quadratic in x_1 and take $\tan^{-1}$ to get

$$\theta_1 = 2\tan^{-1}\left(\frac{-x \pm \sqrt{x^2 + y^2 - d_3^2}}{y + d_3}\right) \tag{12}$$

- Note 1: $\tan^{-1}$ gives an angle between 0 and π and hence $0 \leq \theta_1 \leq 2\pi$.
- Note 2: Two possible values of θ_1 due to the $\pm$ before square root.

- Substitute *tangent half-angle* formulas from trigonometry

$$x_1 = \tan\frac{\theta_1}{2}, \quad c_1 = \frac{1 - x_1^2}{1 + x_1^2}, \quad s_1 = \frac{2x_1}{1 + x_1^2} \tag{11}$$

in $-s_1 x + c_1 y = d_3$
gives $x_1^2(d_3 + y) + (2x)x_1 + (d_3 - y) = 0$

- Solve quadratic in x_1 and take $\tan^{-1}$ to get

$$\theta_1 = 2\tan^{-1}\left(\frac{-x \pm \sqrt{x^2 + y^2 - d_3^2}}{y + d_3}\right) \tag{12}$$

- Note 1: $\tan^{-1}$ gives an angle between 0 and π and hence $0 \le \theta_1 \le 2\pi$.
- Note 2: Two possible values of θ_1 due to the $\pm$ before square root.

- Substitute *tangent half-angle* formulas from trigonometry

$$x_1 = \tan\frac{\theta_1}{2}, \quad c_1 = \frac{1-x_1^2}{1+x_1^2}, \quad s_1 = \frac{2x_1}{1+x_1^2} \tag{11}$$

in $\quad -s_1 x + c_1 y = d_3$
gives $x_1^2(d_3+y) + (2x)x_1 + (d_3-y) = 0$

- Solve quadratic in x_1 and take $\tan^{-1}$ to get

$$\theta_1 = 2\tan^{-1}\left(\frac{-x \pm \sqrt{x^2+y^2-d_3^2}}{y+d_3}\right) \tag{12}$$

- Note 1: $\tan^{-1}$ gives an angle between 0 and π and hence $0 \leq \theta_1 \leq 2\pi$.
- Note 2: Two possible values of θ_1 due to the $\pm$ before square root.

PUMA 560 MANIPULATOR

INVERSE KINEMATICS ALGORITHM (CONTD.)

- Squaring and adding expressions for x, y and z
  $x^2 + y^2 + z^2 = d_3^2 + a_2^2 + a_3^2 + d_4^3 + 2a_2 a_3 c_3 - 2a_2 d_4 s_3$
- Using tangent half-angle formulas

$$\theta_3 = 2\tan^{-1}\left(\frac{-d_4 \pm \sqrt{d_4^2 + a_3^2 - K^2}}{K + a_3}\right) \tag{13}$$

  $K = (1/2a_2)(x^2 + y^2 + z^2 - d_3^2 - a_2^2 - a_3^2 - d_4^2)$.
- Two sets of values of θ_3.
- The expression for z is only a function of θ_2 and θ_3. Hence,
  $-s_2(a_2 + a_3 c_3 - d_4 s_3) + c_2(-a_3 s_3 - d_4 c_3) = z$
- Solve for θ_2 (known θ_3) using tangent half-angle substitutions

$$\theta_2 = 2\tan^{-1}\left(\frac{-a_2 - a_3 c_3 + d_4 s_3 \pm \sqrt{a_2^2 + a_3^2 + d_4^2 + 2a_2(a_3 c_3 - d_4 s_3) - z^2}}{z - (a_3 s_3 + d_4 c_3)}\right) \tag{14}$$

- Two possible values of θ_2 in the range $[0, 2\pi]$.

PUMA 560 Manipulator

Inverse kinematics algorithm (contd.)

- Squaring and adding expressions for x, y and z
  $x^2 + y^2 + z^2 = d_3^2 + a_2^2 + a_3^2 + d_4^3 + 2a_2 a_3 c_3 - 2a_2 d_4 s_3$
- Using tangent half-angle formulas

$$\theta_3 = 2\tan^{-1}\left(\frac{-d_4 \pm \sqrt{d_4^2 + a_3^2 - K^2}}{K + a_3}\right) \tag{13}$$

$K = (1/2a_2)(x^2 + y^2 + z^2 - d_3^2 - a_2^2 - a_3^2 - d_4^2)$.

- Two sets of values of θ_3.
- The expression for z is only a function of θ_2 and θ_3. Hence,
  $-s_2(a_2 + a_3 c_3 - d_4 s_3) + c_2(-a_3 s_3 - d_4 c_3) = z$
- Solve for θ_2 (known θ_3) using tangent half-angle substitutions

$$\theta_2 = 2\tan^{-1}\left(\frac{-a_2 - a_3 c_3 + d_4 s_3 \pm \sqrt{a_2^2 + a_3^2 + d_4^2 + 2a_2(a_3 c_3 - d_4 s_3) - z^2}}{z - (a_3 s_3 + d_4 c_3)}\right) \tag{14}$$

- Two possible values of θ_2 in the range $[0, 2\pi]$.

INVERSE KINEMATICS ALGORITHM (CONTD.)

- Squaring and adding expressions for x, y and z
  $x^2+y^2+z^2 = d_3^2+a_2^2+a_3^2+d_4^3+2a_2a_3c_3-2a_2d_4s_3$
- Using tangent half-angle formulas

$$\theta_3 = 2\tan^{-1}\left(\frac{-d_4 \pm \sqrt{d_4^2+a_3^2-K^2}}{K+a_3}\right) \tag{13}$$

  $K = (1/2a_2)(x^2+y^2+z^2-d_3^2-a_2^2-a_3^2-d_4^2)$.
- Two sets of values of θ_3.
- The expression for z is only a function of θ_2 and θ_3. Hence,
  $-s_2(a_2+a_3c_3-d_4s_3)+c_2(-a_3s_3-d_4c_3) = z$
- Solve for θ_2 (known θ_3) using tangent half-angle substitutions

$$\theta_2 = 2\tan^{-1}\left(\frac{-a_2-a_3c_3+d_4s_3 \pm \sqrt{a_2^2+a_3^2+d_4^2+2a_2(a_3c_3-d_4s_3)-z^2}}{z-(a_3s_3+d_4c_3)}\right) \tag{14}$$

- Two possible values of θ_2 in the range $[0, 2\pi]$.

PUMA 560 MANIPULATOR

INVERSE KINEMATICS ALGORITHM (CONTD.)

- Squaring and adding expressions for x, y and z
  $x^2 + y^2 + z^2 = d_3^2 + a_2^2 + a_3^2 + d_4^3 + 2a_2 a_3 c_3 - 2a_2 d_4 s_3$
- Using tangent half-angle formulas

$$\theta_3 = 2\tan^{-1}\left(\frac{-d_4 \pm \sqrt{d_4^2 + a_3^2 - K^2}}{K + a_3}\right) \tag{13}$$

  $K = (1/2a_2)(x^2 + y^2 + z^2 - d_3^2 - a_2^2 - a_3^2 - d_4^2)$.
- Two sets of values of θ_3.
- The expression for z is only a function of θ_2 and θ_3. Hence,
  $-s_2(a_2 + a_3 c_3 - d_4 s_3) + c_2(-a_3 s_3 - d_4 c_3) = z$
- Solve for θ_2 (known θ_3) using tangent half-angle substitutions

$$\theta_2 = 2\tan^{-1}\left(\frac{-a_2 - a_3 c_3 + d_4 s_3 \pm \sqrt{a_2^2 + a_3^2 + d_4^2 + 2a_2(a_3 c_3 - d_4 s_3) - z^2}}{z - (a_3 s_3 + d_4 c_3)}\right) \tag{14}$$

- Two possible values of θ_2 in the range $[0, 2\pi]$.

PUMA 560 Manipulator

Inverse kinematics algorithm (contd.)

- To obtain θ_4, θ_5 and θ_6, form

$$ {}^3_6[R] = \begin{pmatrix} c_4 c_5 c_6 - s_4 s_6 & -c_4 c_5 s_6 - s_4 c_6 & -c_4 s_5 \\ s_5 c_6 & -s_5 s_6 & c_5 \\ -s_4 c_5 c_6 - c_4 s_6 & s_4 c_5 s_6 - c_4 c_6 & s_4 s_5 \end{pmatrix} \quad (15) $$

- Since

$$ {}^3_6[R] = {}^0_3[R]^T {}^0_6[R] \quad (16) $$

and since θ_1, θ_2 and θ_3 are known, right-hand side is known!

- Compare known right-hand side with elements of ${}^3_6[R]$ and obtain θ_4, θ_5 and θ_6
- Similar to $Z - Y - Z$ Euler angles with Y rotation of $-\theta_5$ – see Lecture 2, Module 2.

- To obtain θ_4, θ_5 and θ_6, form

$$ {}^3_6[R] = \begin{pmatrix} c_4c_5c_6 - s_4s_6 & -c_4c_5s_6 - s_4c_6 & -c_4s_5 \\ s_5c_6 & -s_5s_6 & c_5 \\ -s_4c_5c_6 - c_4s_6 & s_4c_5s_6 - c_4c_6 & s_4s_5 \end{pmatrix} \quad (15) $$

- Since

$$ {}^3_6[R] = {}^0_3[R]^T\, {}^0_6[R] \quad (16) $$

and since θ_1, θ_2 and θ_3 are known, right-hand side is known!

- Compare known right-hand side with elements of ${}^3_6[R]$ and obtain θ_4, θ_5 and θ_6
- Similar to $Z - Y - Z$ Euler angles with Y rotation of $-\theta_5$ – see Lecture 2, Module 2.

- To obtain θ_4, θ_5 and θ_6, form

$$ {}^3_6[R] = \begin{pmatrix} c_4c_5c_6 - s_4s_6 & -c_4c_5s_6 - s_4c_6 & -c_4s_5 \\ s_5c_6 & -s_5s_6 & c_5 \\ -s_4c_5c_6 - c_4s_6 & s_4c_5s_6 - c_4c_6 & s_4s_5 \end{pmatrix} \quad (15) $$

- Since

$$ {}^3_6[R] = {}^0_3[R]^T \, {}^0_6[R] \quad (16) $$

and since θ_1, θ_2 and θ_3 are known, right-hand side is known!

- Compare known right-hand side with elements of ${}^3_6[R]$ and obtain θ_4, θ_5 and θ_6
- Similar to $Z - Y - Z$ Euler angles with Y rotation of $-\theta_5$ – see Lecture 2, Module 2.

PUMA 560 Manipulator

Inverse kinematics algorithm (contd.)

Algorithm $r_{ij} \Rightarrow \theta_4,\ \theta_5$ **and** θ_6

If $r_{23} \neq \pm 1$, *then*

$$\theta_5 = \text{Atan2}(\pm\sqrt{(r_{21}^2 + r_{22}^2)}, r_{23})$$

$$\theta_4 = \text{Atan2}(r_{33}/s_5, -r_{13}/s_5),$$

$$\theta_6 = \text{Atan2}(-r_{22}/s_5, r_{21}/s_5)$$

Else

If $r_{23} = 1$, *then*

$$\theta_4 = 0$$

$$\theta_5 = 0,$$

$$\theta_6 = \text{Atan2}(-r_{12}, r_{11}),$$

If $r_{23} = -1$, *then*

$$\theta_4 = 0$$

$$\theta_5 = \pi,$$

$$\theta_6 = -\text{Atan2}(r_{12}, -r_{11}),$$

- From equation (11) two sets of θ_1
- From equation (13) two sets of θ_3
- Since θ_3 appears on the right-hand side of equation (14) $\rightarrow$ four possible values of θ_2.
- Two possible sets of θ_4, θ_5 and θ_6 from inverse Euler angle algorithm.
- Overall **eight** possible sets of joint angles θ_i, $i = 1, .., 6$ for a given ${}^0_6[T]$.

- From equation (11) two sets of θ_1
- From equation (13) two sets of θ_3
- Since θ_3 appears on the right-hand side of equation (14) $\rightarrow$ four possible values of θ_2.
- Two possible sets of θ_4, θ_5 and θ_6 from inverse Euler angle algorithm.
- Overall **eight** possible sets of joint angles θ_i, $i = 1, .., 6$ for a given ${}^0_6[T]$.

- From equation (11) two sets of θ_1
- From equation (13) two sets of θ_3
- Since θ_3 appears on the right-hand side of equation (14) $\rightarrow$ four possible values of θ_2.
- Two possible sets of θ_4, θ_5 and θ_6 from inverse Euler angle algorithm.
- Overall **eight** possible sets of joint angles $\theta_i,\ i = 1,..,6$ for a given ${}^0_6[T]$.

- Usual definition: All ${}^0_6[T]$ (position and orientation of $\{6\}$) such that inverse kinematics solution exists.
- Six dimensional entity – difficult to imagine or describe!
- Possible to derive the 'position' workspace of 'wrist' point. Position vector of wrist point

$$\begin{aligned} x &= c_1(a_2c_2 + a_3c_{23} - d_4s_{23}) - d_3s_1 \\ y &= s_1(a_2c_2 + a_3c_{23} - d_4s_{23}) + d_3c_1 \\ z &= -a_2s_2 - a_3s_{23} - d_4c_{23} \end{aligned} \tag{17}$$

- (x, y, z) are functions of three independent variables θ_1, θ_2 and θ_3 $\Rightarrow$ represents a **solid** in 3D space.
- Can obtain equation of bounding surfaces.

- Usual definition: All ${}^0_6[T]$ (position and orientation of $\{6\}$) such that inverse kinematics solution exists.
- Six dimensional entity – difficult to imagine or describe!
- Possible to derive the 'position' workspace of 'wrist' point. Position vector of wrist point

$$
\begin{aligned}
x &= c_1(a_2c_2 + a_3c_{23} - d_4s_{23}) - d_3s_1 \\
y &= s_1(a_2c_2 + a_3c_{23} - d_4s_{23}) + d_3c_1 \\
z &= -a_2s_2 - a_3s_{23} - d_4c_{23}
\end{aligned} \tag{17}
$$

- (x, y, z) are functions of three independent variables θ_1, θ_2 and θ_3 $\Rightarrow$ represents a **solid** in 3D space.
- Can obtain equation of bounding surfaces.

- Usual definition: All ${}^0_6[T]$ (position and orientation of $\{6\}$) such that inverse kinematics solution exists.
- Six dimensional entity – difficult to imagine or describe!
- Possible to derive the 'position' workspace of 'wrist' point. Position vector of wrist point

$$\begin{aligned} x &= c_1(a_2c_2 + a_3c_{23} - d_4s_{23}) - d_3s_1 \\ y &= s_1(a_2c_2 + a_3c_{23} - d_4s_{23}) + d_3c_1 \\ z &= -a_2s_2 - a_3s_{23} - d_4c_{23} \end{aligned} \tag{17}$$

- (x, y, z) are functions of three independent variables θ_1, θ_2 and $\theta_3 \Rightarrow$ represents a **solid** in 3D space.
- Can obtain equation of bounding surfaces.

- Usual definition: All ${}^0_6[T]$ (position and orientation of $\{6\}$) such that inverse kinematics solution exists.
- Six dimensional entity – difficult to imagine or describe!
- Possible to derive the 'position' workspace of 'wrist' point. Position vector of wrist point

$$
\begin{aligned}
x &= c_1(a_2c_2 + a_3c_{23} - d_4s_{23}) - d_3s_1 \\
y &= s_1(a_2c_2 + a_3c_{23} - d_4s_{23}) + d_3c_1 \\
z &= -a_2s_2 - a_3s_{23} - d_4c_{23}
\end{aligned}
\tag{17}
$$

- (x, y, z) are functions of three independent variables θ_1, θ_2 and θ_3 $\Rightarrow$ represents a **solid** in 3D space.
- Can obtain equation of bounding surfaces.

- Squaring and adding the three equations in equation (17) gives

$$R^2 = x^2 + y^2 + z^2 = K_1 + K_2 c_3 - K_3 s_3$$

where K_1, K_2, and K_3 are constants.

- The envelope of this family of surfaces must satisfy

$$\frac{\partial R^2}{\partial \theta_3} = 0$$

which gives

$$K_2 s_3 + K_3 c_3 = 0$$

- Eliminating θ_3 and denoting $a_3^2 + d_4^2$ by l^2, gives

$$[x^2 + y^2 + z^2 - ((a_2 + l)^2 + d_3^2)][x^2 + y^2 + z^2 - ((a_2 - l)^2 + d_3^2)] = 0 \quad (18)$$

which implies that the bounding surfaces are **spheres**.

- At every point in the solid all possible orientations, *except* two special 'singular' configurations when $r_{23} = \pm 1$, are possible.

- Squaring and adding the three equations in equation (17) gives

$$R^2 = x^2 + y^2 + z^2 = K_1 + K_2 c_3 - K_3 s_3$$

where K_1, K_2, and K_3 are constants.

- The envelope of this family of surfaces must satisfy

$$\frac{\partial R^2}{\partial \theta_3} = 0$$

which gives

$$K_2 s_3 + K_3 c_3 = 0$$

- Eliminating θ_3 and denoting $a_3^2 + d_4^2$ by l^2, gives

$$[x^2 + y^2 + z^2 - ((a_2 + l)^2 + d_3^2)][x^2 + y^2 + z^2 - ((a_2 - l)^2 + d_3^2)] = 0 \quad (18)$$

which implies that the bounding surfaces are **spheres**.

- At every point in the solid all possible orientations, *except* two special 'singular' configurations when $r_{23} = \pm 1$, are possible.

PUMA 560 MANIPULATOR

WORKSPACE OF PUMA 560 MANIPULATOR (CONTD.)

- Squaring and adding the three equations in equation (17) gives

$$R^2 = x^2 + y^2 + z^2 = K_1 + K_2 c_3 - K_3 s_3$$

where K_1, K_2, and K_3 are constants.

- The envelope of this family of surfaces must satisfy

$$\frac{\partial R^2}{\partial \theta_3} = 0$$

which gives

$$K_2 s_3 + K_3 c_3 = 0$$

- Eliminating θ_3 and denoting $a_3^2 + d_4^2$ by l^2, gives

$$[x^2 + y^2 + z^2 - ((a_2 + l)^2 + d_3^2)][x^2 + y^2 + z^2 - ((a_2 - l)^2 + d_3^2)] = 0 \quad (18)$$

which implies that the bounding surfaces are **spheres**.

- At every point in the solid all possible orientations, *except* two special 'singular' configurations when $r_{23} = \pm 1$, are possible.

PUMA 560 Manipulator

Workspace of PUMA 560 manipulator (contd.)

- Squaring and adding the three equations in equation (17) gives

$$R^2 = x^2 + y^2 + z^2 = K_1 + K_2 c_3 - K_3 s_3$$

  where K_1, K_2, and K_3 are constants.
- The envelope of this family of surfaces must satisfy

$$\frac{\partial R^2}{\partial \theta_3} = 0$$

  which gives

$$K_2 s_3 + K_3 c_3 = 0$$

- Eliminating θ_3 and denoting $a_3^2 + d_4^2$ by l^2, gives

$$[x^2 + y^2 + z^2 - ((a_2 + l)^2 + d_3^2)][x^2 + y^2 + z^2 - ((a_2 - l)^2 + d_3^2)] = 0 \tag{18}$$

  which implies that the bounding surfaces are **spheres**.
- At every point in the solid all possible orientations, *except* two special 'singular' configurations when $r_{23} = \pm 1$, are possible.

Numerical example of a PUMA 560 manipulator

For the PUMA 560, the Denavit-Hartenberg parameters are

i	α_{i-1}	a_{i-1}	d_i	θ_i
	degrees	m	m	degrees
1	0	0	0	45
2	-90	0	0	60
3	0	0.4318	0.125	135
4	-90	0.019	0.432	30
5	90	0	0	-45
6	-90	0	0	120

For the above D-H table

$$
{}^0_6[T] = \begin{bmatrix} 0.9749 & -0.2192 & -0.0388 & 0.1304 \\ 0.1643 & 0.8262 & -0.5388 & 0.3071 \\ 0.1502 & 0.5190 & 0.8415 & 0.0482 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

Numerical example of a PUMA 560 manipulator

For the above ${}^0_6[T]$, the inverse kinematics solutions are

i	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6
1	-91	120	50.04	177.51	-42.65	105.34
2	-91	120	50.04	-2.49	42.65	-74.66
3	45	-77.73	50.04	85.25	-159.22	-132.87
4	45	-77.73	50.04	-94.75	159.22	47.13
5	-91	-102.27	135	92.28	-178.31	15.79
6	-91	-102.27	135	-87.72	178.31	-164.21
7	45	60	135	30	-45	120
8	45	60	135	210	45	300

Note: As expected, one of the solutions (set 7) matches the chosen values of θ_i, $i = 1, \ldots, 6$, in the direct kinematics.

PUMA 560 Manipulator

Inverse kinematics & Workspace of PUMA 560 manipulator

Workspace of the wrist point of the PUMA shown in figure 7.

Note: *Actual* workspace is subset of *shown* workspace due to *joint rotation limits*.

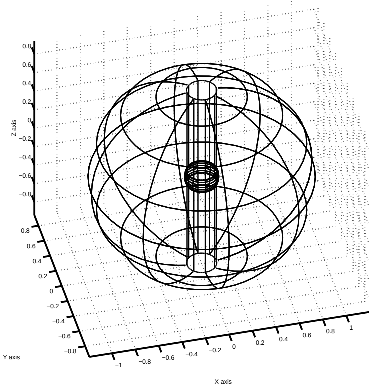


Figure 7: Workspace of the wrist point of the PUMA

Review of IK

- Transcendental equations → polynomial equations using tangent half angle substitution.
- Polynomial equation of higher degree – linear in $\sin(\theta)$ or $\cos(\theta)$ → quadratic in x^2 with $x = \tan(\frac{\theta}{2})$.
- For analytical solutions to IK → *eliminate* joint variable(s) from set of non-linear equations in several joint variables → a single equation in *one* joint variable.
  - Planar 3R example – three equations in three joint variables → two equations in θ_1 and θ_2 → one equation, equation (5), in θ_2 alone.
  - Single equation solved for θ_2 and then solve for θ_1 and θ_3.
  - For PUMA 560 – 3 equations in first 3 joint variables from the position of the wrist point.
  - Solve for the first 3 joint joint variables and then for last 3 joint variables using orientation information.
  - *Decoupling* of the position and orientation was first noticed by Pieper (Pieper, 1968) for manipulators with intersecting wrist. Later generalised to *any* six- degree-of-freedom serial manipulator where three consecutive joint axes intersect.

Review of IK

- Transcendental equations $\rightarrow$ polynomial equations using tangent half angle substitution.
- Polynomial equation of higher degree – linear in $\sin(\theta)$ or $\cos(\theta)$ $\rightarrow$ quadratic in x^2 with $x = \tan(\frac{\theta}{2})$.
- For analytical solutions to IK $\rightarrow$ *eliminate* joint variable(s) from set of non-linear equations in several joint variables $\rightarrow$ a single equation in *one* joint variable.
  - Planar 3R example – three equations in three joint variables $\rightarrow$ two equations in θ_1 and θ_2 $\rightarrow$ one equation, equation (5), in θ_2 alone.
  - Single equation solved for θ_2 and then solve for θ_1 and θ_3.
  - For PUMA 560 – 3 equations in first 3 joint variables from the position of the wrist point.
  - Solve for the first 3 joint joint variables and then for last 3 joint variables using orientation information.
  - *Decoupling* of the position and orientation was first noticed by Pieper (Pieper, 1968) for manipulators with intersecting wrist. Later generalised to *any* six- degree-of-freedom serial manipulator where three consecutive joint axes intersect.

Review of IK

- Transcendental equations $\rightarrow$ polynomial equations using tangent half angle substitution.
- Polynomial equation of higher degree – linear in $\sin(\theta)$ or $\cos(\theta)$ $\rightarrow$ quadratic in x^2 with $x = \tan(\frac{\theta}{2})$.
- For analytical solutions to IK $\rightarrow$ *eliminate* joint variable(s) from set of non-linear equations in several joint variables $\rightarrow$ a single equation in *one* joint variable.
  - Planar 3R example – three equations in three joint variables $\rightarrow$ two equations in θ_1 and θ_2 $\rightarrow$ one equation, equation (5), in θ_2 alone.
  - Single equation solved for θ_2 and then solve for θ_1 and θ_3.
  - For PUMA 560 – 3 equations in first 3 joint variables from the position of the wrist point.
  - Solve for the first 3 joint joint variables and then for last 3 joint variables using orientation information.
  - *Decoupling* of the position and orientation was first noticed by Pieper (Pieper, 1968) for manipulators with intersecting wrist. Later generalised to *any* six- degree-of-freedom serial manipulator where three consecutive joint axes intersect.

REVIEW OF IK (CONTD.)

- For all six- degree-of-freedom serial manipulators, with three joint axes intersecting $\rightarrow$ at most a fourth-order polynomial in the tangent of a joint angle need to be solved.
- The manipulator wrist point can reach any position in the workspace in *at most four* possible ways.
- Fourth-degree polynomials can be solved in closed-form (Korn and Korn, 1968) $\rightarrow$ IK of all six- degree-of-freedom serial manipulators with three intersecting axes can be solved in closed-form.
- For PUMA 560, the workspace of the wrist point is bounded by two spheres and require solution of *only* a quadratic due to special geometry.
- In general geometry robot with intersecting wrist, boundaries of the solid region traced by the wrist point form a torus which is a fourth-degree surface (Tsai and Soni, 1984).

Review of IK (Contd.)

- For all six- degree-of-freedom serial manipulators, with three joint axes intersecting $\rightarrow$ at most a fourth-order polynomial in the tangent of a joint angle need to be solved.
- The manipulator wrist point can reach any position in the workspace in *at most four* possible ways.
- Fourth-degree polynomials can be solved in closed-form (Korn and Korn, 1968) $\rightarrow$ IK of all six- degree-of-freedom serial manipulators with three intersecting axes can be solved in closed-form.
- For PUMA 560, the workspace of the wrist point is bounded by two spheres and require solution of *only* a quadratic due to special geometry.
- In general geometry robot with intersecting wrist, boundaries of the solid region traced by the wrist point form a torus which is a fourth-degree surface (Tsai and Soni, 1984).

Review of IK (contd.)

- For all six- degree-of-freedom serial manipulators, with three joint axes intersecting $\rightarrow$ at most a fourth-order polynomial in the tangent of a joint angle need to be solved.
- The manipulator wrist point can reach any position in the workspace in *at most four* possible ways.
- Fourth-degree polynomials can be solved in closed-form (Korn and Korn, 1968) $\rightarrow$ IK of all six- degree-of-freedom serial manipulators with three intersecting axes can be solved in closed-form.
- For PUMA 560, the workspace of the wrist point is bounded by two spheres and require solution of *only* a quadratic due to special geometry.
- In general geometry robot with intersecting wrist, boundaries of the solid region traced by the wrist point form a torus which is a fourth-degree surface (Tsai and Soni, 1984).

Review of IK (Contd.)

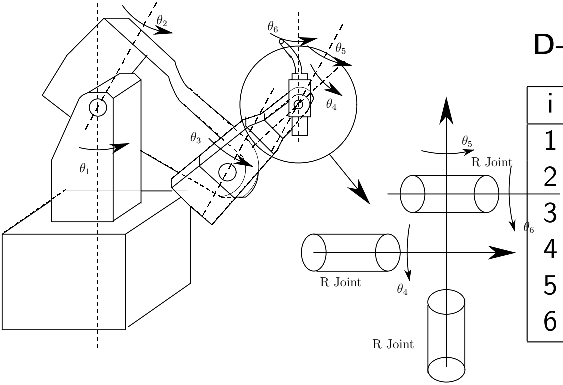
- For all six- degree-of-freedom serial manipulators, with three joint axes intersecting $\rightarrow$ at most a fourth-order polynomial in the tangent of a joint angle need to be solved.
- The manipulator wrist point can reach any position in the workspace in *at most four* possible ways.
- Fourth-degree polynomials can be solved in closed-form (Korn and Korn, 1968) $\rightarrow$ IK of all six- degree-of-freedom serial manipulators with three intersecting axes can be solved in closed-form.
- For PUMA 560, the workspace of the wrist point is bounded by two spheres and require solution of *only* a quadratic due to special geometry.
- In general geometry robot with intersecting wrist, boundaries of the solid region traced by the wrist point form a torus which is a fourth-degree surface (Tsai and Soni, 1984).

Review of IK (Contd.)

- For all six- degree-of-freedom serial manipulators, with three joint axes intersecting $\rightarrow$ at most a fourth-order polynomial in the tangent of a joint angle need to be solved.
- The manipulator wrist point can reach any position in the workspace in *at most four* possible ways.
- Fourth-degree polynomials can be solved in closed-form (Korn and Korn, 1968) $\rightarrow$ IK of all six- degree-of-freedom serial manipulators with three intersecting axes can be solved in closed-form.
- For PUMA 560, the workspace of the wrist point is bounded by two spheres and require solution of *only* a quadratic due to special geometry.
- In general geometry robot with intersecting wrist, boundaries of the solid region traced by the wrist point form a torus which is a fourth-degree surface (Tsai and Soni, 1984).

Non-intersecting wrists

- Difficult to design and manufacture three intersecting axis wrist.
- Much easier if wrist has *two* intersecting axis.



D-H parameters

i	α_{i-1}	a_{i-1}	d_i	θ_i
1	0	0	0	θ_1
2	$-\pi/2$	0	0	θ_2
3	0	a_2	d_3	θ_3
4	$-\pi/2$	a_3	d_4	θ_4
5	$\pi/2$	0	d_5	θ_5
6	$-\pi/2$	0	0	θ_6

Figure 8: A robot with non-intersecting wrist

Non-intersecting wrists

- Figure 8 shows a six- degree-of-freedom robot – First 3 joint axis are similar to PUMA 560.
- Last three axes do not intersect and there is an offset d_5.
- From D-H table compute ${}^0_1[T], \ldots, {}^5_6[T]$ and then ${}^0_6[T]$.
- Last column of ${}^0_6[T]$ is

$$
\begin{aligned}
x &= c_1(a_2c_2 + a_3c_{23} - d_4s_{23}) - d_3s_1 + d_5(s_1c_4 - c_1s_4c_{23}) \\
y &= s_1(a_2c_2 + a_3c_{23} - d_4s_{23}) + d_3c_1 - d_5(c_1c_4 + s_1s_4c_{23}) \\
z &= -a_2s_2 - a_3s_{23} - d_4c_{23} - d_5s_4s_{23}
\end{aligned}
\qquad (19)
$$

Note: (x, y, z) is a function of θ_1, θ_2, θ_3 *and* θ_4.

- Need one more equation in the four joint variables!

Non-intersecting wrists

- From ${}^3_6[R] = {}^0_3[R]^T \, {}^0_6[R]$,

$$\begin{pmatrix} c_4c_5c_6 - s_4s_6 & -c_4c_5s_6 - s_4c_6 & -c_4s_5 \\ s_5c_6 & -s_5s_6 & c_5 \\ -s_4c_5c_6 - c_4s_6 & s_4c_5s_6 - c_4c_6 & s_4s_5 \end{pmatrix} =$$

$$\begin{pmatrix} c_1c_{23} & s_1c_{23} & s_{23} \\ -c_1s_{23} & -s_1s_{23} & -c_{23} \\ -s_1 & c_1 & 0 \end{pmatrix} \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} \tag{20}$$

- Divide the (1,3) and the (3,3) terms of the above matrix equation, to get (for $\theta_5 \neq 0, \pi$),

$$s_4(r_{13}c_1c_{23} + r_{23}s_1c_{23} + r_{33}s_{23}) = c_4(r_{13}s_1 - r_{23}c_1) \tag{21}$$

- Equation (21) is the fourth equation!
- Solve numerically equations (19) and (21) to obtain θ_i, $i = 1,2,3,4$.
- Solve for θ_4, θ_5 and θ_6 using $Z-(-Y)-Z$ inverse Euler angle algorithm (similar to the PUMA 560 example).

- Assume numerical values of the D-H parameters same as a PUMA 560.
- Offset d_5 is chosen to be 20 mm.
- ${}^0_6[T]$ same as used for the PUMA 560 example

$$
{}^0_6[T] = \begin{bmatrix} 0.9749 & -0.2192 & -0.0388 & 0.1304 \\ 0.1643 & 0.8262 & -0.5388 & 0.3071 \\ 0.1502 & 0.5190 & 0.8415 & 0.0482 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

- Solve 4 non-linear equations numerically – *fsolve* in Matlab used here.

$$
\theta_1 = 41.82, \;\; \theta_2 = 60.43, \theta_3 = 135.33, \theta_4 = 31.96
$$

- Using inverse Euler angle algorithm – 2 sets of values $\theta_4 = 31.96, -148.04$, $\theta_5 = -45.22, +45.22$, and $\theta_6 = 121.57, -58.43$.
- As a check – one value of θ_4 matches.
- $\theta_5 = 0$ or π is a *singular* configuration for the non-intersecting wrist and only $\theta_4 \pm \theta_6$ can be found.

Outline

- ${}^{0}_{n}[T]$ define position and orientation of $\{n\}$ with respect to $\{0\}$.
- ${}^{0}_{n}[T]$, in general, provide *up to* 6 (for 3D) and 3 (for planar) task space pieces of information. Note: n is the number of unknown joint variables.
- If $n < 6$ for 3D motion or $n < 3$ for planar motion $\rightarrow$ there exists $6 - n$ ($3 - n$ for planar) functional relationships involving the task space variables – *constrained* manipulators.
- Functional relationships obtained by *inspection* of geometry or by using *theory of elimination* (see Lecture 4).
- Start with a simple example of $n < 6$.

Introduction

Review

- ${}^0_n[T]$ define position and orientation of $\{n\}$ with respect to $\{0\}$.
- ${}^0_n[T]$, in general, provide *up to* 6 (for 3D) and 3 (for planar) task space pieces of information. Note: n is the number of unknown joint variables.
- If $n < 6$ for 3D motion or $n < 3$ for planar motion $\rightarrow$ there exists $6 - n$ ($3 - n$ for planar) functional relationships involving the task space variables – *constrained* manipulators.
- Functional relationships obtained by *inspection* of geometry or by using *theory of elimination* (see Lecture 4).
- Start with a simple example of $n < 6$.

- ${}^{0}_{n}[T]$ define position and orientation of $\{n\}$ with respect to $\{0\}$.
- ${}^{0}_{n}[T]$, in general, provide *up to* 6 (for 3D) and 3 (for planar) task space pieces of information. Note: n is the number of unknown joint variables.
- If $n < 6$ for 3D motion or $n < 3$ for planar motion $\rightarrow$ there exists $6 - n$ ($3 - n$ for planar) functional relationships involving the task space variables – *constrained* manipulators.
- Functional relationships obtained by *inspection* of geometry or by using *theory of elimination* (see Lecture 4).
- Start with a simple example of $n < 6$.

- ${}^{0}_{n}[T]$ define position and orientation of $\{n\}$ with respect to $\{0\}$.
- ${}^{0}_{n}[T]$, in general, provide *up to* 6 (for 3D) and 3 (for planar) task space pieces of information. Note: n is the number of unknown joint variables.
- If $n < 6$ for 3D motion or $n < 3$ for planar motion $\rightarrow$ there exists $6 - n$ ($3 - n$ for planar) functional relationships involving the task space variables – *constrained* manipulators.
- Functional relationships obtained by *inspection* of geometry or by using *theory of elimination* (see Lecture 4).
- Start with a simple example of $n < 6$.

- ${}^{0}_{n}[T]$ define position and orientation of $\{n\}$ with respect to $\{0\}$.
- ${}^{0}_{n}[T]$, in general, provide *up to* 6 (for 3D) and 3 (for planar) task space pieces of information. Note: n is the number of unknown joint variables.
- If $n < 6$ for 3D motion or $n < 3$ for planar motion $\rightarrow$ there exists $6 - n$ ($3 - n$ for planar) functional relationships involving the task space variables – *constrained* manipulators.
- Functional relationships obtained by *inspection* of geometry or by using *theory of elimination* (see Lecture 4).
- Start with a simple example of $n < 6$.

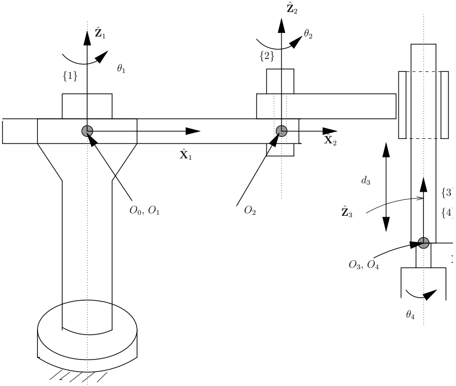


Figure 9: A SCARA manipulator

- ${}^0_4[T]$ can give position and orientation $(x, y, z; [R])$ of $\{4\}$.
- Due to geometry and seen from figure *only* angle ϕ represents orientation of $\{4\}$ – other two Euler angles are zero!!.
- Hence *only* the position (x, y, z) and the angle ϕ of $\{4\}$ is relevant – equal number of equations and unknowns.

$$\begin{aligned} x &= a_1 c_1 + a_2 c_{12} \\ y &= a_1 s_1 + a_2 s_{12} \\ z &= -d_3 \\ \phi &= \theta_1 + \theta_2 + \theta_4 \end{aligned} \tag{22}$$

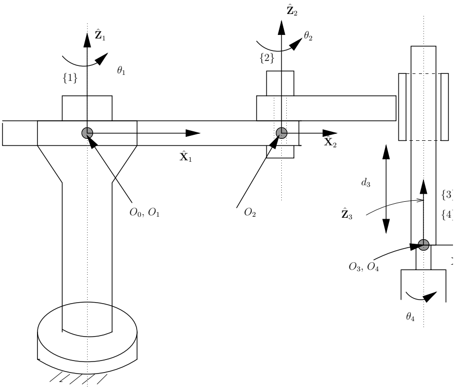


Figure 9: A SCARA manipulator

- ${}^0_4[T]$ can give position and orientation $(x, y, z; [R])$ of $\{4\}$.
- Due to geometry and seen from figure *only* angle ϕ represents orientation of $\{4\}$ – other two Euler angles are zero!!.
- Hence *only* the position (x, y, z) and the angle ϕ of $\{4\}$ is relevant – equal number of equations and unknowns.

$$
\begin{aligned}
x &= a_1 c_1 + a_2 c_{12} \\
y &= a_1 s_1 + a_2 s_{12} \\
z &= -d_3 \\
\phi &= \theta_1 + \theta_2 + \theta_4
\end{aligned}
\tag{22}
$$

Inverse kinematics for $n < 6$

SCARA Robot

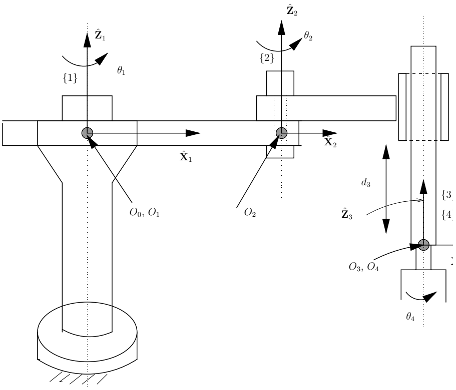


Figure 9: A SCARA manipulator

- ${}^0_4[T]$ can give position and orientation $(x, y, z; [R])$ of $\{4\}$.
- Due to geometry and seen from figure *only* angle ϕ represents orientation of $\{4\}$ – other two Euler angles are zero!!.
- Hence *only* the position (x, y, z) and the angle ϕ of $\{4\}$ is relevant – equal number of equations and unknowns.

$$\begin{aligned} x &= a_1 c_1 + a_2 c_{12} \\ y &= a_1 s_1 + a_2 s_{12} \\ z &= -d_3 \\ \phi &= \theta_1 + \theta_2 + \theta_4 \end{aligned} \tag{22}$$

Inverse kinematics for $n < 6$

SCARA Robot (contd.)

- Inverse kinematics solutions of SCARA robot can be obtained from equation (22).
- The unknown joint variables are:

$$\begin{aligned} \theta_2 &= \pm \cos^{-1}\left(\frac{x^2+y^2-l_1^2-l_2^2}{2 l_1 l_2}\right) \\ \theta_1 &= \text{Atan2}(y,x) - \text{Atan2}(l_2 s_2, l_1 + l_2 c_2) \\ d_3 &= -z \\ \theta_4 &= \phi - \theta_1 - \theta_2 \end{aligned} \tag{23}$$

- Two possible sets of joint variables for a give (x, y, z, ϕ).
- Workspace: All reachable points (x, y, z) lie in an annular cylinder of inner and outer radii given by $l_1 - l_2$ and $l_1 + l_2$ $(l_1 > l_2)$ respectively.

- Inverse kinematics solutions of SCARA robot can be obtained from equation (22).
- The unknown joint variables are:

$$\begin{aligned} \theta_2 &= \pm\cos^{-1}\left(\frac{x^2+y^2-l_1^2-l_2^2}{2l_1l_2}\right) \\ \theta_1 &= \text{Atan2}(y,x) - \text{Atan2}(l_2s_2, l_1+l_2c_2) \\ d_3 &= -z \\ \theta_4 &= \phi - \theta_1 - \theta_2 \end{aligned} \quad (23)$$

- Two possible sets of joint variables for a give (x, y, z, ϕ).
- Workspace: All reachable points (x, y, z) lie in an annular cylinder of inner and outer radii given by $l_1 - l_2$ and $l_1 + l_2$ $(l_1 > l_2)$ respectively.

- Inverse kinematics solutions of SCARA robot can be obtained from equation (22).
- The unknown joint variables are:

$$
\begin{aligned}
\theta_2 &= \pm\cos^{-1}\left(\frac{x^2+y^2-l_1^2-l_2^2}{2l_1l_2}\right) \\
\theta_1 &= \mathrm{Atan2}(y,x) - \mathrm{Atan2}(l_2s_2, l_1+l_2c_2) \\
d_3 &= -z \\
\theta_4 &= \phi - \theta_1 - \theta_2
\end{aligned}
\tag{23}
$$

- Two possible sets of joint variables for a give (x,y,z,ϕ).
- Workspace: All reachable points (x,y,z) lie in an annular cylinder of inner and outer radii given by $l_1 - l_2$ and $l_1 + l_2$ $(l_1 > l_2)$ respectively.

- Inverse kinematics solutions of SCARA robot can be obtained from equation (22).
- The unknown joint variables are:

$$\begin{aligned}
\theta_2 &= \pm\cos^{-1}\left(\frac{x^2+y^2-l_1^2-l_2^2}{2l_1l_2}\right) \\
\theta_1 &= \mathrm{Atan2}(y,x) - \mathrm{Atan2}(l_2s_2, l_1+l_2c_2) \\
d_3 &= -z \\
\theta_4 &= \phi - \theta_1 - \theta_2
\end{aligned} \tag{23}$$

- Two possible sets of joint variables for a give (x,y,z,ϕ).
- Workspace: All reachable points (x,y,z) lie in an annular cylinder of inner and outer radii given by $l_1 - l_2$ and $l_1 + l_2$ ($l_1 > l_2$) respectively.

OUTLINE

- If $n > 6$ for 3D motion or $n > 3$ for planar motion $\rightarrow$ more unknowns than equations and hence infinite number of solutions – *redundant* manipulators.
- A simple example – planar 3R robot *but* not interested in orientation of the last link.
- Direct kinematics equations are

$$\begin{aligned} x &= l_1 c_1 + l_2 c_{12} + l_3 c_{123} \\ y &= l_1 s_1 + l_2 s_{12} + l_3 s_{123} \end{aligned} \tag{24}$$

- Inverse kinematics: Given (x, y) find θ_1, θ_2 and θ_3.
- Two equations and 3 variables – ∞ number of θ_i, $i = 1, 2, 3$.

- If $n > 6$ for 3D motion or $n > 3$ for planar motion $\rightarrow$ more unknowns than equations and hence infinite number of solutions – *redundant* manipulators.
- A simple example – planar 3R robot *but* not interested in orientation of the last link.
- Direct kinematics equations are

$$\begin{aligned} x &= l_1 c_1 + l_2 c_{12} + l_3 c_{123} \\ y &= l_1 s_1 + l_2 s_{12} + l_3 s_{123} \end{aligned} \tag{24}$$

- Inverse kinematics: Given (x, y) find θ_1, θ_2 and θ_3.
- Two equations and 3 variables – ∞ number of θ_i, $i = 1, 2, 3$.

- If $n > 6$ for 3D motion or $n > 3$ for planar motion $\rightarrow$ more unknowns than equations and hence infinite number of solutions – *redundant* manipulators.
- A simple example – planar 3R robot *but* not interested in orientation of the last link.
- Direct kinematics equations are

$$\begin{aligned} x &= l_1 c_1 + l_2 c_{12} + l_3 c_{123} \\ y &= l_1 s_1 + l_2 s_{12} + l_3 s_{123} \end{aligned} \tag{24}$$

- Inverse kinematics: Given (x, y) find θ_1, θ_2 and θ_3.
- Two equations and 3 variables – ∞ number of θ_i, $i = 1, 2, 3$.

- If $n > 6$ for 3D motion or $n > 3$ for planar motion $\rightarrow$ more unknowns than equations and hence infinite number of solutions – *redundant* manipulators.
- A simple example – planar 3R robot *but* not interested in orientation of the last link.
- Direct kinematics equations are

$$\begin{aligned} x &= l_1 c_1 + l_2 c_{12} + l_3 c_{123} \\ y &= l_1 s_1 + l_2 s_{12} + l_3 s_{123} \end{aligned} \tag{24}$$

- Inverse kinematics: Given (x, y) find θ_1, θ_2 and θ_3.
- Two equations and 3 variables – ∞ number of θ_i, $i = 1, 2, 3$.

- Need to use additional equation so that unique $\theta_i,\ i = 1, 2, 3$ can be obtained.
- *Optimisation* of a function of joint variables (Nakamura, 1991).
  - Minimisation of joint rotations, velocities and acceleration.
  - Avoiding obstacles and singularities.
  - Minimisation of actuator torques.
- *Resolution of redundancy*: Obtaining additional useful and meaningful equation(s) or constraint(s) to obtain unique joint values.
- Two resolution schemes
  1. Minimise joint rotations – illustrated using the planar 3R example.
  2. Minimise Cartesian motion of links.

Redundant Manipulators

- Need to use additional equation so that unique θ_i, $i = 1, 2, 3$ can be obtained.
- *Optimisation* of a function of joint variables (Nakamura, 1991).
  - Minimisation of joint rotations, velocities and acceleration.
  - Avoiding obstacles and singularities.
  - Minimisation of actuator torques.
- *Resolution of redundancy*: Obtaining additional useful and meaningful equation(s) or constraint(s) to obtain unique joint values.
- Two resolution schemes
  1. Minimise joint rotations – illustrated using the planar 3R example.
  2. Minimise Cartesian motion of links.

Redundant Manipulators

- Need to use additional equation so that unique $\theta_i, \; i = 1,2,3$ can be obtained.
- *Optimisation* of a function of joint variables (Nakamura, 1991).
  - Minimisation of joint rotations, velocities and acceleration.
  - Avoiding obstacles and singularities.
  - Minimisation of actuator torques.
- *Resolution of redundancy*: Obtaining additional useful and meaningful equation(s) or constraint(s) to obtain unique joint values.
- Two resolution schemes
  1. Minimise joint rotations – illustrated using the planar 3R example.
  2. Minimise Cartesian motion of links.

Redundant Manipulators

- Need to use additional equation so that unique $\theta_i, \; i = 1,2,3$ can be obtained.
- *Optimisation* of a function of joint variables (Nakamura, 1991).
  - Minimisation of joint rotations, velocities and acceleration.
  - Avoiding obstacles and singularities.
  - Minimisation of actuator torques.
- *Resolution of redundancy*: Obtaining additional useful and meaningful equation(s) or constraint(s) to obtain unique joint values.
- Two resolution schemes
  1. Minimise joint rotations – illustrated using the planar 3R example.
  2. Minimise Cartesian motion of links.

Redundant Manipulators

Minimise joint rotations

- For planar 3R manipulator minimise joint rotation $\rightarrow$ minimise $\theta_1^2 + \theta_2^2 + \theta_3^2$.
- Optimisation problem: Minimize $f(\theta) = \theta_1^2 + \theta_2^2 + \theta_3^2$ subject to

$$
\begin{aligned}
g_1(\theta) &= -x + l_1 c_1 + l_2 c_{12} + l_3 c_{123} = 0 \\
g_2(\theta) &= -y + l_1 s_1 + l_2 s_{12} + l_3 s_{123} = 0
\end{aligned}
$$

  $\theta = (\theta_1, \theta_2, \theta_3)^T$, and (x, y) denote trajectory of end-effector.
- Solve using classical method of Lagrange multipliers.
  - Form the function

$$F(\theta) = f(\theta) - \lambda_1 g_1(\theta) - \lambda_2 g_2(\theta) \tag{25}$$

  - Equate the derivatives of $F(\theta)$ to zero

$$
\begin{aligned}
\frac{\partial f}{\partial \theta} &= \lambda_1 \frac{\partial g_1}{\partial \theta} + \lambda_2 \frac{\partial g_2}{\partial \theta} \\
g_1(\theta) = g_2(\theta) &= 0
\end{aligned} \tag{26}
$$

- For planar 3R manipulator minimise joint rotation $\rightarrow$ minimise $\theta_1^2 + \theta_2^2 + \theta_3^2$.
- Optimisation problem: Minimize $f(\theta) = \theta_1^2 + \theta_2^2 + \theta_3^2$ subject to

$$\begin{aligned} g_1(\theta) &= -x + l_1 c_1 + l_2 c_{12} + l_3 c_{123} = 0 \\ g_2(\theta) &= -y + l_1 s_1 + l_2 s_{12} + l_3 s_{123} = 0 \end{aligned}$$

$\theta = (\theta_1, \theta_2, \theta_3)^T$, and (x, y) denote trajectory of end-effector.
- Solve using classical method of Lagrange multipliers.
  - Form the function

$$F(\theta) = f(\theta) - \lambda_1 g_1(\theta) - \lambda_2 g_2(\theta) \tag{25}$$

  - Equate the derivatives of $F(\theta)$ to zero

$$\begin{aligned} \frac{\partial f}{\partial \theta} &= \lambda_1 \frac{\partial g_1}{\partial \theta} + \lambda_2 \frac{\partial g_2}{\partial \theta} \\ g_1(\theta) = g_2(\theta) &= 0 \end{aligned} \tag{26}$$

- For planar 3R manipulator minimise joint rotation $\rightarrow$ minimise $\theta_1^2 + \theta_2^2 + \theta_3^2$.
- Optimisation problem: Minimize $f(\theta) = \theta_1^2 + \theta_2^2 + \theta_3^2$ subject to

$$
\begin{aligned}
g_1(\theta) &= -x + l_1 c_1 + l_2 c_{12} + l_3 c_{123} = 0 \\
g_2(\theta) &= -y + l_1 s_1 + l_2 s_{12} + l_3 s_{123} = 0
\end{aligned}
$$

  $\theta = (\theta_1, \theta_2, \theta_3)^T$, and (x, y) denote trajectory of end-effector.
- Solve using classical method of Lagrange multipliers.
  - Form the function

$$
F(\theta) = f(\theta) - \lambda_1 g_1(\theta) - \lambda_2 g_2(\theta) \tag{25}
$$

  - Equate the derivatives of $F(\theta)$ to zero

$$
\begin{aligned}
\frac{\partial f}{\partial \theta} &= \lambda_1 \frac{\partial g_1}{\partial \theta} + \lambda_2 \frac{\partial g_2}{\partial \theta} \\
g_1(\theta) = g_2(\theta) &= 0
\end{aligned} \tag{26}
$$

- Eliminate λ_1 and λ_2 by rewriting the first equation as

$$\begin{pmatrix} \frac{\partial f}{\partial \theta_1} & \frac{\partial g_1}{\partial \theta_1} & \frac{\partial g_2}{\partial \theta_1} \\ \frac{\partial f}{\partial \theta_2} & \frac{\partial g_1}{\partial \theta_2} & \frac{\partial g_2}{\partial \theta_2} \\ \frac{\partial f}{\partial \theta_3} & \frac{\partial g_1}{\partial \theta_3} & \frac{\partial g_2}{\partial \theta_3} \end{pmatrix} \begin{pmatrix} 1 \\ -\lambda_1 \\ -\lambda_2 \end{pmatrix} = 0 \qquad (27)$$

- For non-trivial λ_1 and λ_2 $\rightarrow$ equate determinant of the 3×3 matrix as zero.

$$l_1 l_2 \theta_3 s_2 + l_2 l_3 (\theta_1 - \theta_2) s_3 + l_3 l_1 (\theta_3 - \theta_2) s_{23} = 0 \qquad (28)$$

- Solve equation (28) together with $g_1(\theta) = 0$ and $g_2(\theta) = 0$ numerically.
- Figure 10 shows the plot of θ_1, θ_2, θ_3, and $f(\theta)$

- Eliminate λ_1 and λ_2 by rewriting the first equation as

$$\begin{pmatrix} \frac{\partial f}{\partial \theta_1} & \frac{\partial g_1}{\partial \theta_1} & \frac{\partial g_2}{\partial \theta_1} \\ \frac{\partial f}{\partial \theta_2} & \frac{\partial g_1}{\partial \theta_2} & \frac{\partial g_2}{\partial \theta_2} \\ \frac{\partial f}{\partial \theta_3} & \frac{\partial g_1}{\partial \theta_3} & \frac{\partial g_2}{\partial \theta_3} \end{pmatrix} \begin{pmatrix} 1 \\ -\lambda_1 \\ -\lambda_2 \end{pmatrix} = 0 \tag{27}$$

- For non-trivial λ_1 and λ_2 $\rightarrow$ equate determinant of the 3×3 matrix as zero.

$$l_1 l_2 \theta_3 s_2 + l_2 l_3 (\theta_1 - \theta_2) s_3 + l_3 l_1 (\theta_3 - \theta_2) s_{23} = 0 \tag{28}$$

- Solve equation (28) together with $g_1(\theta) = 0$ and $g_2(\theta) = 0$ numerically.
- Figure 10 shows the plot of θ_1, θ_2, θ_3, and $f(\theta)$

Redundant Manipulators

Minimise joint rotations

- l_1, l_2, and l_3 are chosen to be 5, 3, and 1, respectively.
- The end-effector traces a straight line parallel to the Y axis.

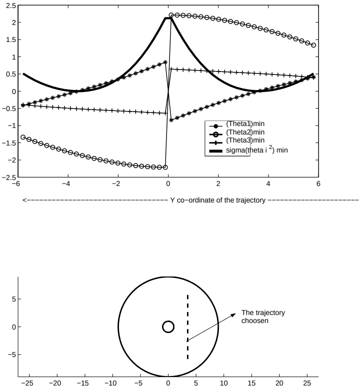


Figure 10: Plot of joint variables for redundant planar 3R robot

Redundant Manipulators

Minimise joint rotations with joint constraints

- Solve same optimisation problem with $-120^\circ \leq \theta_2 \leq 120^\circ$.
- *All* joint variables different when θ_2 is constrained.

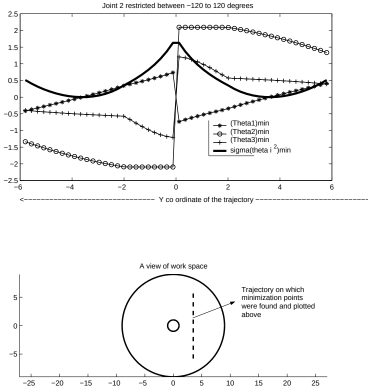


Figure 11: Plot of joint variables for redundant planar 3R robot with joint limit

- Classical *tractrix* curve – called hund or hound curve by Leibniz
- A link moves such that the head P moves along the X axis and the velocity of tail j_0 is *along* the link.
- The curve traced by the tail is the *tractrix*.

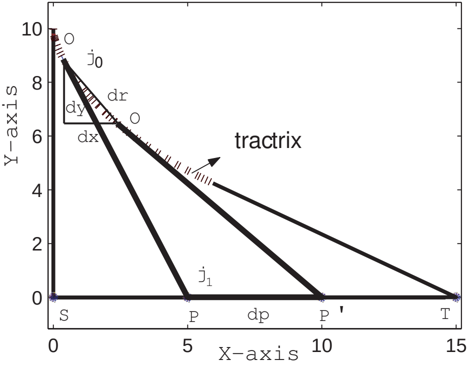


Figure 12: Motion of a link when one end is pulled parallel to X axis

See link for more details on *tractrix* curve.

- Classical *tractrix* curve – called hund or hound curve by Leibniz
- A link moves such that the head P moves along the X axis and the velocity of tail j_0 is *along* the link.
- The curve traced by the tail is the *tractrix*.

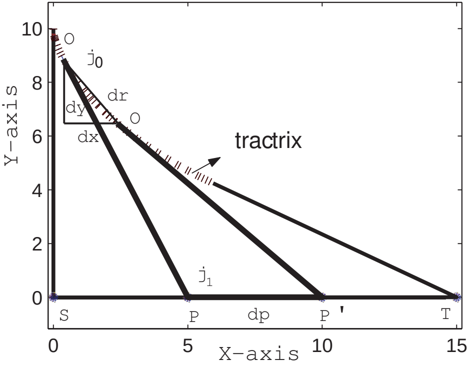


Figure 12: Motion of a link when one end is pulled parallel to X axis

See link for more details on *tractrix* curve.

- Classical *tractrix* curve – called hund or hound curve by Leibniz
- A link moves such that the head P moves along the X axis and the velocity of tail j_0 is *along* the link.
- The curve traced by the tail is the *tractrix*.

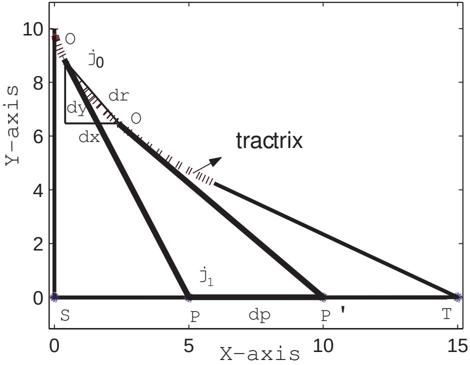


Figure 12: Motion of a link when one end is pulled parallel to X axis

See link for more details on *tractrix* curve.

- Since the velocity vector at j_0 is always aligned with the link, the tractrix equation is

$$\frac{dy}{dx} = \frac{-y}{\sqrt{L^2 - y^2}} \tag{29}$$

where L is length of the link.

- Solution in *closed form* and *parametric form*

$$\begin{aligned} x &= L \log \frac{y}{L - \sqrt{L^2 - y^2}} - \sqrt{L^2 - y^2} \\ x(p) &= p - L \tanh(\frac{p}{L}), \quad y(p) = L \operatorname{sech}(\frac{p}{L}) \end{aligned} \tag{30}$$

- Some key properties of the tractrix curve
  - For an *infinitesimal* motion of head dp, the length of path traversed by tail dr is *minimum* of all possible paths.
  - $dr \leq dp$ and equal when velocity of head is along link.

REDUNDANT MANIPULATORS

TRACTRIX EQUATION

- Since the velocity vector at j_0 is always aligned with the link, the tractrix equation is

$$\frac{dy}{dx} = \frac{-y}{\sqrt{L^2 - y^2}} \tag{29}$$

where L is length of the link.

- Solution in *closed form* and *parametric form*

$$\begin{aligned} x &= L \log \frac{y}{L - \sqrt{L^2 - y^2}} - \sqrt{L^2 - y^2} \\ x(p) &= p - L \tanh(\frac{p}{L}), \quad y(p) = L \operatorname{sech}(\frac{p}{L}) \end{aligned} \tag{30}$$

- Some key properties of the tractrix curve
  - For an *infinitesimal* motion of head dp, the length of path traversed by tail dr is *minimum* of all possible paths.
  - $dr \leq dp$ and equal when velocity of head is along link.

- Since the velocity vector at j_0 is always aligned with the link, the tractrix equation is

$$\frac{dy}{dx} = \frac{-y}{\sqrt{L^2 - y^2}} \tag{29}$$

  where L is length of the link.
- Solution in *closed form* and *parametric form*

$$\begin{aligned} x &= L \log \frac{y}{L - \sqrt{L^2 - y^2}} - \sqrt{L^2 - y^2} \\ x(p) &= p - L \tanh(\frac{p}{L}), \quad y(p) = L \operatorname{sech}(\frac{p}{L}) \end{aligned} \tag{30}$$

- Some key properties of the tractrix curve
  - For an *infinitesimal* motion of head dp, the length of path traversed by tail dr is *minimum* of all possible paths.
  - $dr \leq dp$ and equal when velocity of head is along link.

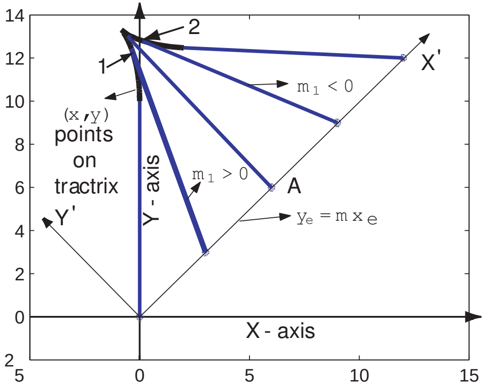


Figure 13: Motion of a link when one end is pulled along a line $y_e = mx_e$

- Head moving along $y_e = mx_e$, $m = y_p/x_p$ is slope and (x_p, y_p) is the destination point.
- Modified differential equation of tractrix – $\frac{dy}{dx} = \frac{y-y_e}{x-x_e}$
- Solution is $x_e = \frac{-B \pm \sqrt{B^2-4AC}}{2A}$ where $A = 1+m^2$, $B = 2my+2x$, $C = x^2+y^2-L^2$.
- Use $+$ when slope of the link (m_1) in $X' - Y'$ is negative and vice versa.
- Initially as head moves along X', the tail moves backward!

Algorithm *TRACTRIX3D*

1. Define $\mathbf{S} = \mathbf{X}_p - \mathbf{X}_h$ where $\mathbf{X}_h$ is the current location of the head.
2. Define $\mathbf{T} = \mathbf{X} - \mathbf{X}_h$ where $\mathbf{X} = (x, y, z)^T$ is the tail of the link lying on the tractrix.
3. Define reference coordinate system $\{r\}$ with the X-axis along $\mathbf{S}$.
4. Define the Z-axis as $\hat{\mathbf{Z}}_r = \frac{\mathbf{S}\times\mathbf{T}}{|\mathbf{S}\times\mathbf{T}|}$.
5. Define rotation matrix ${}^0_r[R]$ from X, Y and Z axis.
6. Obtain $y = \hat{\mathbf{Y}}_r \cdot \mathbf{T}$ and parameter p from $p = L\ \mathrm{sech}^{-1}(\frac{y}{L}) \pm |\mathbf{S}|$.
7. From p obtain (x_r, y_r) in $\{r\}$

$$x_r = \pm|\mathbf{S}| - L\ \tanh(\frac{p}{L}) \qquad y_r = L\ \mathrm{sech}(\frac{p}{L}) \tag{31}$$

8. Obtain $(x, y, z)^T$ in $\{0\}$ by $(x, y, z)^T = \mathbf{X}_h + {}^0_r[R](x_r, y_r, 0)^T$.

REDUNDANT MANIPULATORS

RESOLUTION OF REDUNDANCY USING TRACTRIX

- Consider a redundant manipulator with n links and joints $j_1, j_2, ..., j_{n-1}$ where j_i is the joint connecting link l_i and link l_{i+1} – joints are either spherical joints or rotary.
- Consider the last two links l_n and l_{n-1} – the head of the link l_n denoted by j_n is to be moved to $j_{n_{\mathrm{new}}}$ given by $\mathbf{X}_p = (x_p, y_p, z_p)^T$.
- Obtain new displaced location of tail j_{n-1} using algorithm *TRACTRIX3D* – denote by $\mathbf{X} = (x, y, z)^T$.
- Tail of the link l_n is the *head of the link* l_{n-1} – Desired location of head of the link l_{n-1} is $(x, y, z)^T$.
- Obtain location of the tail of link l_{n-1} using algorithm *TRACTRIX3D*.
- Recursively obtain the motion of the head and tail of all links down to the first link l_1.

- Consider a redundant manipulator with n links and joints $j_1, j_2, ..., j_{n-1}$ where j_i is the joint connecting link l_i and link l_{i+1} – joints are either spherical joints or rotary.
- Consider the last two links l_n and l_{n-1} – the head of the link l_n denoted by j_n is to be moved to $j_{n_{\text{new}}}$ given by $\mathbf{X}_p = (x_p, y_p, z_p)^T$.
- Obtain new displaced location of tail j_{n-1} using algorithm *TRACTRIX3D* – denote by $\mathbf{X} = (x, y, z)^T$.
- Tail of the link l_n is the *head of the link* l_{n-1} – Desired location of head of the link l_{n-1} is $(x, y, z)^T$.
- Obtain location of the tail of link l_{n-1} using algorithm *TRACTRIX3D*.
- Recursively obtain the motion of the head and tail of all links down to the first link l_1.

- Consider a redundant manipulator with n links and joints $j_1, j_2, ..., j_{n-1}$ where j_i is the joint connecting link l_i and link l_{i+1} – joints are either spherical joints or rotary.
- Consider the last two links l_n and l_{n-1} – the head of the link l_n denoted by j_n is to be moved to $j_{n_{\mathrm{new}}}$ given by $\mathbf{X}_p = (x_p, y_p, z_p)^T$.
- Obtain new displaced location of tail j_{n-1} using algorithm *TRACTRIX3D* – denote by $\mathbf{X} = (x, y, z)^T$.
- Tail of the link l_n is the *head of the link* l_{n-1} – Desired location of head of the link l_{n-1} is $(x, y, z)^T$.
- Obtain location of the tail of link l_{n-1} using algorithm *TRACTRIX3D*.
- Recursively obtain the motion of the head and tail of all links down to the first link l_1.

- Consider a redundant manipulator with n links and joints $j_1, j_2, ..., j_{n-1}$ where j_i is the joint connecting link l_i and link l_{i+1} – joints are either spherical joints or rotary.
- Consider the last two links l_n and l_{n-1} – the head of the link l_n denoted by j_n is to be moved to $j_{n_{\text{new}}}$ given by $\mathbf{X}_p = (x_p, y_p, z_p)^T$.
- Obtain new displaced location of tail j_{n-1} using algorithm *TRACTRIX3D* – denote by $\mathbf{X} = (x, y, z)^T$.
- Tail of the link l_n is the *head of the link* l_{n-1} – Desired location of head of the link l_{n-1} is $(x, y, z)^T$.
- Obtain location of the tail of link l_{n-1} using algorithm *TRACTRIX3D*.
- Recursively obtain the motion of the head and tail of all links down to the first link l_1.

- Consider a redundant manipulator with n links and joints $j_1, j_2, ..., j_{n-1}$ where j_i is the joint connecting link l_i and link l_{i+1} – joints are either spherical joints or rotary.
- Consider the last two links l_n and l_{n-1} – the head of the link l_n denoted by j_n is to be moved to $j_{n_{\mathrm{new}}}$ given by $\mathbf{X}_p = (x_p, y_p, z_p)^T$.
- Obtain new displaced location of tail j_{n-1} using algorithm *TRACTRIX3D* – denote by $\mathbf{X} = (x, y, z)^T$.
- Tail of the link l_n is the *head of the link* l_{n-1} – Desired location of head of the link l_{n-1} is $(x, y, z)^T$.
- Obtain location of the tail of link l_{n-1} using algorithm *TRACTRIX3D*.
- Recursively obtain the motion of the head and tail of all links down to the first link l_1.

- Consider a redundant manipulator with n links and joints $j_1, j_2, ..., j_{n-1}$ where j_i is the joint connecting link l_i and link l_{i+1} – joints are either spherical joints or rotary.
- Consider the last two links l_n and l_{n-1} – the head of the link l_n denoted by j_n is to be moved to $j_{n_{\text{new}}}$ given by $\mathbf{X}_p = (x_p, y_p, z_p)^T$.
- Obtain new displaced location of tail j_{n-1} using algorithm *TRACTRIX3D* – denote by $\mathbf{X} = (x, y, z)^T$.
- Tail of the link l_n is the *head of the link* l_{n-1} – Desired location of head of the link l_{n-1} is $(x, y, z)^T$.
- Obtain location of the tail of link l_{n-1} using algorithm *TRACTRIX3D*.
- Recursively obtain the motion of the head and tail of all links down to the first link l_1.

Algorithm *RESOLUTION-TRACTRIX*

1. Input desired location of head of link l_n $(x_p, y_p, z_p)^T$ and set $j_{n_{\mathrm{new}}} = (x_p, y_p, z_p)^T$.
2. for $i : n \rightarrow 1$
   - Call *TRACTRIX3D* and obtain location of the tail of link i $(x,y,z)^T_{i-1}$
   - Set new location of head of link $i-1$, $j_{i-1_{\mathrm{new}}} \leftarrow (x,y,z)^T_{i-1}$
3. At end of step **2**, j_0, would have moved. To *fix* j_0
   - Move j_0 to the origin $(0,0,0)^T$ and translate 'rigidly' *all* other links with no rotations at the joints.
   - Due to 'rigid' translation, the end-effector will not be at the desired (x_p, y_p, z_p).
   - Repeat step **2** and **3** until the head reaches (x_p, y_p, z_p) and the point j_0 is within a prescribed error bound of $(0,0,0)$.

See also Reznick and Lumelsky(1992, 1993, 1995).

Algorithm *RESOLUTION-TRACTRIX*

1. Input desired location of head of link l_n $(x_p, y_p, z_p)^T$ and set $j_{n_{\text{new}}} = (x_p, y_p, z_p)^T$.
2. for $i : n \rightarrow 1$
   - Call *TRACTRIX3D* and obtain location of the tail of link i $(x, y, z)^T_{i-1}$
   - Set new location of head of link $i-1$, $j_{i-1_{\text{new}}} \leftarrow (x, y, z)^T_{i-1}$
3. At end of step **2**, j_0, would have moved. To *fix* j_0
   - Move j_0 to the origin $(0,0,0)^T$ and translate 'rigidly' *all* other links with no rotations at the joints.
   - Due to 'rigid' translation, the end-effector will not be at the desired (x_p, y_p, z_p).
   - Repeat step **2** and **3** until the head reaches (x_p, y_p, z_p) and the point j_0 is within a prescribed error bound of $(0,0,0)$.

See also Reznick and Lumelsky(1992, 1993, 1995).

Algorithm *RESOLUTION-TRACTRIX*

1. Input desired location of head of link l_n $(x_p, y_p, z_p)^T$ and set $j_{n_{\text{new}}} = (x_p, y_p, z_p)^T$.
2. for $i : n \rightarrow 1$
   - Call *TRACTRIX3D* and obtain location of the tail of link i $(x, y, z)^T_{i-1}$
   - Set new location of head of link $i-1$, $j_{i-1_{\text{new}}} \leftarrow (x, y, z)^T_{i-1}$
3. At end of step **2**, j_0, would have moved. To *fix* j_0
   - Move j_0 to the origin $(0, 0, 0)^T$ and translate 'rigidly' *all* other links with no rotations at the joints.
   - Due to 'rigid' translation, the end-effector will not be at the desired (x_p, y_p, z_p).
   - Repeat step **2** and **3** until the head reaches (x_p, y_p, z_p) and the point j_0 is within a prescribed error bound of $(0, 0, 0)$.

See also Reznick and Lumelsky(1992, 1993, 1995).

1. Algorithm complexity is $\mathscr{O}(n)$ where n is the number of rigid links $\rightarrow$ amenable for real time computation.
2. θ_i is given by $\theta_i = \cos^{-1}(\overrightarrow{j_{i-1}j_i}(k+1) \cdot \overrightarrow{j_{i-1}j_i}(k))$ where $\overrightarrow{j_{i-1}j_i}(k)$ is the unit vector from the tail to the head of the i-th link at k-th instant.
3. The resolution of redundancy is done in *Cartesian space* and then the joint angles are computed.
4. When the head of the link l_n moves by dr_n the displacements obey the inequality $dr_0 \leq dr_1 \leq ... \leq dr_{n-1} \leq dr_n$.
   - The motion of the links appears to 'die' out as we move toward the first link.
   - Joints near to base 'see' large inertia and a desirable strategy would be to move them the least.
5. To 'fix' the tail of the first link, perform iterations of step 3 – convergence is guaranteed due to 'dying' out property.

Properties of Algorithm *Resolution-Tractrix*

1. Algorithm complexity is $\mathcal{O}(n)$ where n is the number of rigid links $\rightarrow$ amenable for real time computation.
2. θ_i is given by $\theta_i = \cos^{-1}(\overrightarrow{j_{i-1}j_i}(k+1) \cdot \overrightarrow{j_{i-1}j_i}(k))$ where $\overrightarrow{j_{i-1}j_i}(k)$ is the unit vector from the tail to the head of the i-th link at k-th instant.
3. The resolution of redundancy is done in *Cartesian space* and then the joint angles are computed.
4. When the head of the link l_n moves by dr_n the displacements obey the inequality $dr_0 \leq dr_1 \leq ... \leq dr_{n-1} \leq dr_n$.
   - The motion of the links appears to 'die' out as we move toward the first link.
   - Joints near to base 'see' large inertia and a desirable strategy would be to move them the least.
5. To 'fix' the tail of the first link, perform iterations of step 3 – convergence is guaranteed due to 'dying' out property.

1. Algorithm complexity is $\mathcal{O}(n)$ where n is the number of rigid links $\rightarrow$ amenable for real time computation.
2. θ_i is given by $\theta_i = \cos^{-1}(\overrightarrow{j_{i-1}j_i}(k+1) \cdot \overrightarrow{j_{i-1}j_i}(k))$ where $\overrightarrow{j_{i-1}j_i}(k)$ is the unit vector from the tail to the head of the i-th link at k-th instant.
3. The resolution of redundancy is done in *Cartesian space* and then the joint angles are computed.
4. When the head of the link l_n moves by dr_n the displacements obey the inequality $dr_0 \leq dr_1 \leq ... \leq dr_{n-1} \leq dr_n$.
   - The motion of the links appears to 'die' out as we move toward the first link.
   - Joints near to base 'see' large inertia and a desirable strategy would be to move them the least.
5. To 'fix' the tail of the first link, perform iterations of step **3** – convergence is guaranteed due to 'dying' out property.

Redundant Manipulators

Properties of Algorithm *Resolution-Tractrix*

1. Algorithm complexity is $\mathcal{O}(n)$ where n is the number of rigid links $\rightarrow$ amenable for real time computation.
2. θ_i is given by $\theta_i = \cos^{-1}(\overrightarrow{j_{i-1}j_i}(k+1) \cdot \overrightarrow{j_{i-1}j_i}(k))$ where $\overrightarrow{j_{i-1}j_i}(k)$ is the unit vector from the tail to the head of the i-th link at k-th instant.
3. The resolution of redundancy is done in *Cartesian space* and then the joint angles are computed.
4. When the head of the link l_n moves by dr_n the displacements obey the inequality $dr_0 \leq dr_1 \leq ... \leq dr_{n-1} \leq dr_n$.
   - The motion of the links appears to 'die' out as we move toward the first link.
   - Joints near to base 'see' large inertia and a desirable strategy would be to move them the least.
5. To 'fix' the tail of the first link, perform iterations of step **3** – convergence is guaranteed due to 'dying' out property.

1. Algorithm complexity is $\mathcal{O}(n)$ where n is the number of rigid links $\rightarrow$ amenable for real time computation.
2. θ_i is given by $\theta_i = \cos^{-1}(\overrightarrow{j_{i-1}j_i}(k+1) \cdot \overrightarrow{j_{i-1}j_i}(k))$ where $\overrightarrow{j_{i-1}j_i}(k)$ is the unit vector from the tail to the head of the i-th link at k-th instant.
3. The resolution of redundancy is done in *Cartesian space* and then the joint angles are computed.
4. When the head of the link l_n moves by dr_n the displacements obey the inequality $dr_0 \leq dr_1 \leq ... \leq dr_{n-1} \leq dr_n$.
   - The motion of the links appears to 'die' out as we move toward the first link.
   - Joints near to base 'see' large inertia and a desirable strategy would be to move them the least.
5. To 'fix' the tail of the first link, perform iterations of step **3** – convergence is guaranteed due to 'dying' out property.

Redundant Manipulators

Experimental hardware

- Experimental 8-link planar manipulator – each link is 70 mm long.
- Joint driven by Futaba S3003 RC hobby servos.

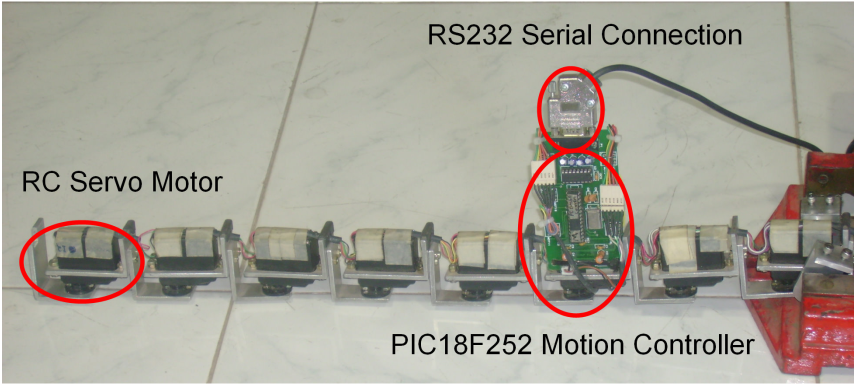
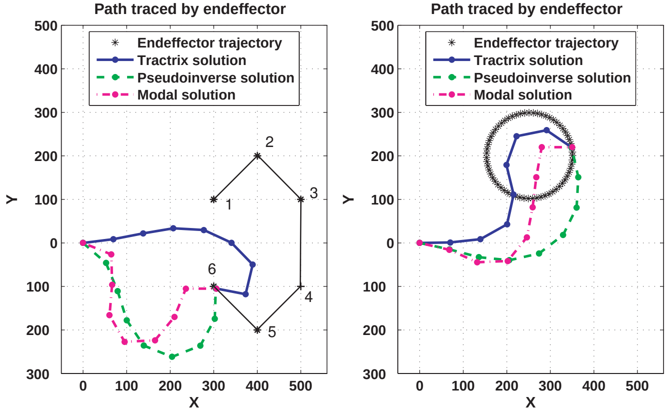


Figure 14: Experimental 8-link hyper-redundant manipulator.

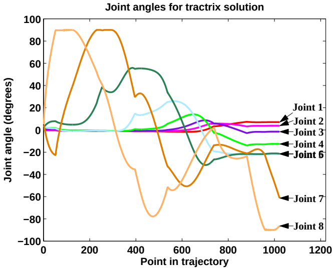


(a) Desired straight line motions *(b) Desired circular motion*

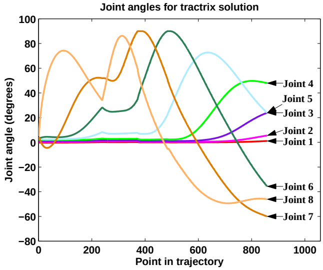
Figures show final configurations of robot using 3 approaches.

Redundant Manipulators (contd.)

Simulation results

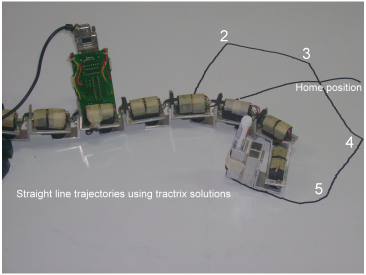


(c) Plot of joint variables for straight line motions

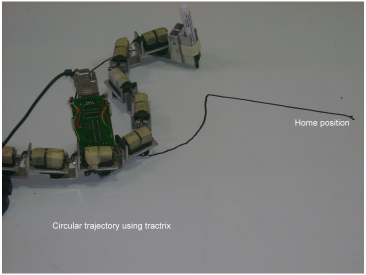


(d) Plot of joint variables for circular motion

Joints toward the base move the least.
See references Ravi et al.(2010) for more details and comparison with other approaches.



(e) *Straight line motion*



(f) *Circular motion*

- Minimising Cartesian motion of links as motion 'die' out from end-effector to base.
- Tractrix based resolution scheme is more *natural*.
- Videos: Pseudo-inverse method, Modal approach, and Tractrix based approach for straight line trajectory.
- Videos: Pseudo-inverse method, Modal approach, and Tractrix approach circular trajectory.

Redundant Manipulators

Free motion

- One end of redundant manipulator is not held *fixed* – becomes a snake.
- Desired (x_p, y_p) provided from a computer and joint motions computed using tractrix approach.
- 8-link planar manipulator moves in a snake-like manner.
- Motion appears to be natural.

See movie – free motion of a hyper-redundant snake manipulator.

Tying knots, snake motion ....

- Tractrix based approaches can be extended to spatial hyper-redundant systems.
- Link to videos on single-hand knot tying, two-hand knot tying and simulated motion of a snake.
- Each of the simulation uses a tractrix based approach (Goel et al., 2010), and motion appears to be more *natural*.

Outline

Review of Inverse Kinematics

- Inverse kinematics involves solution of a set of non-linear transcendental equations.
- A closed-form (analytical) solution is desired over a purely iterative or numerical approach.
- Closed-form solutions provide criterion for workspace and multiple configurations.
- General approach for inverse kinematics:
  - Convert transcendental equations to polynomial equations using tangent half angle substitution.
  - Eliminate sequentially (or if possible in one step) joint variables to arrive at *single polynomial* in *one joint variable*.
  - Solve if possible in closed-form (for polynomials up to quartic – see Korn and Korn, 1968) for the unknown joint variable.
  - Obtain other joint variables by back substitution.
- Key step is to obtain the *monomial* by *elimination*.

Review of Inverse Kinematics

- Inverse kinematics involves solution of a set of non-linear transcendental equations.
- A closed-form (analytical) solution is desired over a purely iterative or numerical approach.
- Closed-form solutions provide criterion for workspace and multiple configurations.
- General approach for inverse kinematics:
  - Convert transcendental equations to polynomial equations using tangent half angle substitution.
  - Eliminate sequentially (or if possible in one step) joint variables to arrive at *single polynomial* in *one joint variable*.
  - Solve if possible in closed-form (for polynomials up to quartic – see Korn and Korn, 1968) for the unknown joint variable.
  - Obtain other joint variables by back substitution.
- Key step is to obtain the *monomial* by *elimination*.

Review of Inverse Kinematics

- Inverse kinematics involves solution of a set of non-linear transcendental equations.
- A closed-form (analytical) solution is desired over a purely iterative or numerical approach.
- Closed-form solutions provide criterion for workspace and multiple configurations.
- General approach for inverse kinematics:
  - Convert transcendental equations to polynomial equations using tangent half angle substitution.
  - Eliminate sequentially (or if possible in one step) joint variables to arrive at *single polynomial* in *one joint variable*.
  - Solve if possible in closed-form (for polynomials up to quartic – see Korn and Korn, 1968) for the unknown joint variable.
  - Obtain other joint variables by back substitution.
- Key step is to obtain the *monomial* by *elimination*.

Review of Inverse Kinematics

- Inverse kinematics involves solution of a set of non-linear transcendental equations.
- A closed-form (analytical) solution is desired over a purely iterative or numerical approach.
- Closed-form solutions provide criterion for workspace and multiple configurations.
- General approach for inverse kinematics:
  - Convert transcendental equations to polynomial equations using tangent half angle substitution.
  - Eliminate sequentially (or if possible in one step) joint variables to arrive at *single polynomial* in *one joint variable*.
  - Solve if possible in closed-form (for polynomials up to quartic – see Korn and Korn, 1968) for the unknown joint variable.
  - Obtain other joint variables by back substitution.
- Key step is to obtain the *monomial* by *elimination*.

- Polynomial equations $f(x,y)=0$ and $g(x,y)=0$ of degree m and n
- Degree of a polynomial is sum of exponents of the highest degree term.
- Bézout Theorem (Semple and Roth, 1949): a maximum of $m \times n$ (x,y) values satisfy both the equations.
- Bézout Theorem give *upper bound* and includes real, complex conjugate and solutions at infinity.
- Example 1: $x^2+y^2=1$ and $y-x=0$ are satisfied by two sets of (x,y) values – $\pm(\frac{1}{\sqrt{2}},\frac{1}{\sqrt{2}})$.
- Example 2: $x^2+y^2=1$ and $y-x=2$ are *not* satisfied by any *real* values of (x,y).
- Example 3: $x^2+y^2=1$ and $y-x=\sqrt{2}$ satisfied by *two* coincident *real* values of (x,y).

- Polynomial equations $f(x,y) = 0$ and $g(x,y) = 0$ of degree m and n
- Degree of a polynomial is sum of exponents of the highest degree term.
- Bézout Theorem (Semple and Roth, 1949): a maximum of $m \times n$ (x,y) values satisfy both the equations.
- Bézout Theorem give *upper bound* and includes real, complex conjugate and solutions at infinity.
- Example 1: $x^2 + y^2 = 1$ and $y - x = 0$ are satisfied by two sets of (x,y) values – $\pm(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}})$.
- Example 2: $x^2 + y^2 = 1$ and $y - x = 2$ are *not* satisfied by any *real* values of (x,y).
- Example 3: $x^2 + y^2 = 1$ and $y - x = \sqrt{2}$ satisfied by *two* coincident *real* values of (x,y).

- Polynomial equations $f(x,y)=0$ and $g(x,y)=0$ of degree m and n
- Degree of a polynomial is sum of exponents of the highest degree term.
- Bézout Theorem (Semple and Roth, 1949): a maximum of $m \times n$ (x,y) values satisfy both the equations.
- Bézout Theorem give *upper bound* and includes real, complex conjugate and solutions at infinity.
- Example 1: $x^2+y^2=1$ and $y-x=0$ are satisfied by two sets of (x,y) values – $\pm(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}})$.
- Example 2: $x^2+y^2=1$ and $y-x=2$ are *not* satisfied by any *real* values of (x,y).
- Example 3: $x^2+y^2=1$ and $y-x=\sqrt{2}$ satisfied by *two* coincident *real* values of (x,y).

- Polynomial equations $f(x,y)=0$ and $g(x,y)=0$ of degree m and n
- Degree of a polynomial is sum of exponents of the highest degree term.
- Bézout Theorem (Semple and Roth, 1949): a maximum of $m \times n$ (x,y) values satisfy both the equations.
- Bézout Theorem give *upper bound* and includes real, complex conjugate and solutions at infinity.
- Example 1: $x^2+y^2=1$ and $y-x=0$ are satisfied by two sets of (x,y) values – $\pm(\frac{1}{\sqrt{2}},\frac{1}{\sqrt{2}})$.
- Example 2: $x^2+y^2=1$ and $y-x=2$ are *not* satisfied by any *real* values of (x,y).
- Example 3: $x^2+y^2=1$ and $y-x=\sqrt{2}$ satisfied by *two* coincident *real* values of (x,y).

- Example 1, 2 and 3 can also be interpreted *geometrically*.
  - Sketch show that line $y - x = 0$ *intersects* circle $x^2 + y^2 = 1$ at two points.
  - Sketch show that line $y - x = 2$ *does not intersects* circle $x^2 + y^2 = 1$.
  - Sketch show that line $y - x = \sqrt{2}$ is *tangent* to circle $x^2 + y^2 = 1$.
- Verify: Two parabolas, ellipses or hyperbolas (quadratic curves) *can* intersect in 4 points.
- Apparent contradiction: Two circles *never* intersect at 4 points.
- Contradiction can be resolved if *homogeneous coordinates* (see additional material in Module 2) (x, y, w) is used to represent equations of circles.
- In terms of homogeneous coordinates – two complex conjugates solutions at ∞ for *any two* circles.

- Example 1, 2 and 3 can also be interpreted *geometrically*.
  - Sketch show that line $y - x = 0$ *intersects* circle $x^2 + y^2 = 1$ at two points.
  - Sketch show that line $y - x = 2$ *does not intersects* circle $x^2 + y^2 = 1$.
  - Sketch show that line $y - x = \sqrt{2}$ is *tangent* to circle $x^2 + y^2 = 1$.
- Verify: Two parabolas, ellipses or hyperbolas (quadratic curves) *can* intersect in 4 points.
- Apparent contradiction: Two circles *never* intersect at 4 points.
- Contradiction can be resolved if *homogeneous coordinates* (see additional material in Module 2) (x, y, w) is used to represent equations of circles.
- In terms of homogeneous coordinates – two complex conjugates solutions at ∞ for *any two* circles.

- Example 1, 2 and 3 can also be interpreted *geometrically*.
  - Sketch show that line $y - x = 0$ *intersects* circle $x^2 + y^2 = 1$ at two points.
  - Sketch show that line $y - x = 2$ *does not intersects* circle $x^2 + y^2 = 1$.
  - Sketch show that line $y - x = \sqrt{2}$ is *tangent* to circle $x^2 + y^2 = 1$.
- Verify: Two parabolas, ellipses or hyperbolas (quadratic curves) *can* intersect in 4 points.
- Apparent contradiction: Two circles *never* intersect at 4 points.
- Contradiction can be resolved if *homogeneous coordinates* (see additional material in Module 2) (x, y, w) is used to represent equations of circles.
- In terms of homogeneous coordinates – two complex conjugates solutions at ∞ for *any two* circles.

- Example 1, 2 and 3 can also be interpreted *geometrically*.
  - Sketch show that line $y - x = 0$ *intersects* circle $x^2 + y^2 = 1$ at two points.
  - Sketch show that line $y - x = 2$ *does not intersects* circle $x^2 + y^2 = 1$.
  - Sketch show that line $y - x = \sqrt{2}$ is *tangent* to circle $x^2 + y^2 = 1$.
- Verify: Two parabolas, ellipses or hyperbolas (quadratic curves) *can* intersect in 4 points.
- Apparent contradiction: Two circles *never* intersect at 4 points.
- Contradiction can be resolved if *homogeneous coordinates* (see additional material in Module 2) (x, y, w) is used to represent equations of circles.
- In terms of homogeneous coordinates – two complex conjugates solutions at ∞ for *any two* circles.

- Bézout Theorem can be extended to two $m-$ and $n-$ order manifolds $\rightarrow$ they intersect in *at most* a $m \times n$ order sub-manifold.
- Example 1: A sphere $x^2 + y^2 + z^2 = 1$ $(m = 2)$ intersects a plane $x = 0$ $(n = 1)$ in a circle $y^2 + z^2 = 1$ – a second-order curve.
- Example 2: Two cylinders $(m = n = 2)$ can intersect in a fourth degree curve.
- Bézout theorem is of no use in *obtaining* the solutions – it is not a constructive theorem.
- One constructive method is Sylvester's dialytic elimination method (Salmon, 1964).

- Bézout Theorem can be extended to two $m-$ and $n-$ order manifolds $\rightarrow$ they intersect in *at most* a $m \times n$ order sub-manifold.
- Example 1: A sphere $x^2 + y^2 + z^2 = 1$ $(m = 2)$ intersects a plane $x = 0$ $(n = 1)$ in a circle $y^2 + z^2 = 1$ – a second-order curve.
- Example 2: Two cylinders $(m = n = 2)$ can intersect in a fourth degree curve.
- Bézout theorem is of no use in *obtaining* the solutions – it is not a constructive theorem.
- One constructive method is Sylvester's dialytic elimination method (Salmon, 1964).

- Bézout Theorem can be extended to two $m-$ and $n-$ order manifolds $\rightarrow$ they intersect in *at most* a $m \times n$ order sub-manifold.
- Example 1: A sphere $x^2 + y^2 + z^2 = 1$ $(m = 2)$ intersects a plane $x = 0$ $(n = 1)$ in a circle $y^2 + z^2 = 1$ – a second-order curve.
- Example 2: Two cylinders $(m = n = 2)$ can intersect in a fourth degree curve.
- Bézout theorem is of no use in *obtaining* the solutions – it is not a constructive theorem.
- One constructive method is Sylvester's dialytic elimination method (Salmon, 1964).

- Bézout Theorem can be extended to two $m-$ and $n-$ order manifolds $\rightarrow$ they intersect in *at most* a $m \times n$ order sub-manifold.
- Example 1: A sphere $x^2+y^2+z^2=1$ $(m=2)$ intersects a plane $x=0$ $(n=1)$ in a circle $y^2+z^2=1$ – a second-order curve.
- Example 2: Two cylinders $(m=n=2)$ can intersect in a fourth degree curve.
- Bézout theorem is of no use in *obtaining* the solutions – it is not a constructive theorem.
- One constructive method is Sylvester's dialytic elimination method (Salmon, 1964).

- Two polynomials $P(x) = \sum_{i=0}^{m} a_i x^i = 0$ and $Q(x) = \sum_{i=0}^{n} b_i x^i = 0$, a_i and b_i are co-efficients.
- Construct the Sylvester's matrix of $P(x)$ and $Q(x)$

$$[SM] = \begin{bmatrix} a_m & a_{m-1} & \dots & a_1 & a_0 & & \\ & a_m & a_{m-1} & \dots & a_1 & a_0 & \\ & & & \dots & \dots & \dots & \\ & & & a_m & \dots & \dots & a_0 \\ b_n & b_{n-1} & \dots & b_1 & b_0 & & \\ & b_n & b_{n-1} & \dots & b_1 & b_0 & \\ & & & \dots & \dots & \dots & \\ & & & b_n & \dots & \dots & b_0 \end{bmatrix} \quad (32)$$

where the unfilled entries are 0 & $[SM]$ is $(m+n) \times (m+n)$.
- The i^{th} row of the top half are the co-efficients of $P(x) \times x^i$ for $i = n-1, n-2, \dots, 1, 0$
- The i^{th} row in the bottom half are the co-efficients of $Q(x) \times x^i$ for $i = m-1, m-2, \dots, 1, 0$.

- Two polynomials $P(x) = \sum_{i=0}^{m} a_i x^i = 0$ and $Q(x) = \sum_{i=0}^{n} b_i x^i = 0$, a_i and b_i are co-efficients.
- Construct the Sylvester's matrix of $P(x)$ and $Q(x)$

$$[SM] = \begin{bmatrix} a_m & a_{m-1} & \ldots & a_1 & a_0 & & \\ & a_m & a_{m-1} & \ldots & a_1 & a_0 & \\ & & & \ldots & \ldots & \ldots & \\ & & & a_m & \ldots & \ldots & a_0 \\ b_n & b_{n-1} & \ldots & b_1 & b_0 & & \\ & b_n & b_{n-1} & \ldots & b_1 & b_0 & \\ & & & \ldots & \ldots & \ldots & \\ & & & b_n & \ldots & \ldots & b_0 \end{bmatrix} \quad (32)$$

where the unfilled entries are 0 & $[SM]$ is $(m+n)\times(m+n)$.

- The i^{th} row of the top half are the co-efficients of $P(x)\times x^i$ for $i = n-1, n-2, ..., 1, 0$
- The i^{th} row in the bottom half are the co-efficients of $Q(x)\times x^i$ for $i = m-1, m-2, ..., 1, 0$.

- Two polynomials $P(x) = \sum_{i=0}^{m} a_i x^i = 0$ and $Q(x) = \sum_{i=0}^{n} b_i x^i = 0$, a_i and b_i are co-efficients.
- Construct the Sylvester's matrix of $P(x)$ and $Q(x)$

$$[SM] = \begin{bmatrix} a_m & a_{m-1} & \ldots & a_1 & a_0 & & \\ & a_m & a_{m-1} & \ldots & a_1 & a_0 & \\ & & & \ldots & \ldots & \ldots & \\ & & & a_m & \ldots & \ldots & a_0 \\ b_n & b_{n-1} & \ldots & b_1 & b_0 & & \\ & b_n & b_{n-1} & \ldots & b_1 & b_0 & \\ & & & \ldots & \ldots & \ldots & \\ & & & b_n & \ldots & \ldots & b_0 \end{bmatrix} \quad (32)$$

where the unfilled entries are 0 & $[SM]$ is $(m+n) \times (m+n)$.
- The i^{th} row of the top half are the co-efficients of $P(x) \times x^i$ for $i = n-1, n-2, \ldots, 1, 0$
- The i^{th} row in the bottom half are the co-efficients of $Q(x) \times x^i$ for $i = m-1, m-2, \ldots, 1, 0$.

- Sylvester criterion[1]: $P(x) = 0$ and $Q(x) = 0$ have a *non-trivial* common factor *if and only if* $\det[SM] = 0$.
- The Sylvester criterion follows from the analogy with linear equations.
  - The n equations $P(x) \times x^i = 0,\ i = n-1, n-2, ..., 1, 0$ and the m equations $Q(x) \times x^i$ for $i = m-1, m-2, ..., 1, 0$ can be written as

$$[SM](x^{m+n-1}, x^{m+n-2}, ..., x^1, x^0)^T = \mathbf{0} \tag{33}$$

  - Note: all powers of x, $x^{m+n-1}, x^{m+n-2}, ..., x^1, x^0$, *including* the constant term x^0 are treated as *linearly* independent variables.
  - Note: The matrix $[SM]$ is *square* and is of dimension $(m+n) \times (m+n)$.
- The set of linear equations (33) can have a non-trivial solution *if and only if* $\det[SM] = 0$ – Same as the Sylvester's criterion!

[1] Sylvester and Trudi worked in the late 19th century on the theory of equations, later called the theory of algebraic curves, which forms the foundation of algebraic geometry.

Elimination Theory

Sylvester's Method (Contd.)

- Sylvester criterion[1]: $P(x) = 0$ and $Q(x) = 0$ have a *non-trivial* common factor *if and only if* $\det[SM] = 0$.
- The Sylvester criterion follows from the analogy with linear equations.
  - The n equations $P(x) \times x^i = 0,\ i = n-1, n-2, ..., 1, 0$ and the m equations $Q(x) \times x^i$ for $i = m-1, m-2, ..., 1, 0$ can be written as

$$[SM](x^{m+n-1}, x^{m+n-2}, ..., x^1, x^0)^T = \mathbf{0} \tag{33}$$

  - Note: all powers of x, $x^{m+n-1}, x^{m+n-2}, ..., x^1, x^0$, *including* the constant term x^0 are treated as *linearly* independent variables.
  - Note: The matrix $[SM]$ is *square* and is of dimension $(m+n) \times (m+n)$.
- The set of linear equations (33) can have a non-trivial solution *if and only if* $\det[SM] = 0$ – Same as the Sylvester's criterion!

[1] Sylvester and Trudi worked in the late 19th century on the theory of equations, later called the theory of algebraic curves, which forms the foundation of algebraic geometry.

- Sylvester criterion[1]: $P(x) = 0$ and $Q(x) = 0$ have a *non-trivial* common factor *if and only if* $\det[SM] = 0$.
- The Sylvester criterion follows from the analogy with linear equations.
  - The n equations $P(x) \times x^i = 0,\ i = n-1, n-2, ..., 1, 0$ and the m equations $Q(x) \times x^i$ for $i = m-1, m-2, ..., 1, 0$ can be written as

$$[SM](x^{m+n-1}, x^{m+n-2}, ..., x^1, x^0)^T = \mathbf{0} \tag{33}$$

  - Note: all powers of x, $x^{m+n-1}, x^{m+n-2}, ..., x^1, x^0$, *including* the constant term x^0 are treated as *linearly* independent variables.
  - Note: The matrix $[SM]$ is *square* and is of dimension $(m+n) \times (m+n)$.
- The set of linear equations (33) can have a non-trivial solution *if and only if* $\det[SM] = 0$ – Same as the Sylvester's criterion!

[1] Sylvester and Trudi worked in the late 19th century on the theory of equations, later called the theory of algebraic curves, which forms the foundation of algebraic geometry.

Algorithm to solve two polynomials, $f(x,y) = 0$ and $g(x,y) = 0$

1. Rewrite $f(x,y) = 0$ and $g(x,y) = 0$ as $P(x) = \sum_{i=0}^{m} a_i(y)x^i = 0$ and $Q(x) = \sum_{i=0}^{n} b_i(y)x^i = 0$. Note: all coefficients function of y or constant.
2. Obtain $[SM](y)$ and compute $\det[SM](y) = 0 \rightarrow$ A polynomial in y alone.
3. Solve $\det[SM](y) = 0$ for all roots analytically (if possible) or numerically.
4. Linear equations (33) can be solved, using linear algebra techniques, for the *linearly independent* unknowns $x^{m+n-1}, x^{m+n-2}, \ldots, x^1, x^0$.
5. The integrity of the numerical procedure can be verified by checking that x^1 and say x^2 are related by $(x^1)^2 = x^2$.

- Consider two polynomial equations

$$\begin{aligned} f_1(x,y) &= a_2(y)x^2 + a_1(y)x + a_0(y) = 0 \\ f_2(x,y) &= b_2(y)x^2 + b_1(y)x + b_0(y) = 0 \end{aligned} \quad (34)$$

where a_i and b_i, $i = 0, 1, 2$ are arbitrary polynomials in y or constants.

- Sylvester's matrix is given by

$$[SM] = \begin{bmatrix} a_2 & a_1 & a_0 & 0 \\ 0 & a_2 & a_1 & a_0 \\ b_2 & b_1 & b_0 & 0 \\ 0 & b_2 & b_1 & b_0 \end{bmatrix} \quad (35)$$

- $\det[SM](y) = 0$ reduces to
$(a_2 b_1 - b_2 a_1)(a_1 b_0 - b_1 a_0) - (a_2 b_0 - b_2 a_0)^2 = 0$

- Consider two polynomial equations

$$\begin{aligned} f_1(x,y) &= a_2(y)x^2 + a_1(y)x + a_0(y) = 0 \\ f_2(x,y) &= b_2(y)x^2 + b_1(y)x + b_0(y) = 0 \end{aligned} \tag{34}$$

where a_i and b_i, $i = 0,1,2$ are arbitrary polynomials in y or constants.

- Sylvester's matrix is given by

$$[SM] = \begin{bmatrix} a_2 & a_1 & a_0 & 0 \\ 0 & a_2 & a_1 & a_0 \\ b_2 & b_1 & b_0 & 0 \\ 0 & b_2 & b_1 & b_0 \end{bmatrix} \tag{35}$$

- $\det[SM](y) = 0$ reduces to
$(a_2 b_1 - b_2 a_1)(a_1 b_0 - b_1 a_0) - (a_2 b_0 - b_2 a_0)^2 = 0$

- Consider two polynomial equations

$$\begin{aligned} f_1(x,y) &= a_2(y)x^2 + a_1(y)x + a_0(y) = 0 \\ f_2(x,y) &= b_2(y)x^2 + b_1(y)x + b_0(y) = 0 \end{aligned} \tag{34}$$

where a_i and b_i, $i = 0, 1, 2$ are arbitrary polynomials in y or constants.

- Sylvester's matrix is given by

$$[SM] = \begin{bmatrix} a_2 & a_1 & a_0 & 0 \\ 0 & a_2 & a_1 & a_0 \\ b_2 & b_1 & b_0 & 0 \\ 0 & b_2 & b_1 & b_0 \end{bmatrix} \tag{35}$$

- $\det[SM](y) = 0$ reduces to
$(a_2 b_1 - b_2 a_1)(a_1 b_0 - b_1 a_0) - (a_2 b_0 - b_2 a_0)^2 = 0$

- The variable x can be obtained from the set of 'linear equations'

$$\begin{bmatrix} a_2 & a_1 & a_0 & 0 \\ 0 & a_2 & a_1 & a_0 \\ b_2 & b_1 & b_0 & 0 \\ 0 & b_2 & b_1 & b_0 \end{bmatrix} \begin{pmatrix} x^3 \\ x^2 \\ x^1 \\ x^0 \end{pmatrix} = \mathbf{0} \tag{36}$$

as

$$x = x^1 = -\frac{a_1 b_0 - b_1 a_0}{a_2 b_0 - b_2 a_0} = -\frac{a_2 b_0 - b_2 a_0}{a_1 b_2 - a_2 b_1} \tag{37}$$

- Note: x computed using the *two* expressions on the right-hand side must be same and can be used as a programming/numerical consistency check
- Note: $\frac{a_1 b_0 - b_1 a_0}{a_2 b_0 - b_2 a_0} = \frac{a_2 b_0 - b_2 a_0}{a_1 b_2 - a_2 b_1}$ is same as $\det[SM](y) = 0$!

- The variable x can be obtained from the set of 'linear equations'

$$\begin{bmatrix} a_2 & a_1 & a_0 & 0 \\ 0 & a_2 & a_1 & a_0 \\ b_2 & b_1 & b_0 & 0 \\ 0 & b_2 & b_1 & b_0 \end{bmatrix} \begin{pmatrix} x^3 \\ x^2 \\ x^1 \\ x^0 \end{pmatrix} = \mathbf{0} \qquad (36)$$

as

$$x = x^1 = -\frac{a_1 b_0 - b_1 a_0}{a_2 b_0 - b_2 a_0} = -\frac{a_2 b_0 - b_2 a_0}{a_1 b_2 - a_2 b_1} \qquad (37)$$

- Note: x computed using the *two* expressions on the right-hand side must be same and can be used as a programming/numerical consistency check
- Note: $\frac{a_1 b_0 - b_1 a_0}{a_2 b_0 - b_2 a_0} = \frac{a_2 b_0 - b_2 a_0}{a_1 b_2 - a_2 b_1}$ is same as $\det[SM](y) = 0$!

- The variable x can be obtained from the set of 'linear equations'

$$\begin{bmatrix} a_2 & a_1 & a_0 & 0 \\ 0 & a_2 & a_1 & a_0 \\ b_2 & b_1 & b_0 & 0 \\ 0 & b_2 & b_1 & b_0 \end{bmatrix} \begin{pmatrix} x^3 \\ x^2 \\ x^1 \\ x^0 \end{pmatrix} = \mathbf{0} \quad (36)$$

as

$$x = x^1 = -\frac{a_1 b_0 - b_1 a_0}{a_2 b_0 - b_2 a_0} = -\frac{a_2 b_0 - b_2 a_0}{a_1 b_2 - a_2 b_1} \quad (37)$$

- Note: x computed using the *two* expressions on the right-hand side must be same and can be used as a programming/numerical consistency check
- Note: $\frac{a_1 b_0 - b_1 a_0}{a_2 b_0 - b_2 a_0} = \frac{a_2 b_0 - b_2 a_0}{a_1 b_2 - a_2 b_1}$ is same as $\det[SM](y) = 0$!

- det$[SM]$ is also called the resultant of $P(x)$ and $Q(x)$ and is denoted by $\mathrm{res}(P,Q)$.
- $[SM]$ is $(m+n)\times(m+n)$ and $\mathrm{res}(P,Q)$ can become computationally expensive.
- Bézout in the 18th century proposed a method where $\mathrm{res}(P,Q)$ can be computed as a determinant of order $\max(m,n)$.
- The key idea is to *divide* instead of *multiplying* to get required number of *independent* equations and a square matrix.
- Although dimension of matrix is less, each element of the matrix is more complex.

- det[SM] is also called the resultant of $P(x)$ and $Q(x)$ and is denoted by $\mathrm{res}(P, Q)$.
- [SM] is $(m+n) \times (m+n)$ and $\mathrm{res}(P, Q)$ can become computationally expensive.
- Bézout in the 18th century proposed a method where $\mathrm{res}(P, Q)$ can be computed as a determinant of order $\max(m, n)$.
- The key idea is to *divide* instead of *multiplying* to get required number of *independent* equations and a square matrix.
- Although dimension of matrix is less, each element of the matrix is more complex.

- det$[SM]$ is also called the resultant of $P(x)$ and $Q(x)$ and is denoted by $\mathrm{res}(P, Q)$.
- $[SM]$ is $(m+n) \times (m+n)$ and $\mathrm{res}(P, Q)$ can become computationally expensive.
- Bézout in the 18th century proposed a method where $\mathrm{res}(P, Q)$ can be computed as a determinant of order $\max(m, n)$.
- The key idea is to *divide* instead of *multiplying* to get required number of *independent* equations and a square matrix.
- Although dimension of matrix is less, each element of the matrix is more complex.

- det$[SM]$ is also called the resultant of $P(x)$ and $Q(x)$ and is denoted by $\mathrm{res}(P, Q)$.
- $[SM]$ is $(m+n) \times (m+n)$ and $\mathrm{res}(P, Q)$ can become computationally expensive.
- Bézout in the 18th century proposed a method where $\mathrm{res}(P, Q)$ can be computed as a determinant of order $\max(m, n)$.
- The key idea is to *divide* instead of *multiplying* to get required number of *independent* equations and a square matrix.
- Although dimension of matrix is less, each element of the matrix is more complex.

- det$[SM]$ is also called the resultant of $P(x)$ and $Q(x)$ and is denoted by $\mathrm{res}(P, Q)$.
- $[SM]$ is $(m+n) \times (m+n)$ and $\mathrm{res}(P, Q)$ can become computationally expensive.
- Bézout in the 18th century proposed a method where $\mathrm{res}(P, Q)$ can be computed as a determinant of order $\max(m, n)$.
- The key idea is to *divide* instead of *multiplying* to get required number of *independent* equations and a square matrix.
- Although dimension of matrix is less, each element of the matrix is more complex.

Elimination Theory

Bézouts Matrix (contd.)

- Consider $P(x) = \sum_{i=0}^{m} a_i x^i = 0$ and $Q(x) = \sum_{i=0}^{n} b_i x^i = 0$ with $m > n$.
- Eliminate x^m from $P(x) = 0$ and $x^{m-n}Q(x) = 0$ by writing

$$\frac{a_m}{b_n} = \frac{a_{m-1}x^{m-1} + ... + a_0}{b_{n-1}x^{m-1} + ... + b_0 x^{m-n}} \qquad \text{to get}$$

$$(a_{m-1}b_n - a_m b_{n-1})x^{m-1} + (a_{m-2}b_n - a_m b_{n-2})x^{m-2} + ... + a_0 b_n = 0 \tag{38}$$

- Also eliminate x^m by writing

$$\frac{a_m x + a_{m-1}}{b_n x + b_{n-1}} = \frac{a_{m-2}x^{m-2} + ... + a_0}{b_{n-2}x^{m-2} + ... + b_0 x^{m-n}} \qquad \text{to get}$$

$$\begin{aligned}
&(a_{m-2}b_n - b_{n-2}a_m)x^{m-1} + \\
&[(a_{m-3}b_n - b_{n-3}a_m) + (a_{m-2}b_{n-1} - b_{n-2}a_{m-1})]x^{m-2} \\
&+ ... + a_0 b_{n-1} = 0
\end{aligned} \tag{39}$$

Elimination Theory

Bézouts Matrix (contd.)

- Consider $P(x) = \sum_{i=0}^{m} a_i x^i = 0$ and $Q(x) = \sum_{i=0}^{n} b_i x^i = 0$ with $m > n$.
- Eliminate x^m from $P(x) = 0$ and $x^{m-n}Q(x) = 0$ by writing

$$\frac{a_m}{b_n} = \frac{a_{m-1}x^{m-1} + ... + a_0}{b_{n-1}x^{m-1} + ... + b_0 x^{m-n}} \qquad \text{to get}$$

$$(a_{m-1}b_n - a_m b_{n-1})x^{m-1} + (a_{m-2}b_n - a_m b_{n-2})x^{m-2} + ... + a_0 b_n = 0 \tag{38}$$

- Also eliminate x^m by writing

$$\frac{a_m x + a_{m-1}}{b_n x + b_{n-1}} = \frac{a_{m-2}x^{m-2} + ... + a_0}{b_{n-2}x^{m-2} + ... + b_0 x^{m-n}} \qquad \text{to get}$$

$$\begin{aligned} &(a_{m-2}b_n - b_{n-2}a_m)x^{m-1} + \\ &[(a_{m-3}b_n - b_{n-3}a_m) + (a_{m-2}b_{n-1} - b_{n-2}a_{m-1})]x^{m-2} \\ &+ ... + a_0 b_{n-1} = 0 \end{aligned} \tag{39}$$

Elimination Theory

Bézouts Matrix (contd.)

- Consider $P(x) = \sum_{i=0}^{m} a_i x^i = 0$ and $Q(x) = \sum_{i=0}^{n} b_i x^i = 0$ with $m > n$.
- Eliminate x^m from $P(x) = 0$ and $x^{m-n} Q(x) = 0$ by writing

$$\frac{a_m}{b_n} = \frac{a_{m-1}x^{m-1} + ... + a_0}{b_{n-1}x^{m-1} + ... + b_0 x^{m-n}} \qquad \text{to get}$$

$$(a_{m-1}b_n - a_m b_{n-1})x^{m-1} + (a_{m-2}b_n - a_m b_{n-2})x^{m-2} + ... + a_0 b_n = 0 \tag{38}$$

- Also eliminate x^m by writing

$$\frac{a_m x + a_{m-1}}{b_n x + b_{n-1}} = \frac{a_{m-2}x^{m-2} + ... + a_0}{b_{n-2}x^{m-2} + ... + b_0 x^{m-n}} \qquad \text{to get}$$

$$\begin{aligned}
&(a_{m-2}b_n - b_{n-2}a_m)x^{m-1} + \\
&[(a_{m-3}b_n - b_{n-3}a_m) + (a_{m-2}b_{n-1} - b_{n-2}a_{m-1})]x^{m-2} \\
&+ ... + a_0 b_{n-1} = 0
\end{aligned} \tag{39}$$

Elimination Theory

Bézouts Matrix (contd.)

- Repeat to obtain n equations with the n^{th} equation given by

$$(a_{m-n}b_n - a_m b_0)x^{m-1} + (a_{m-n-1}b_n + a_{m-n}b_{n-1} - a_{m-1}b_0)x^{m-2} + ... + a_0 b_1 = 0 \quad (40)$$

- Construct $m-n$ equations

$$\begin{aligned} x^{m-n-1}Q(x) = b_n x^{m-1} + b_{n-1}x^{m-2} + \ldots + b_0 x^{m-n-1} &= 0 \\ x^{m-n-2}Q(x) = b_n x^{m-2} + \ldots + b_0 x^{m-n-2} &= 0 \\ \ldots.. &= 0 \\ Q(x) = b_n x^n + \ldots + b_0 &= 0 \quad (41) \end{aligned}$$

- Repeat to obtain n equations with the n^{th} equation given by

$$(a_{m-n}b_n - a_m b_0)x^{m-1} + (a_{m-n-1}b_n + a_{m-n}b_{n-1} - a_{m-1}b_0)x^{m-2} + ... + a_0 b_1 = 0 \qquad (40)$$

- Construct $m-n$ equations

$$\begin{aligned} x^{m-n-1}Q(x) = b_n x^{m-1} + b_{n-1}x^{m-2} + ... + b_0 x^{m-n-1} &= 0 \\ x^{m-n-2}Q(x) = b_n x^{m-2} + ... + b_0 x^{m-n-2} &= 0 \\ ..... &= 0 \\ Q(x) = b_n x^n + ... + b_0 &= 0 \quad (41) \end{aligned}$$

- The Bézout matrix is given as

$$\begin{bmatrix} a_{m-1}b_n - a_m b_{n-1} & a_{m-2}b_n - a_m b_{n-2} & .. & .. & a_0 b_n \\ \dots & \dots & .. & .. & .. \\ a_{m-n}b_n - a_m b_0 & a_{m-n-1}b_n + a_{m-n}b_{n-1} - a_{m-1}b_0 & .. & .. & a_0 b_1 \\ b_n & b_{n-1} & b_0 & & \\ & b_n & .. & b_0 & \\ & \dots & .. & .. & \dots \\ & & .. & .. & b_0 \end{bmatrix} \tag{42}$$

where the unfilled entries are 0's.

- The criterion for a *non-trivial* common factor is $\det[BM] = 0$.
- If $m = n$, then in equations (38) - (40), a set of n 'linearly independent equations' in n unknowns $x^{n-1}, \dots, x^0$ are already available.
- Solve for the unknowns by standard linear algebra techniques.

- The Bézout matrix is given as

$$\begin{bmatrix} a_{m-1}b_n - a_m b_{n-1} & a_{m-2}b_n - a_m b_{n-2} & .. & .. & a_0 b_n \\ ... & ... & .. & .. & .. \\ a_{m-n}b_n - a_m b_0 & a_{m-n-1}b_n + a_{m-n}b_{n-1} - a_{m-1}b_0 & .. & .. & a_0 b_1 \\ b_n & b_{n-1} & b_0 & & \\ & b_n & .. & b_0 & \\ & ... & .. & .. & ... \\ & & .. & .. & b_0 \end{bmatrix} \tag{42}$$

where the unfilled entries are 0's.

- The criterion for a *non-trivial* common factor is $\det[BM] = 0$.
- If $m = n$, then in equations (38) - (40), a set of n 'linearly independent equations' in n unknowns $x^{n-1}, ..., x^0$ are already available.
- Solve for the unknowns by standard linear algebra techniques.

Elimination Theory

Bézouts Matrix (contd.)

- The Bézout matrix is given as

$$\begin{bmatrix} a_{m-1}b_n - a_m b_{n-1} & a_{m-2}b_n - a_m b_{n-2} & .. & .. & a_0 b_n \\ \cdots & \cdots & .. & .. & .. \\ a_{m-n}b_n - a_m b_0 & a_{m-n-1}b_n + a_{m-n}b_{n-1} - a_{m-1}b_0 & .. & .. & a_0 b_1 \\ b_n & b_{n-1} & b_0 & & \\ & b_n & .. & b_0 & \\ & \cdots & .. & .. & \cdots \\ & & .. & .. & b_0 \end{bmatrix} \quad (42)$$

  where the unfilled entries are 0's.

- The criterion for a *non-trivial* common factor is $\det[BM] = 0$.
- If $m = n$, then in equations (38) - (40), a set of n 'linearly independent equations' in n unknowns $x^{n-1}, \ldots, x^0$ are already available.
- Solve for the unknowns by standard linear algebra techniques.

- The Bézout matrix is given as

$$\begin{bmatrix} a_{m-1}b_n - a_m b_{n-1} & a_{m-2}b_n - a_m b_{n-2} & .. & .. & a_0 b_n \\ \dots & \dots & .. & .. & .. \\ a_{m-n}b_n - a_m b_0 & a_{m-n-1}b_n + a_{m-n}b_{n-1} - a_{m-1}b_0 & .. & .. & a_0 b_1 \\ b_n & b_{n-1} & b_0 & & \\ & b_n & .. & b_0 & \\ & \dots & .. & .. & \dots \\ & & .. & .. & b_0 \end{bmatrix} \tag{42}$$

where the unfilled entries are 0's.

- The criterion for a *non-trivial* common factor is $\det[BM] = 0$.
- If $m = n$, then in equations (38) - (40), a set of n 'linearly independent equations' in n unknowns $x^{n-1}, \dots, x^0$ are already available.
- Solve for the unknowns by standard linear algebra techniques.

- Consider two cubics of the form

$$
\begin{aligned}
a_3x^3 + a_2x^2 + a_1x + a_0 &= 0 \\
b_3x^3 + b_2x^2 + b_1x + b_0 &= 0
\end{aligned}
\tag{43}
$$

- The Bézout matrix is given by

$$
[BM] = \begin{bmatrix} b_3a_2 - a_3b_2 & b_3a_1 - a_3b_1 & b_3a_0 - a_3b_0 \\ b_3a_1 - a_3b_1 & (b_3a_0 - a_3b_0) + (b_2a_1 - a_2b_1) & b_2a_0 - a_2b_0 \\ b_3a_0 - a_0b_3 & b_2a_0 - a_2b_0 & b_1a_0 - a_1b_0 \end{bmatrix}
\tag{44}
$$

- Note: $[BM]$ is 3×3 while $[SM]$ would be 6×6 for this case.

- Consider two cubics of the form

$$\begin{aligned} a_3x^3 + a_2x^2 + a_1x + a_0 &= 0 \\ b_3x^3 + b_2x^2 + b_1x + b_0 &= 0 \end{aligned} \quad (43)$$

- The Bézout matrix is given by

$$[BM] = \begin{bmatrix} b_3a_2 - a_3b_2 & b_3a_1 - a_3b_1 & b_3a_0 - a_3b_0 \\ b_3a_1 - a_3b_1 & (b_3a_0 - a_3b_0) + (b_2a_1 - a_2b_1) & b_2a_0 - a_2b_0 \\ b_3a_0 - a_0b_3 & b_2a_0 - a_2b_0 & b_1a_0 - a_1b_0 \end{bmatrix} \quad (44)$$

- Note: $[BM]$ is 3×3 while $[SM]$ would be 6×6 for this case.

- Consider two cubics of the form

$$\begin{aligned} a_3x^3 + a_2x^2 + a_1x + a_0 &= 0 \\ b_3x^3 + b_2x^2 + b_1x + b_0 &= 0 \end{aligned} \tag{43}$$

- The Bézout matrix is given by

$$[BM] = \begin{bmatrix} b_3a_2 - a_3b_2 & b_3a_1 - a_3b_1 & b_3a_0 - a_3b_0 \\ b_3a_1 - a_3b_1 & (b_3a_0 - a_3b_0) + (b_2a_1 - a_2b_1) & b_2a_0 - a_2b_0 \\ b_3a_0 - a_0b_3 & b_2a_0 - a_2b_0 & b_1a_0 - a_1b_0 \end{bmatrix} \tag{44}$$

- Note: $[BM]$ is 3×3 while $[SM]$ would be 6×6 for this case.

- Intuitively – Bézout matrix and Sylvester matrix should be related as no new information is possible!
- Example: Consider $P(x) = a_3x^3 + a_2x^2 + a_1x + a_0 = 0$ and $Q(x) = b_2x^2 + b_1x + b_0 = 0$.
- Sylvester's matrix is given by

$$[SM] = \begin{bmatrix} a_3 & a_2 & a_1 & a_0 & 0 \\ 0 & a_3 & a_2 & a_1 & a_0 \\ b_2 & b_1 & b_0 & 0 & 0 \\ 0 & b_2 & b_1 & b_0 & 0 \\ 0 & 0 & b_2 & b_1 & b_0 \end{bmatrix} \tag{45}$$

- The Bézout matrix is given as

$$[BM] = \begin{bmatrix} a_1b_2 - a_3b_0 & a_0b_2 + a_1b_1 - a_2b_0 & a_0b_1 \\ a_2b_2 - a_3b_1 & a_1b_2 - a_3b_0 & a_0b_2 \\ b_2 & b_1 & b_0 \end{bmatrix} \tag{46}$$

- Intuitively – Bézout matrix and Sylvester matrix should be related as no new information is possible!
- Example: Consider $P(x) = a_3x^3 + a_2x^2 + a_1x + a_0 = 0$ and $Q(x) = b_2x^2 + b_1x + b_0 = 0$.
- Sylvester's matrix is given by

$$[SM] = \begin{bmatrix} a_3 & a_2 & a_1 & a_0 & 0 \\ 0 & a_3 & a_2 & a_1 & a_0 \\ b_2 & b_1 & b_0 & 0 & 0 \\ 0 & b_2 & b_1 & b_0 & 0 \\ 0 & 0 & b_2 & b_1 & b_0 \end{bmatrix} \tag{45}$$

- The Bézout matrix is given as

$$[BM] = \begin{bmatrix} a_1b_2 - a_3b_0 & a_0b_2 + a_1b_1 - a_2b_0 & a_0b_1 \\ a_2b_2 - a_3b_1 & a_1b_2 - a_3b_0 & a_0b_2 \\ b_2 & b_1 & b_0 \end{bmatrix} \tag{46}$$

- Intuitively – Bézout matrix and Sylvester matrix should be related as no new information is possible!
- Example: Consider $P(x) = a_3x^3 + a_2x^2 + a_1x + a_0 = 0$ and $Q(x) = b_2x^2 + b_1x + b_0 = 0$.
- Sylvester's matrix is given by

$$[SM] = \begin{bmatrix} a_3 & a_2 & a_1 & a_0 & 0 \\ 0 & a_3 & a_2 & a_1 & a_0 \\ b_2 & b_1 & b_0 & 0 & 0 \\ 0 & b_2 & b_1 & b_0 & 0 \\ 0 & 0 & b_2 & b_1 & b_0 \end{bmatrix} \tag{45}$$

- The Bézout matrix is given as

$$[BM] = \begin{bmatrix} a_1b_2 - a_3b_0 & a_0b_2 + a_1b_1 - a_2b_0 & a_0b_1 \\ a_2b_2 - a_3b_1 & a_1b_2 - a_3b_0 & a_0b_2 \\ b_2 & b_1 & b_0 \end{bmatrix} \tag{46}$$

Elimination Theory

Equivalence of $[BM]$ and $[SM]$

- Intuitively – Bézout matrix and Sylvester matrix should be related as no new information is possible!
- Example: Consider $P(x) = a_3x^3 + a_2x^2 + a_1x + a_0 = 0$ and $Q(x) = b_2x^2 + b_1x + b_0 = 0$.
- Sylvester's matrix is given by

$$[SM] = \begin{bmatrix} a_3 & a_2 & a_1 & a_0 & 0 \\ 0 & a_3 & a_2 & a_1 & a_0 \\ b_2 & b_1 & b_0 & 0 & 0 \\ 0 & b_2 & b_1 & b_0 & 0 \\ 0 & 0 & b_2 & b_1 & b_0 \end{bmatrix} \tag{45}$$

- The Bézout matrix is given as

$$[BM] = \begin{bmatrix} a_1b_2 - a_3b_0 & a_0b_2 + a_1b_1 - a_2b_0 & a_0b_1 \\ a_2b_2 - a_3b_1 & a_1b_2 - a_3b_0 & a_0b_2 \\ b_2 & b_1 & b_0 \end{bmatrix} \tag{46}$$

Elimination Theory

Equivalence of $[BM]$ and $[SM]$ - Contd.

- Pre-multiply $[SM]$ by matrix

$$[A] = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ b_2 & b_1 & -a_3 & -a_2 & 0 \\ 0 & b_2 & 0 & -a_3 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (47)$$

to get

$$[B] = \begin{bmatrix} a_3 & a_2 & a_1 & a_0 & 0 \\ 0 & a_3 & a_2 & a_1 & a_0 \\ 0 & 0 & a_1 b_2 - a_3 b_0 & a_0 b_2 + a_1 b_1 - a_2 b_0 & a_0 b_1 \\ 0 & 0 & a_2 b_2 - a_3 b_1 & a_1 b_2 - a_3 b_0 & a_0 b_2 \\ 0 & 0 & b_2 & b_1 & b_0 \end{bmatrix} \quad (48)$$

- Observe that $\det[A]\det[SM] = (a_3)^2 \det[SM] = \det[B] = (a_3)^2 \det[BM]$ which shows that $\det[SM] = \det[BM]$.

Elimination Theory

Equivalence of $[BM]$ and $[SM]$ - Contd.

- Pre-multiply $[SM]$ by matrix

$$[A] = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ b_2 & b_1 & -a_3 & -a_2 & 0 \\ 0 & b_2 & 0 & -a_3 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (47)$$

to get

$$[B] = \begin{bmatrix} a_3 & a_2 & a_1 & a_0 & 0 \\ 0 & a_3 & a_2 & a_1 & a_0 \\ 0 & 0 & a_1 b_2 - a_3 b_0 & a_0 b_2 + a_1 b_1 - a_2 b_0 & a_0 b_1 \\ 0 & 0 & a_2 b_2 - a_3 b_1 & a_1 b_2 - a_3 b_0 & a_0 b_2 \\ 0 & 0 & b_2 & b_1 & b_0 \end{bmatrix} \quad (48)$$

- Observe that $\det[A]\det[SM] = (a_3)^2 \det[SM] = \det[B] = (a_3)^2 \det[BM]$ which shows that $\det[SM] = \det[BM]$.

Outline

- General 6R robot: No constant D-H link parameters have special values, such as 0, $\pi/2$, or π.
- Special D-H values (such as in PUMA 560) result in easier elimination.
- If Prismatic joint is present $\rightarrow$ elimination is easier.
- Inverse kinematics of a general 6R was *unsolved* for a long time.
  - Several researchers worked on it – Duffy and Crane (1980) first derived a 32^{nd} order polynomial in one joint angle.
  - Eventually Raghavan and Roth (1993) derived a 16^{th} degree polynomial in one joint angle.
- Follow the development in Raghavan and Roth (1993) & Extensive use elimination theory discussed in Lecture 3.
- The direct kinematics equations for a general 6R manipulator is

$$ {}^0_6[T] = {}^0_1[T]\,{}^1_2[T]\,{}^2_3[T]\,{}^3_4[T]\,{}^4_5[T]\,{}^5_6[T] \tag{49}$$

- General 6R robot: No constant D-H link parameters have special values, such as 0, $\pi/2$, or π.
- Special D-H values (such as in PUMA 560) result in easier elimination.
- If Prismatic joint is present $\rightarrow$ elimination is easier.
- Inverse kinematics of a general 6R was *unsolved* for a long time.
  - Several researchers worked on it – Duffy and Crane (1980) first derived a 32^{nd} order polynomial in one joint angle.
  - Eventually Raghavan and Roth (1993) derived a 16^{th} degree polynomial in one joint angle.
- Follow the development in Raghavan and Roth (1993) & Extensive use elimination theory discussed in Lecture 3.
- The direct kinematics equations for a general 6R manipulator is

$$ {}^0_6[T] = {}^0_1[T]\,{}^1_2[T]\,{}^2_3[T]\,{}^3_4[T]\,{}^4_5[T]\,{}^5_6[T] \tag{49} $$

- General 6R robot: No constant D-H link parameters have special values, such as 0, $\pi/2$, or π.
- Special D-H values (such as in PUMA 560) result in easier elimination.
- If Prismatic joint is present $\rightarrow$ elimination is easier.
- Inverse kinematics of a general 6R was *unsolved* for a long time.
  - Several researchers worked on it – Duffy and Crane (1980) first derived a 32^{nd} order polynomial in one joint angle.
  - Eventually Raghavan and Roth (1993) derived a 16^{th} degree polynomial in one joint angle.
- Follow the development in Raghavan and Roth (1993) & Extensive use elimination theory discussed in Lecture 3.
- The direct kinematics equations for a general 6R manipulator is

$$ {}^0_6[T] = {}^0_1[T]{}^1_2[T]{}^2_3[T]{}^3_4[T]{}^4_5[T]{}^5_6[T] \tag{49} $$

- General 6R robot: No constant D-H link parameters have special values, such as 0, $\pi/2$, or π.
- Special D-H values (such as in PUMA 560) result in easier elimination.
- If Prismatic joint is present $\rightarrow$ elimination is easier.
- Inverse kinematics of a general 6R was *unsolved* for a long time.
  - Several researchers worked on it – Duffy and Crane (1980) first derived a 32^{nd} order polynomial in one joint angle.
  - Eventually Raghavan and Roth (1993) derived a 16^{th} degree polynomial in one joint angle.
- Follow the development in Raghavan and Roth (1993) & Extensive use elimination theory discussed in Lecture 3.
- The direct kinematics equations for a general 6R manipulator is

$$ {}^0_6[T] = {}^0_1[T]\,{}^1_2[T]\,{}^2_3[T]\,{}^3_4[T]\,{}^4_5[T]\,{}^5_6[T] \tag{49} $$

Inverse Kinematics of a General 6R Robot

- Recall with respect to equation (49)
  - ${}^{i-1}_{i}[T]$ is a function of *only one* joint variable θ_i and *three D-H parameters which are constants*.
  - For IK problem, ${}^{0}_{6}[T]$ is given $\rightarrow$ find the six joint variables in each of ${}^{i-1}_{i}[T],\ i = 1, 2, ..., 6$.
- **Step 1**: write ${}^{i-1}_{i}[T]$ as product of two matrices $({}^{i-1}_{i}[T])_{st}({}^{i-1}_{i}[T])_{jt}$.

$$
\begin{aligned}
{}^{i-1}_{i}[T] &= ({}^{i-1}_{i}[T])_{st}({}^{i-1}_{i}[T])_{jt} \\
&= \begin{pmatrix} 1 & 0 & 0 & a_{i-1} \\ 0 & c_{\alpha_{i-1}} & -s_{\alpha_{i-1}} & 0 \\ 0 & s_{\alpha_{i-1}} & c_{\alpha_{i-1}} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} c_{\theta_i} & -s_{\theta_i} & 0 & 0 \\ s_{\theta_i} & c_{\theta_i} & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (50)
\end{aligned}
$$

- The matrix $({}^{i-1}_{i}[T])_{st}$ is constant
- The matrix $({}^{i-1}_{i}[T])_{jt}$ is a function of the joint variable θ_i (for a rotary joint) or d_i (for a prismatic joint).

Inverse Kinematics of a General 6R Robot

- Recall with respect to equation (49)
  - ${}^{i-1}_{i}[T]$ is a function of *only one* joint variable θ_i and *three D-H parameters which are constants*.
  - For IK problem, ${}^{0}_{6}[T]$ is given $\rightarrow$ find the six joint variables in each of ${}^{i-1}_{i}[T]$, $i = 1, 2, ..., 6$.
- **Step 1**: write ${}^{i-1}_{i}[T]$ as product of two matrices $({}^{i-1}_{i}[T])_{st}({}^{i-1}_{i}[T])_{jt}$.

$$
\begin{aligned}
{}^{i-1}_{i}[T] &= ({}^{i-1}_{i}[T])_{st}({}^{i-1}_{i}[T])_{jt} \\
&= \begin{pmatrix} 1 & 0 & 0 & a_{i-1} \\ 0 & c_{\alpha_{i-1}} & -s_{\alpha_{i-1}} & 0 \\ 0 & s_{\alpha_{i-1}} & c_{\alpha_{i-1}} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} c_{\theta_i} & -s_{\theta_i} & 0 & 0 \\ s_{\theta_i} & c_{\theta_i} & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad (50)
\end{aligned}
$$

- The matrix $({}^{i-1}_{i}[T])_{st}$ is constant
- The matrix $({}^{i-1}_{i}[T])_{jt}$ is a function of the joint variable θ_i (for a rotary joint) or d_i (for a prismatic joint).

- Recall with respect to equation (49)
  - ${}_{i}^{i-1}[T]$ is a function of *only one* joint variable θ_i and *three D-H parameters which are constants*.
  - For IK problem, ${}_{6}^{0}[T]$ is given $\rightarrow$ find the six joint variables in each of ${}_{i}^{i-1}[T],\ i = 1, 2, ..., 6$.
- **Step 1**: write ${}_{i}^{i-1}[T]$ as product of two matrices $({}_{i}^{i-1}[T])_{st}({}_{i}^{i-1}[T])_{jt}$.

$$
\begin{aligned}
{}_{i}^{i-1}[T] &= ({}_{i}^{i-1}[T])_{st}({}_{i}^{i-1}[T])_{jt} \\
&= \begin{pmatrix} 1 & 0 & 0 & a_{i-1} \\ 0 & c_{\alpha_{i-1}} & -s_{\alpha_{i-1}} & 0 \\ 0 & s_{\alpha_{i-1}} & c_{\alpha_{i-1}} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} c_{\theta_i} & -s_{\theta_i} & 0 & 0 \\ s_{\theta_i} & c_{\theta_i} & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{pmatrix}
\end{aligned} \tag{50}
$$

- The matrix $({}_{i}^{i-1}[T])_{st}$ is constant
- The matrix $({}_{i}^{i-1}[T])_{jt}$ is a function of the joint variable θ_i (for a rotary joint) or d_i (for a prismatic joint).

Step 2: Reorganize equation of direct kinematics

- Rewrite equation (49) as

$$({}^{2}_{3}[T])_{jt}\ {}^{3}_{4}[T]{}^{4}_{5}[T]({}^{5}_{6}[T])_{st} = ({}^{2}_{3}[T])^{-1}_{st}\ ({}^{1}_{2}[T])^{-1}({}^{0}_{1}[T])^{-1}\ {}^{0}_{6}[T]({}^{5}_{6}[T])^{-1}_{jt} \quad (51)$$

  - The left-hand side is only a function of $(\theta_3, \theta_4, \theta_5)$
  - The right-hand side is only a function of $(\theta_1, \theta_2, \theta_6)$.
  - Six scalar equations obtained by equating the top three elements of columns 3 and 4 on both sides of equation (51) do not contain θ_6.

$$\begin{aligned}[A](s_4s_5\ s_4c_5\ c_4s_5\ c_4c_5\ s_4\ c_4\ s_5\ c_5\ 1)^T \\ = [B](s_1s_2\ s_1c_2\ c_1s_2\ c_1c_2\ s_1\ c_1\ s_2\ c_2)^T\end{aligned} \quad (52)$$

  where $[A]$ is 6×9 matrix whose elements are linear in s_3, c_3, 1, and $[B]$ is 6×8 matrix of constants.
  - Denote columns 3 and 4 by **p** and **l**

Step 3: Eliminate four of five variables, $\theta_i, \; i = 1, .., 5$ in equation (52).

- *Key step* is to obtain minimal set of equations.
- The minimal set of equations is 14 (Raghavan & Roth, 1993).
  - Three equations from $\mathbf{p}$.
  - Three equations from $\mathbf{l}$.
  - One scalar equation from the scalar dot product $\mathbf{p} \cdot \mathbf{p}$
  - One scalar equation from the scalar dot product $\mathbf{p} \cdot \mathbf{l}$
  - Three equations from the vector cross product $\mathbf{p} \times \mathbf{l}$
  - Three scalar equations from $(\mathbf{p} \cdot \mathbf{p})\mathbf{l} - (2\mathbf{p} \cdot \mathbf{l})\mathbf{p}$.
- The 14 equations can be written as

$$[P](s_4 s_5 \;\; s_4 c_5 \;\; c_4 s_5 \;\; c_4 c_5 \;\; s_4 \;\; c_4 \;\; s_5 \;\; c_5 \;\; 1)^T = [Q](s_1 s_2 \;\; s_1 c_2 \;\; c_1 s_2 \;\; c_1 c_2 \;\; s_1 \;\; c_1 \;\; s_2 \;\; c_2)^T \quad (53)$$

$[P]$ is a 14×9 matrix whose elements are linear in c_3, s_3, 1, and $[Q]$ is a 14×8 matrix of constants.

Step 3: Eliminate four of five variables, θ_i, $i = 1, .., 5$ in equation (52).

- *Key step* is to obtain minimal set of equations.
- The minimal set of equations is 14 (Raghavan & Roth, 1993).
  - Three equations from $\mathbf{p}$.
  - Three equations from $\mathbf{l}$.
  - One scalar equation from the scalar dot product $\mathbf{p} \cdot \mathbf{p}$
  - One scalar equation from the scalar dot product $\mathbf{p} \cdot \mathbf{l}$
  - Three equations from the vector cross product $\mathbf{p} \times \mathbf{l}$
  - Three scalar equations from $(\mathbf{p} \cdot \mathbf{p})\mathbf{l} - (2\mathbf{p} \cdot \mathbf{l})\mathbf{p}$.
- The 14 equations can be written as

$$[P](s_4 s_5 \;\; s_4 c_5 \;\; c_4 s_5 \;\; c_4 c_5 \;\; s_4 \;\; c_4 \;\; s_5 \;\; c_5 \;\; 1)^T = [Q](s_1 s_2 \;\; s_1 c_2 \;\; c_1 s_2 \;\; c_1 c_2 \;\; s_1 \;\; c_1 \;\; s_2 \;\; c_2)^T \tag{53}$$

$[P]$ is a 14×9 matrix whose elements are linear in c_3, s_3, 1, and $[Q]$ is a 14×8 matrix of constants.

Step 3: Eliminate four of five variables, θ_i, $i = 1,..,5$ in equation (52).

- *Key step* is to obtain minimal set of equations.
- The minimal set of equations is 14 (Raghavan & Roth, 1993).
  - Three equations from $\mathbf{p}$.
  - Three equations from $\mathbf{l}$.
  - One scalar equation from the scalar dot product $\mathbf{p} \cdot \mathbf{p}$
  - One scalar equation from the scalar dot product $\mathbf{p} \cdot \mathbf{l}$
  - Three equations from the vector cross product $\mathbf{p} \times \mathbf{l}$
  - Three scalar equations from $(\mathbf{p} \cdot \mathbf{p})\mathbf{l} - (2\mathbf{p} \cdot \mathbf{l})\mathbf{p}$.
- The 14 equations can be written as

$$\begin{aligned} &[P](s_4 s_5 \ s_4 c_5 \ c_4 s_5 \ c_4 c_5 \ s_4 \ c_4 \ s_5 \ c_5 \ 1)^T \\ &= [Q](s_1 s_2 \ s_1 c_2 \ c_1 s_2 \ c_1 c_2 \ s_1 \ c_1 \ s_2 \ c_2)^T \end{aligned} \quad (53)$$

$[P]$ is a 14×9 matrix whose elements are linear in c_3, s_3, 1, and $[Q]$ is a 14×8 matrix of constants.

Step 3: Elimination of four θ_i (Contd.)

- First use any eight of the 14 equations in equation (53) and solve for the eight variables $s_1 s_2, s_1 c_2, c_1 s_2, c_1 c_2, s_1, c_1, s_2, c_2$.
- Always possible to solve eight linear equations in eight unknowns.
- Substitute the eight variables in the rest of the six equations to get

$$[R] (s_4 s_5 \ \ s_4 c_5 \ \ c_4 s_5 \ \ c_4 c_5 \ \ s_4 \ \ c_4 \ \ s_5 \ \ c_5 \ \ 1)^T = \mathbf{0} \tag{54}$$

$[R]$ is a 6×9 matrix whose elements are linear in s_3 and c_3.

Step 3: Elimination of four θ_i (Contd.)

- First use any eight of the 14 equations in equation (53) and solve for the eight variables $s_1 s_2, s_1 c_2, c_1 s_2, c_1 c_2, s_1, c_1, s_2, c_2$.
- Always possible to solve eight linear equations in eight unknowns.
- Substitute the eight variables in the rest of the six equations to get

$$[R]\left(s_4 s_5 \;\; s_4 c_5 \;\; c_4 s_5 \;\; c_4 c_5 \;\; s_4 \;\; c_4 \;\; s_5 \;\; c_5 \;\; 1\right)^T = \mathbf{0} \qquad (54)$$

$[R]$ is a 6×9 matrix whose elements are linear in s_3 and c_3.

Step 3: Elimination of four θ_i (Contd.)

- First use any eight of the 14 equations in equation (53) and solve for the eight variables $s_1 s_2, s_1 c_2, c_1 s_2, c_1 c_2, s_1, c_1, s_2, c_2$.
- Always possible to solve eight linear equations in eight unknowns.
- Substitute the eight variables in the rest of the six equations to get

$$[R]\,(s_4 s_5 \;\; s_4 c_5 \;\; c_4 s_5 \;\; c_4 c_5 \;\; s_4 \;\; c_4 \;\; s_5 \;\; c_5 \;\; 1)^T = \mathbf{0} \tag{54}$$

$[R]$ is a 6×9 matrix whose elements are linear in s_3 and c_3.

Step 4: Elimination of θ_4 and θ_5

- Use tangent half-angle formulas for s_3, c_3, s_4, c_4, s_5, and c_5.
- On simplifying get

$$[S]\left(x_4^2x_5^2\ x_4^2x_5\ x_4^2\ x_4x_5^2\ x_4x_5\ x_4\ x_5^2\ x_5\ 1\right)^T = \mathbf{0} \qquad (55)$$

  where $[S]$ is a 6×9 matrix and $x_{(\cdot)} = \tan(\frac{\theta_{\cdot}}{2})$.
- Eliminate x_4 and x_5 using Sylvester's dialytic method.
  - Six additional equations are generated by multiplying equations in equation (55) by x_4.
  - Three additional 'linearly' independent variables, namely, $x_4^3x_5^2$, $x_4^3x_5$, and x_4^3, are generated.
  - A system of 12 equations in 12 unknowns.

Step 4: Elimination of θ_4 and θ_5

- Use tangent half-angle formulas for s_3, c_3, s_4, c_4, s_5, and c_5.
- On simplifying get

$$[S]\left(x_4^2 x_5^2 \; x_4^2 x_5 \; x_4^2 \; x_4 x_5^2 \; x_4 x_5 \; x_4 \; x_5^2 \; x_5 \; 1\right)^T = \mathbf{0} \tag{55}$$

where $[S]$ is a 6×9 matrix and $x_{(\cdot)} = \tan(\frac{\theta_{\cdot}}{2})$.

- Eliminate x_4 and x_5 using Sylvester's dialytic method.
  - Six additional equations are generated by multiplying equations in equation (55) by x_4.
  - Three additional 'linearly' independent variables, namely, $x_4^3 x_5^2$, $x_4^3 x_5$, and x_4^3, are generated.
  - A system of 12 equations in 12 unknowns.

Step 4: Elimination of θ_4 and θ_5

- Use tangent half-angle formulas for s_3, c_3, s_4, c_4, s_5, and c_5.
- On simplifying get

$$[S]\left(x_4^2x_5^2 \ x_4^2x_5 \ x_4^2 \ x_4x_5^2 \ x_4x_5 \ x_4 \ x_5^2 \ x_5 \ 1\right)^T = \mathbf{0} \qquad (55)$$

where $[S]$ is a 6×9 matrix and $x_{(\cdot)} = \tan(\frac{\theta_{\cdot}}{2})$.

- Eliminate x_4 and x_5 using Sylvester's dialytic method.
  - Six additional equations are generated by multiplying equations in equation (55) by x_4.
  - Three additional 'linearly' independent variables, namely, $x_4^3x_5^2$, $x_4^3x_5$, and x_4^3, are generated.
  - A system of 12 equations in 12 unknowns.

Step 4: Elimination of θ_4 and θ_5 (Contd.)

- The 12 equations can be written as

$$\begin{pmatrix} S & 0 \\ 0 & S \end{pmatrix} \begin{pmatrix} x_4^3 x_5^2 \\ x_4^3 x_5 \\ x_4^3 \\ x_4^2 x_5^2 \\ x_4^2 x_5 \\ x_4^2 \\ x_4 x_5^2 \\ x_4 x_5 \\ x_4 \\ x_5^2 \\ x_5 \\ 1 \end{pmatrix} = \mathbf{0} \tag{56}$$

- Following Sylvester's method, set $\det \begin{pmatrix} S & 0 \\ 0 & S \end{pmatrix} = 0$
- On simplification, a 16th-degree polynomial in x_3 is obtained.
- Solve for roots of this polynomial and find $\theta_3 = 2\tan^{-1}(x_3)$.

Step 5: Obtain other joint angles

- Once θ_3 is known, find x_4 and x_5 from equation (56) using standard linear algebra.
- From x_4 and x_5 find θ_4 and θ_5.
- Once θ_3, θ_4, and θ_5 are known, solve $s_1 s_2$, $s_1 c_2$, ..., s_2, c_2 from eight linearly independent equations (53).
- Obtain unique θ_1 and θ_2.
- To obtain θ_6, rewrite equation (49) as

$$ {}^{5}_{6}[T] = {}^{4}_{5}[T]^{-1} \, {}^{3}_{4}[T]^{-1} \, {}^{2}_{3}[T]^{-1} \, {}^{1}_{2}[T]^{-1} \, {}^{0}_{1}[T]^{-1} \, {}^{0}_{6}[T] \qquad (57) $$

- $\theta_i, i = 1, 2, ..., 5$ are known $\rightarrow$ (1,1) and (2,1) elements gives two equations in s_6 and c_6 $\rightarrow$ unique value of θ_6.

Step 5: Obtain other joint angles

- Once θ_3 is known, find x_4 and x_5 from equation (56) using standard linear algebra.
- From x_4 and x_5 find θ_4 and θ_5.
- Once θ_3, θ_4, and θ_5 are known, solve $s_1 s_2$, $s_1 c_2$, ..., s_2, c_2 from eight linearly independent equations (53).
- Obtain unique θ_1 and θ_2.
- To obtain θ_6, rewrite equation (49) as

$$ {}^5_6[T] = {}^4_5[T]^{-1} {}^3_4[T]^{-1} {}^2_3[T]^{-1} {}^1_2[T]^{-1} {}^0_1[T]^{-1} {}^0_6[T] \tag{57} $$

- $\theta_i, i = 1, 2, ..., 5$ are known $\rightarrow$ (1,1) and (2,1) elements gives two equations in s_6 and c_6 $\rightarrow$ unique value of θ_6.

Step 5: Obtain other joint angles

- Once θ_3 is known, find x_4 and x_5 from equation (56) using standard linear algebra.
- From x_4 and x_5 find θ_4 and θ_5.
- Once θ_3, θ_4, and θ_5 are known, solve $s_1 s_2$, $s_1 c_2$, ..., s_2, c_2 from eight linearly independent equations (53).
- Obtain unique θ_1 and θ_2.
- To obtain θ_6, rewrite equation (49) as

$$ {}^{5}_{6}[T] = {}^{4}_{5}[T]^{-1}\,{}^{3}_{4}[T]^{-1}\,{}^{2}_{3}[T]^{-1}\,{}^{1}_{2}[T]^{-1}\,{}^{0}_{1}[T]^{-1}\,{}^{0}_{6}[T] \tag{57} $$

- $\theta_i, i = 1, 2, ..., 5$ are known $\rightarrow$ (1,1) and (2,1) elements gives two equations in s_6 and c_6 $\rightarrow$ unique value of θ_6.

Step 5: Obtain other joint angles

- Once θ_3 is known, find x_4 and x_5 from equation (56) using standard linear algebra.
- From x_4 and x_5 find θ_4 and θ_5.
- Once θ_3, θ_4, and θ_5 are known, solve $s_1 s_2$, $s_1 c_2$, ..., s_2, c_2 from eight linearly independent equations (53).
- Obtain unique θ_1 and θ_2.
- To obtain θ_6, rewrite equation (49) as

$$ {}^5_6[T] = {}^4_5[T]^{-1} \, {}^3_4[T]^{-1} \, {}^2_3[T]^{-1} \, {}^1_2[T]^{-1} \, {}^0_1[T]^{-1} \, {}^0_6[T] \tag{57} $$

- $\theta_i, i = 1, 2, ..., 5$ are known $\rightarrow$ (1,1) and (2,1) elements gives two equations in s_6 and c_6 $\rightarrow$ unique value of θ_6.

Inverse Kinematics of a General 6R Robot

Summary

- A sixteenth degree polynomial in x_3 is obtained in **Step 4** $\rightarrow$ general 6R serial manipulator has 16 possible solutions.
- A 6R manipulator with special geometry $\rightarrow$ 16th-degree polynomial in x_3 can be of lower order.
- If one or more joints are prismatic $\rightarrow$ Inverse kinematics becomes simpler since the prismatic joint variable is not in terms of sines or cosines.
- Not possible to find general expression for workspace boundary since closed-form solution for 16th-degree polynomial not possible.
- Check: If *all the roots* of the 16th-degree polynomial are *complex*, then ${}^0_6[T]$ is *not in the workspace* of the manipulator.
- All the inverse kinematics solutions & entire workspace may not be available due to the presence of joint limits and limitations of hardware (see, Rastegar and Deravi, 1987 & Dwarakanath et al. 1992).

Inverse Kinematics of a General 6R Robot

Summary

- A sixteenth degree polynomial in x_3 is obtained in **Step 4** $\rightarrow$ general 6R serial manipulator has 16 possible solutions.
- A 6R manipulator with special geometry $\rightarrow$ 16th-degree polynomial in x_3 can be of lower order.
- If one or more joints are prismatic $\rightarrow$ Inverse kinematics becomes simpler since the prismatic joint variable is not in terms of sines or cosines.
- Not possible to find general expression for workspace boundary since closed-form solution for 16th-degree polynomial not possible.
- Check: If *all the roots* of the 16th-degree polynomial are *complex*, then ${}^0_6[T]$ is *not in the workspace* of the manipulator.
- All the inverse kinematics solutions & entire workspace may not be available due to the presence of joint limits and limitations of hardware (see, Rastegar and Deravi, 1987 & Dwarakanath et al. 1992).

- A sixteenth degree polynomial in x_3 is obtained in **Step 4** $\rightarrow$ general 6R serial manipulator has 16 possible solutions.
- A 6R manipulator with special geometry $\rightarrow$ 16th-degree polynomial in x_3 can be of lower order.
- If one or more joints are prismatic $\rightarrow$ Inverse kinematics becomes simpler since the prismatic joint variable is not in terms of sines or cosines.
- Not possible to find general expression for workspace boundary since closed-form solution for 16th-degree polynomial not possible.
- Check: If *all the roots* of the 16th-degree polynomial are *complex*, then ${}^0_6[T]$ is *not in the workspace* of the manipulator.
- All the inverse kinematics solutions & entire workspace may not be available due to the presence of joint limits and limitations of hardware (see, Rastegar and Deravi, 1987 & Dwarakanath et al. 1992).

- A sixteenth degree polynomial in x_3 is obtained in **Step 4** $\rightarrow$ general 6R serial manipulator has 16 possible solutions.
- A 6R manipulator with special geometry $\rightarrow$ 16th-degree polynomial in x_3 can be of lower order.
- If one or more joints are prismatic $\rightarrow$ Inverse kinematics becomes simpler since the prismatic joint variable is not in terms of sines or cosines.
- Not possible to find general expression for workspace boundary since closed-form solution for 16th-degree polynomial not possible.
- Check: If *all the roots* of the 16th-degree polynomial are *complex*, then ${}^0_6[T]$ is *not in the workspace* of the manipulator.
- All the inverse kinematics solutions & entire workspace may not be available due to the presence of joint limits and limitations of hardware (see, Rastegar and Deravi, 1987 & Dwarakanath et al. 1992).

- A sixteenth degree polynomial in x_3 is obtained in **Step 4** $\rightarrow$ general 6R serial manipulator has 16 possible solutions.
- A 6R manipulator with special geometry $\rightarrow$ 16th-degree polynomial in x_3 can be of lower order.
- If one or more joints are prismatic $\rightarrow$ Inverse kinematics becomes simpler since the prismatic joint variable is not in terms of sines or cosines.
- Not possible to find general expression for workspace boundary since closed-form solution for 16th-degree polynomial not possible.
- Check: If *all the roots* of the 16th-degree polynomial are *complex*, then ${}^0_6[T]$ is *not in the workspace* of the manipulator.
- All the inverse kinematics solutions & entire workspace may not be available due to the presence of joint limits and limitations of hardware (see, Rastegar and Deravi, 1987 & Dwarakanath et al. 1992).

Outline

Module 3 – Additional Material

- Exercise Problems
- References & Suggested Reading