大学院講義

「先端機械要素」

東京工業大学
工学院機械系　　教授

イワツキ　ノブユキ
岩附　信行

Lecture for Graduate Students

Advanced Mechanical Elements

Dept. of Mechanical Engineering
School of Engineering
Tokyo Institute of Technology

Prof. Nobuyuki Iwatsuki

Lecture syllabus

Course title: Advanced Mechanical Elements
Academic major: Mechanical Engineering
Offered quarter: 1Q
Day/period: Thursday 3-4
Credits: 1-0-0
Lecture room: Main Building M-101(H116)
Registration number: MECH.H431
Lecturer: Prof. Nobuyuki IWATSUKI
Office: I1-305
E-mail: iwatsuki.n.aa@m.titech.ac.jp

Course description and aims:

This course offers the knowledge on kinematic and dynamic analysis of planar and spatial link mechanisms and methods to control redundant and over/underactuated mechanisms.

The characteristic of motion mechanisms affects the performance of advanced mechanical systems such as robots. It is thus required to quantitatively evaluate the characteristic and to design and control mechanisms based on the evaluation criteria.

For this purpose, students will understand how to analyze the kinematics and dynamics of planar and spatial closed-loop link mechanisms by utilizing the systematic kinematic analysis method and will experience to apply them to kinematic and dynamic analyses and motion control of redundant mechanisms, overactuated mechanisms of which actuator inputs are more than mobility of mechanism and underactuated mechanisms constrained by elastic elements or gravitational force.

Student learning outcomes:

By the end of this course, students will be able to:

(1)Explain mobility of mechanism and relation between input/output motion of mechanism

(2)Analyze displacement, velocity and acceleration of planar/spatial closed-loop link mechanisms with the systematic kinematic analysis method

(3)Analyze the dynamics of planar/spatial closed-loop link mechanisms utilizing the systematic kinematic analysis method

(4)Explain the optimum motion control of redundant link mechanisms

(5)Explain motion control of overactuated or underactuated mechanisms with elastic elements

Keywords:

Close-loop link mechanism, kinematics, dynamics, systematic kinematic analysis, motion control, optimum control, redundancy, overactuated mechanism, underactuated mechanism, wire-driven mechanism

Class flow:

Important issues are summarized at the end of lecture every week. Because sample software to analyze mechanisms, which are explained in this lecture, will be offered to students, students are expected to try to analyze various mechanisms with the software by themselves so as to understand what they learn.

Schedule:

Class 1 (April 13) — Kinematic analysis of planar link mechanism with the systematic kinematic analysis method
- Displacement, velocity and acceleration analyses of planar closed-loop link mechanism with the systematic kinematic analysis method -

Class 2 (April 20) — Kinematic analysis of spatial link mechanism with the systematic kinematic analysis method
- Expansion of the systematic kinematic analysis method to spatial mechanisms -

Class 3 (April 27) — Dynamic analysis of planar/spatial link mechanisms
- Driving force and joint force analysis using the systematic kinematic analysis -

Class 4 (May 8*) — Motion control of redundant link mechanisms
- Optimum motion control to utilize redundancy -

*Monday

Class 5 (May 11) Kinetostatics analysis and motion control of overactuated mechanisms

- Motion control of overactuated mechanisms using relaxation with elastic elements -

Class 6 (May 18) Kinetostatic analysis and motion control of underactuated mechanisms with elastic elements

- Motion control of underactuated mechanisms constrained by elastic elements -

Class 7 (May 25) Kinetostatic analysis and motion control of underactuated wire-driven mechanisms

- Motion control of wire-driven underactuated mechanisms under gravitational force-

Textbook, reference book, course material:

No textbook is required. Several handouts will be distributed via T2SCHOLA. Students should download and print the handouts before class.

The following book is recommended as a reference book:

JSME Textbook series 'Kinematics of Machinery',
edited by the Japan Society of Mechanical Engineers (2007).

Assessment criteria and methods:

Students' knowledge of kinematic and dynamic analyses of planar/spatial closed-loop link mechanisms with the systematic kinematic analysis method and their application to analyses and motion controls of redundant and under/overactuated mechanisms is assessed via submitted reports on several issues.

Related courses:

Synthesis of robotic mechanisms, Design of robot controller

Prerequisites:

Students must have successfully completed 'Mechanical elements and draftings' or have equivalent knowledge.

April 13, 2023

Advanced Mechanical Elements
(Lecture 1)

Kinematic analysis of planar link mechanism with the systematic kinematic analysis method

Dept. of Mechanical Engineering
School of Engineering
Tokyo Institute of Technology

Prof. Nobuyuki Iwatsuki

1. Introduction of lecturer

Please visit my WEB-site:

http://www.rmsv.mech.e.titech.ac.jp/index.php

or

Search by Google with a keyword 'Iwatsuki lab'

Nobuyuki Iwatsuki, Professor, Dr. Eng.

Affiliation:

Department of Mechanical Engineering
School of Engineering
Tokyo Institute of Technology

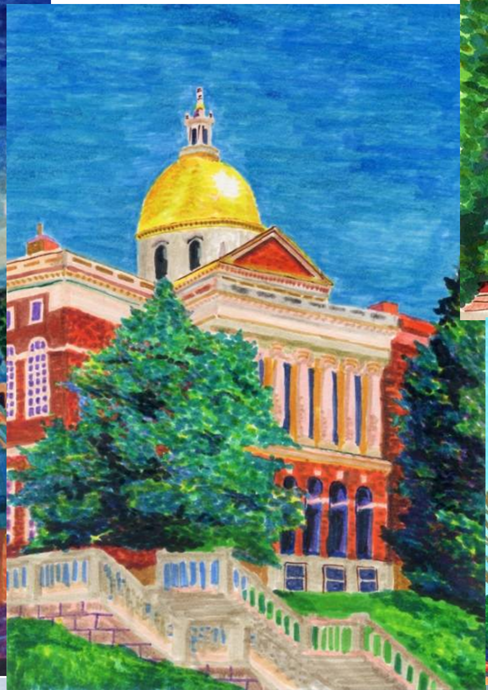
Office:

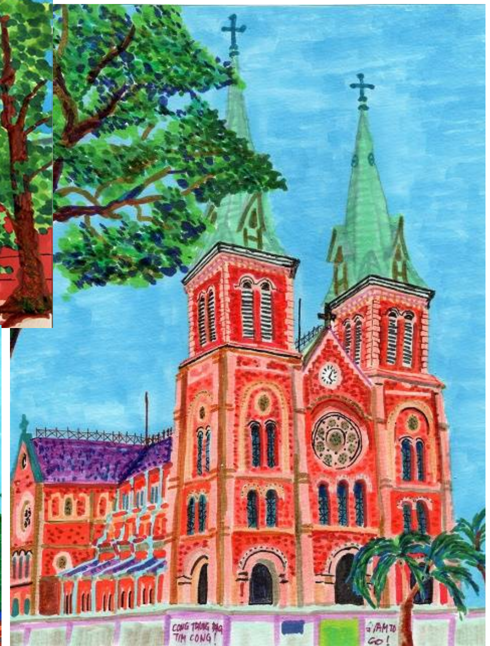
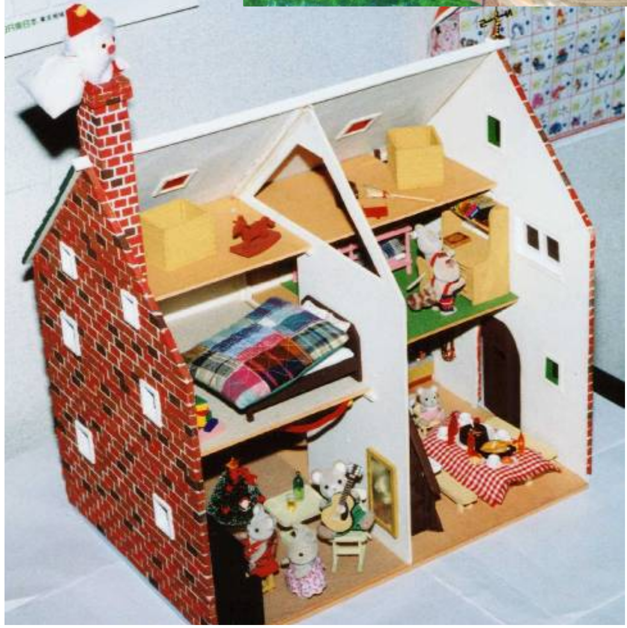
Room 305, Ishikawadai-1 Building,
Ohokayama Campus

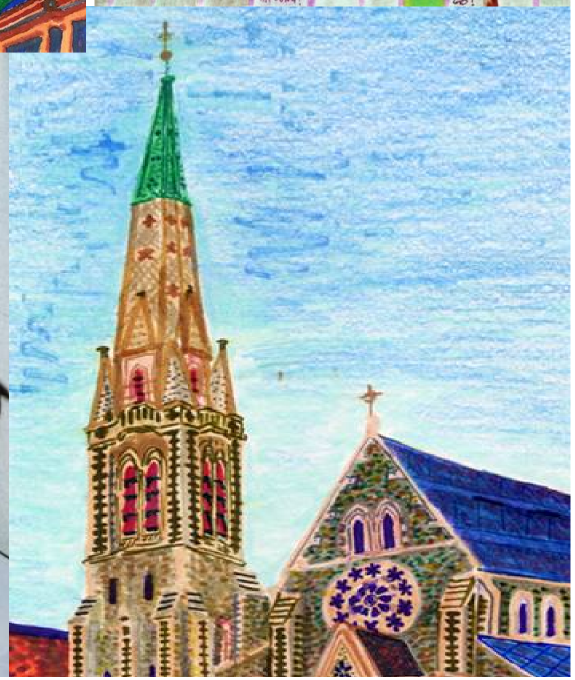
Contact:

Tel. +81-3-5734-2538
E-mail: iwatsuki.n.aa@m.titech.ac.jp

Several artistic works:



Research Themes:

(A)Synthesis and Control of Robotic Mechanisms

Especially focused on 'Mechanism Design and Motion Control of Hyper Redundant/Underactuated Robots'

(B)Silent Engineering

Especially focused on 'Estimation of Sound Power Radiating from Vibrating Structure and Structural Optimization to Reduce the Sound Power'

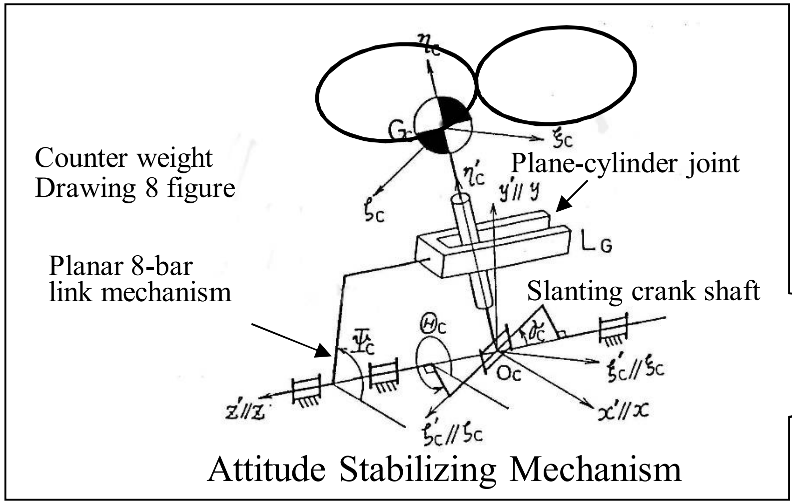
(C)Functional Material Actuators

Especially focused on 'Development of Micro Ciliary Actuators in Group'

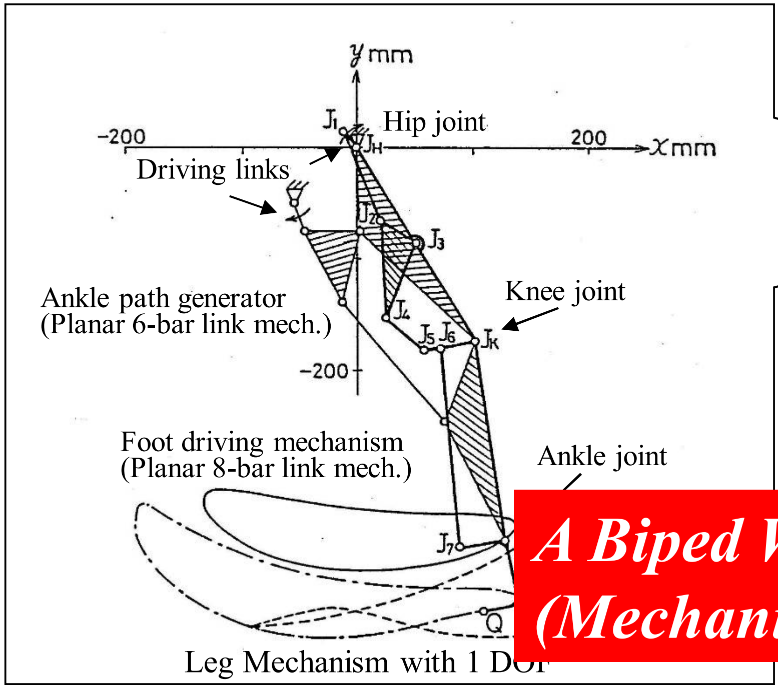
Several Examples of My Research Works:

(1)Biped Walking Machine

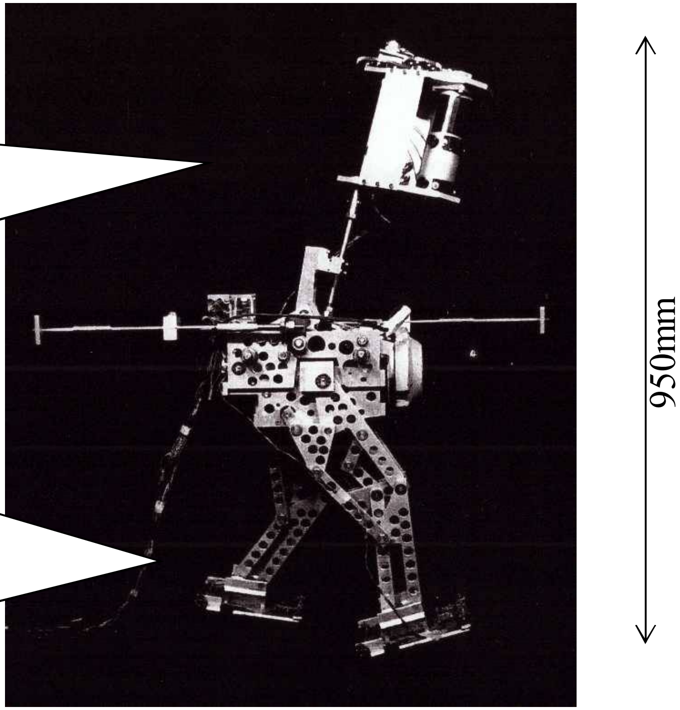
– While undergraduate/Ph.D student (-1987)



Attitude Stabilizing Mechanism

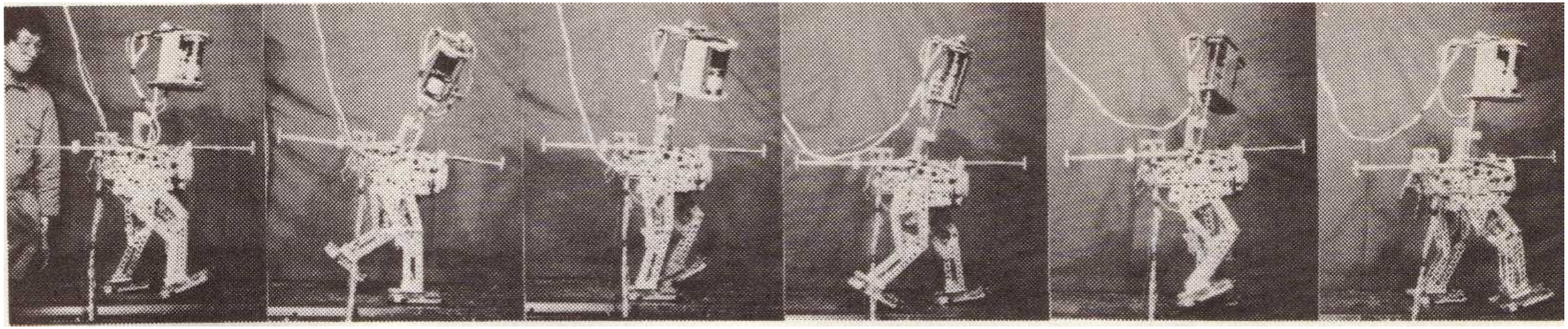


Leg Mechanism with 1 DOF



Straight Walk of Biped Walking Machine(1982)

A Biped Waking Machine with only 1 DOF (Mechanical synthesis for simple control)



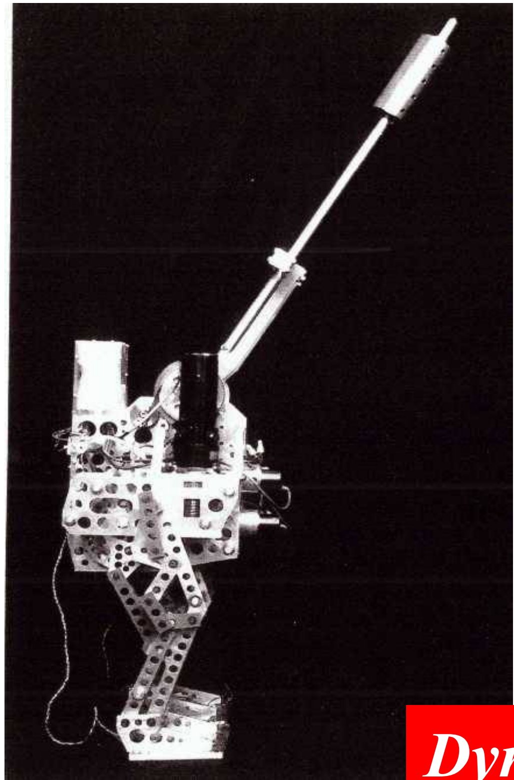
Walking speed :120 steps/min (World record)

⇩ Improvement for non steady walking

○Change stride
○Independent drive of foot driving mechanism
○Arbitrary control of attitude stabilizing mechanism

A new prototype with 5 DOF

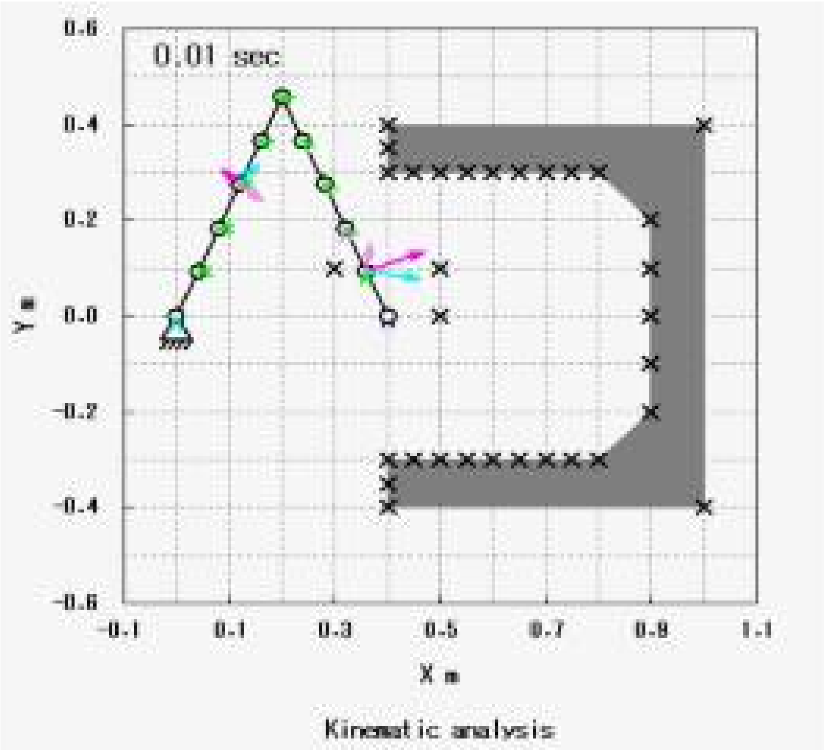
Items	Prototype-1	Prototype-2
Degree-of-freedom	1	5
Weight	20.6kgf	48.8kgf
Straight walk	Max. 120step/min	Max.15 steps/min
Start/stop to walk	-	Sucessful

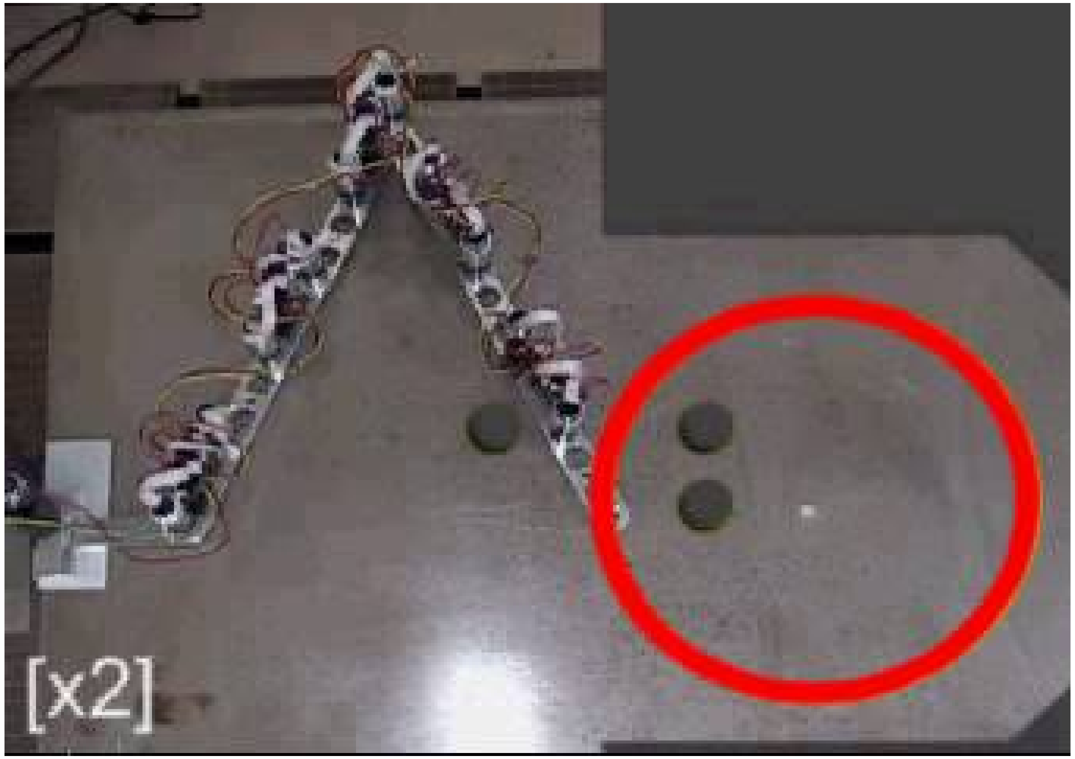


2nd prototype which c
and stop walking

Dynamics is important not only to stabilize biped machine but also to drive actuators !

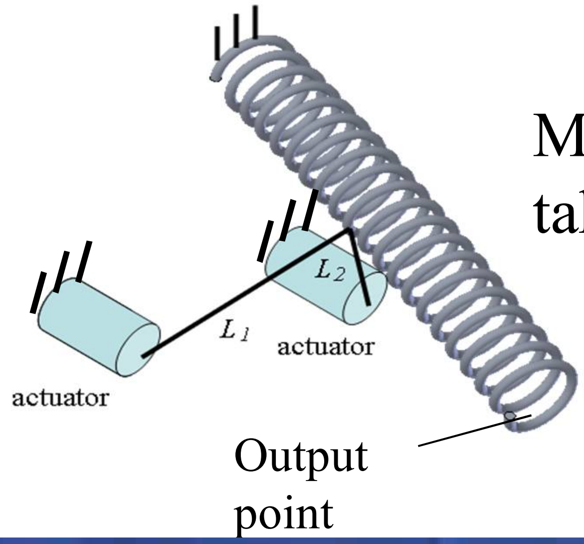
(2)Dexterous Motion of Hyper Redundant Robot
- Optimum learning control of snake-like robot with 10DOF to avoid obstackles



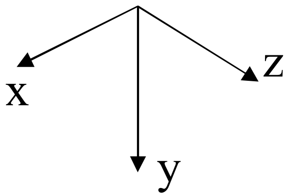


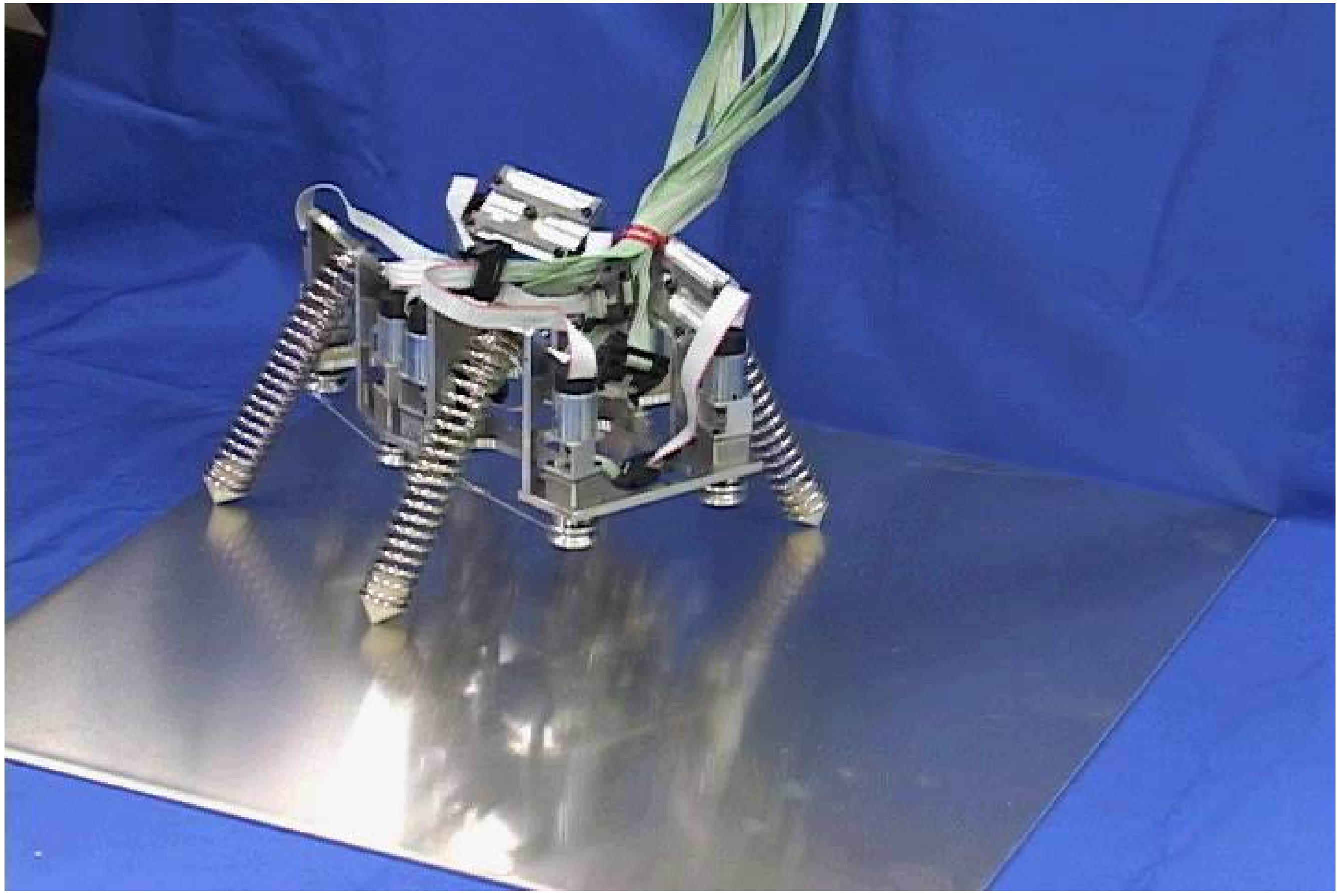
Dexterity is set as an objective function to generate optimum trajectory.

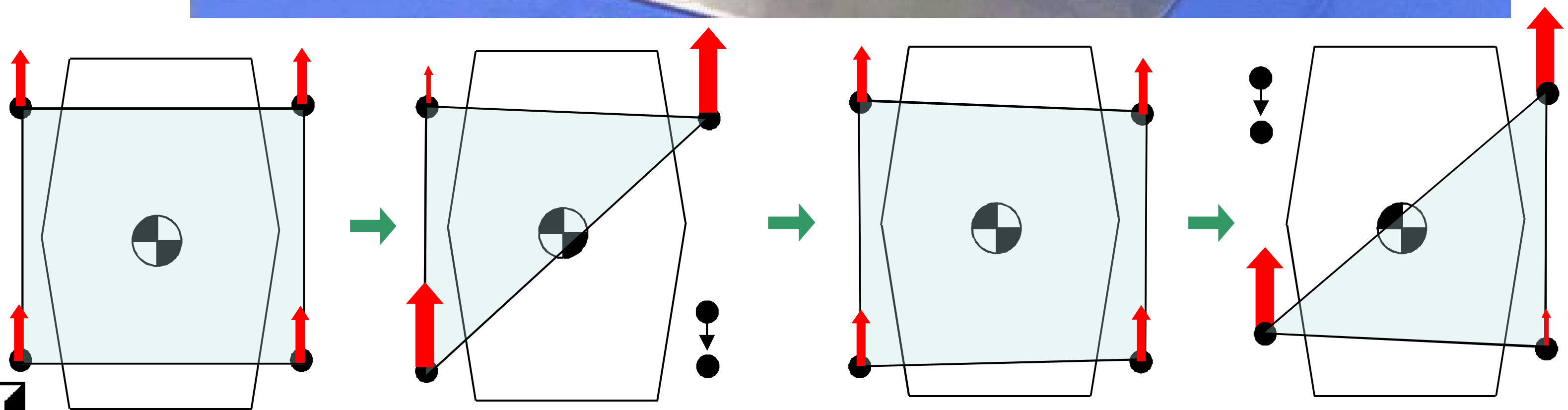
(3)Control of Structural Flexibility and Its application



Motion control of a flexible link (coil spring) taking account of its reaction force

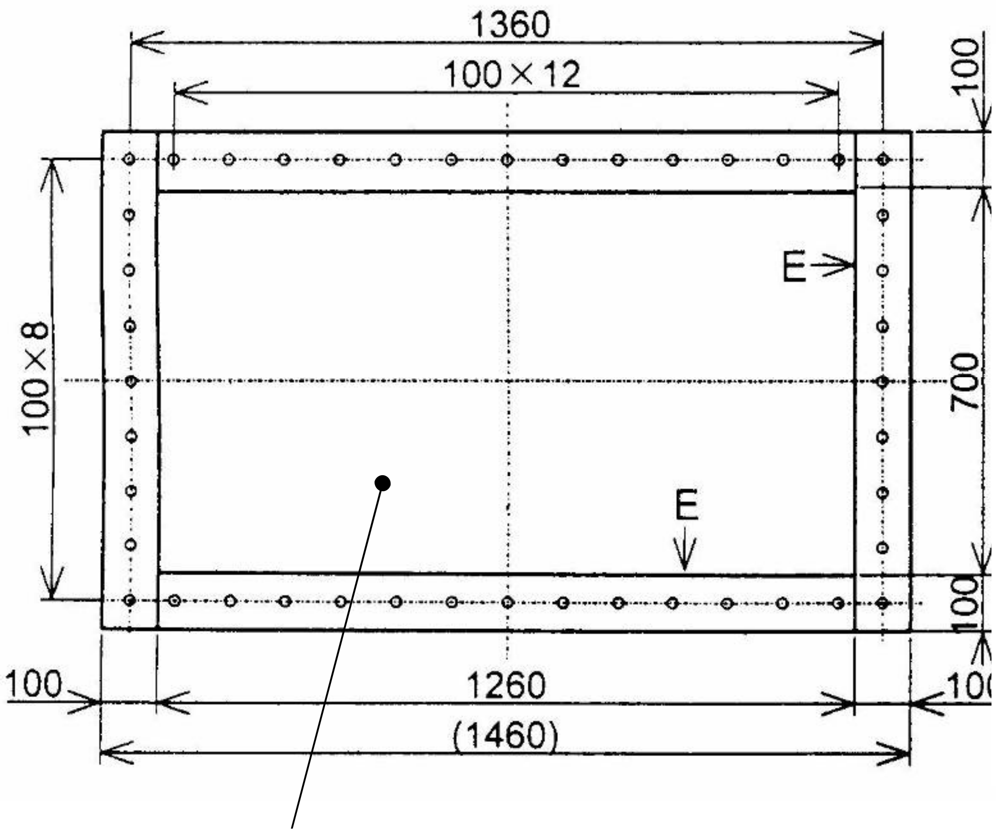




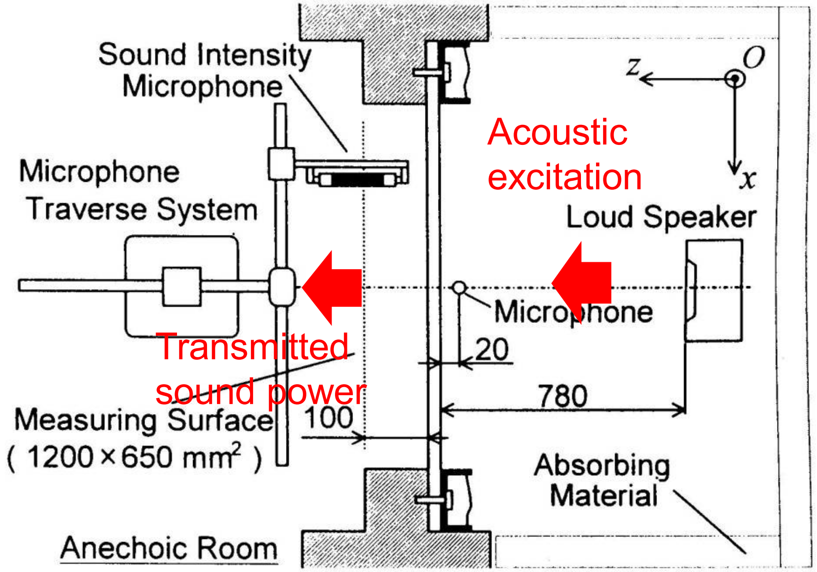


Four legged walking machine with spring leg

(4)Estimation of Sound Radiation from Thin Plate subjected to Acoustic Excitation

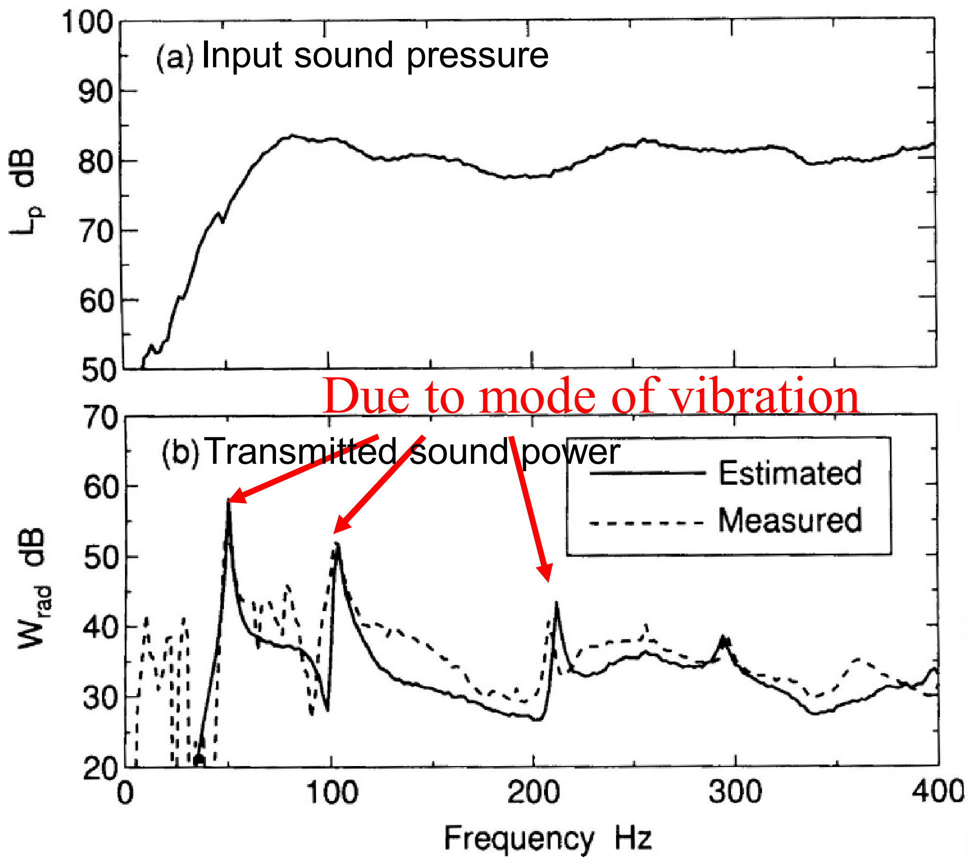


Thin rectangular plate

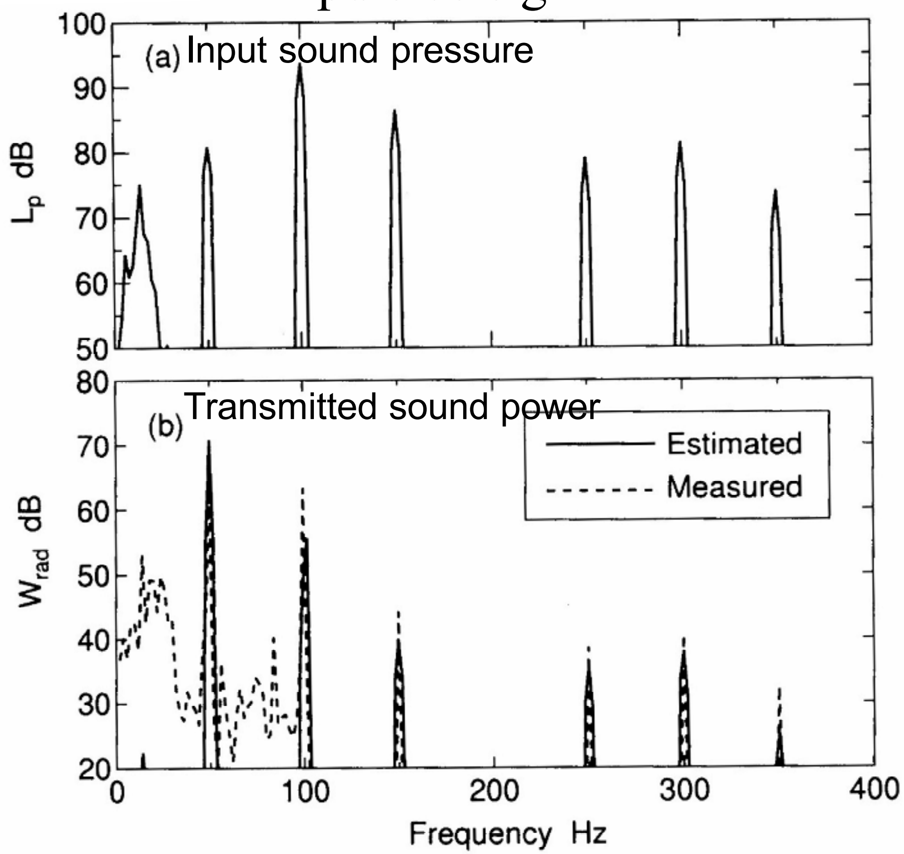


Transmitted sound power should be estimated to design silent structure.

Pseudo random noise

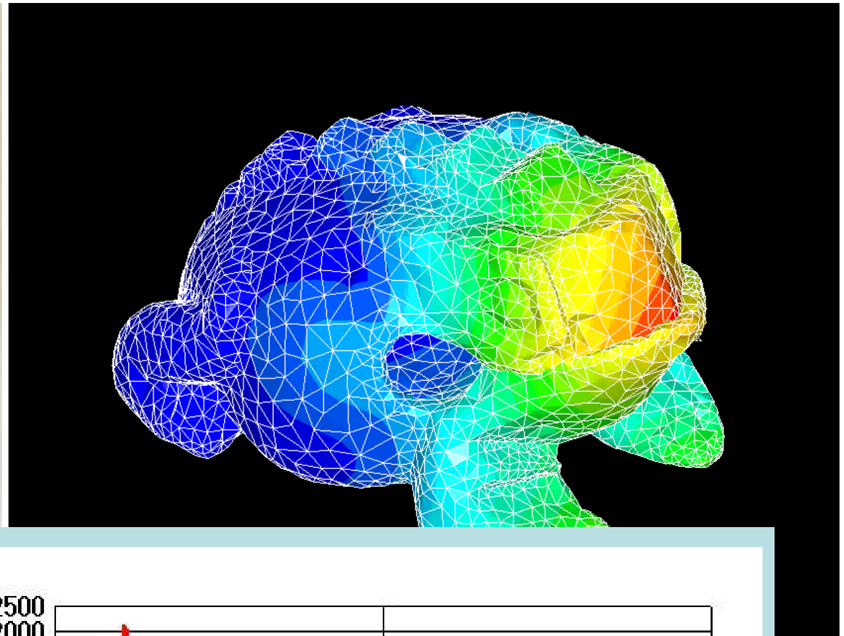


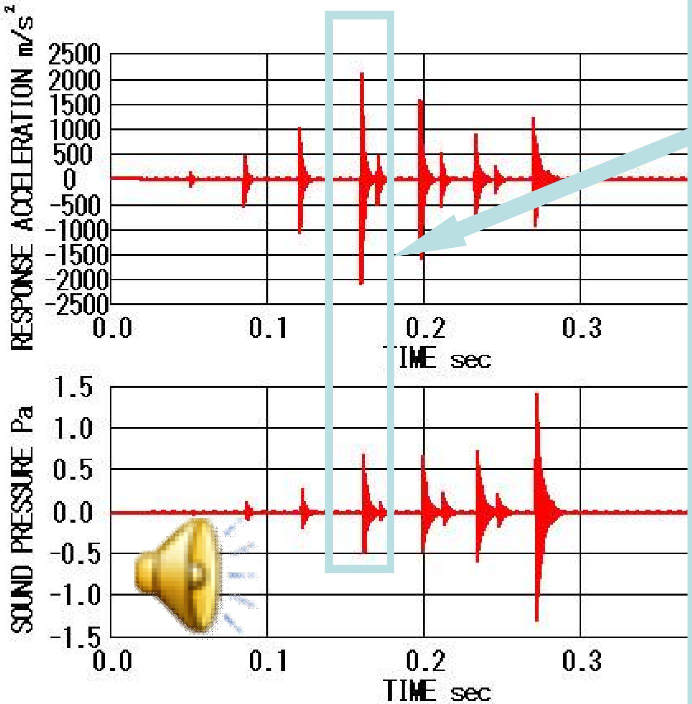
Impulsive signal

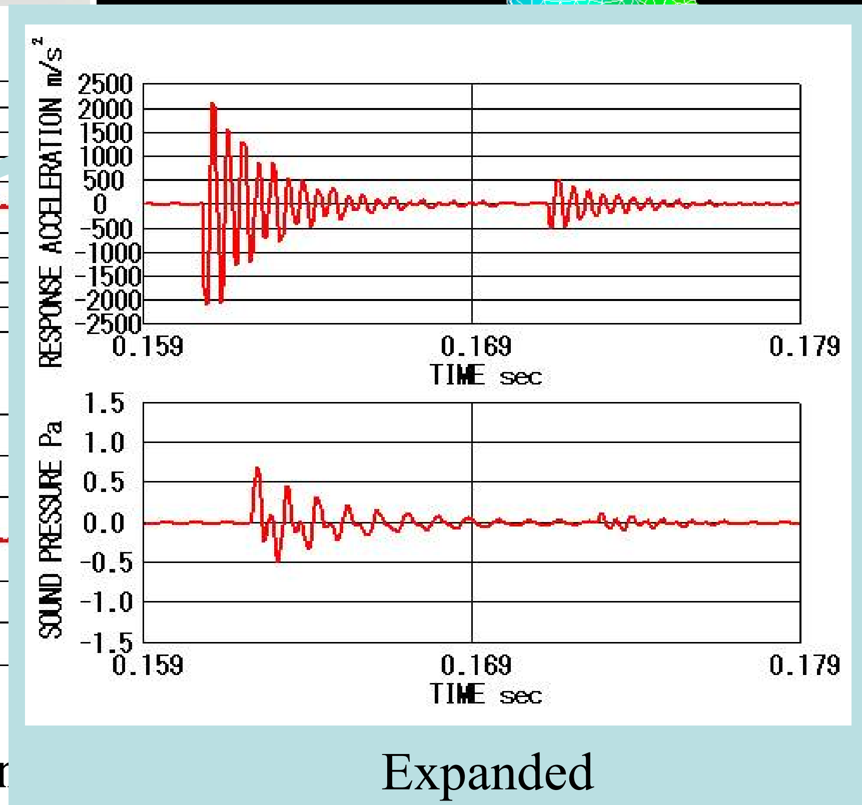


Transmitted sound power can be estimated with an adequate accuracy.

(5)Estimation of Sound Radiation from Frog-type Guiro



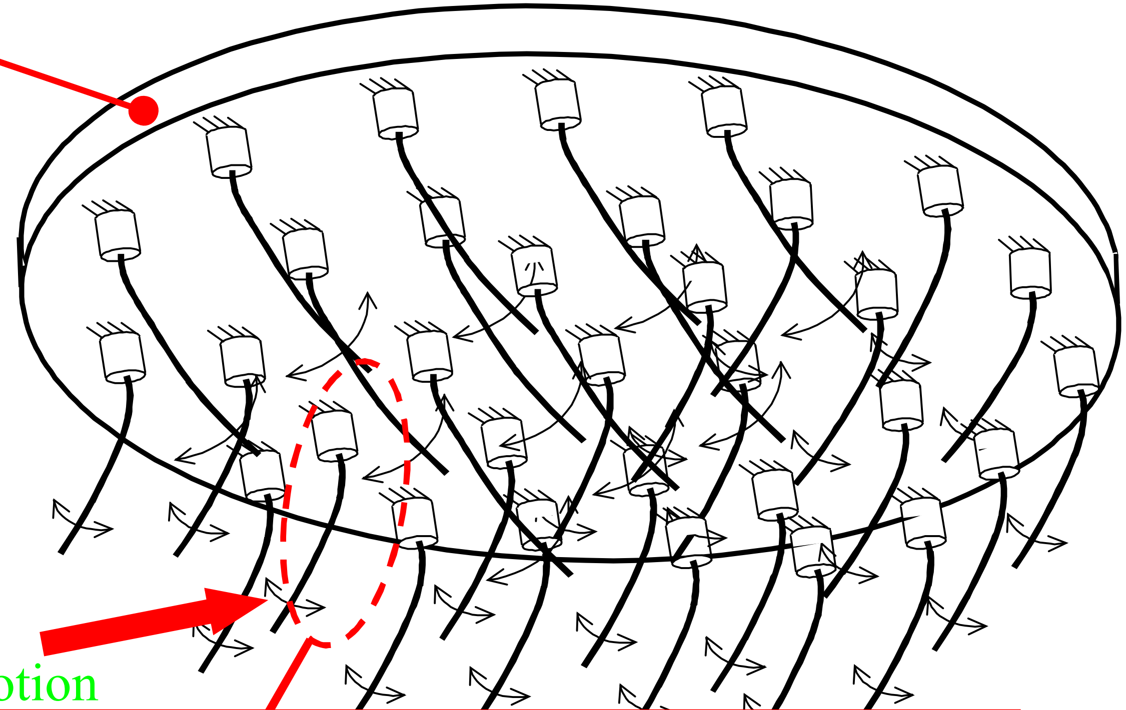


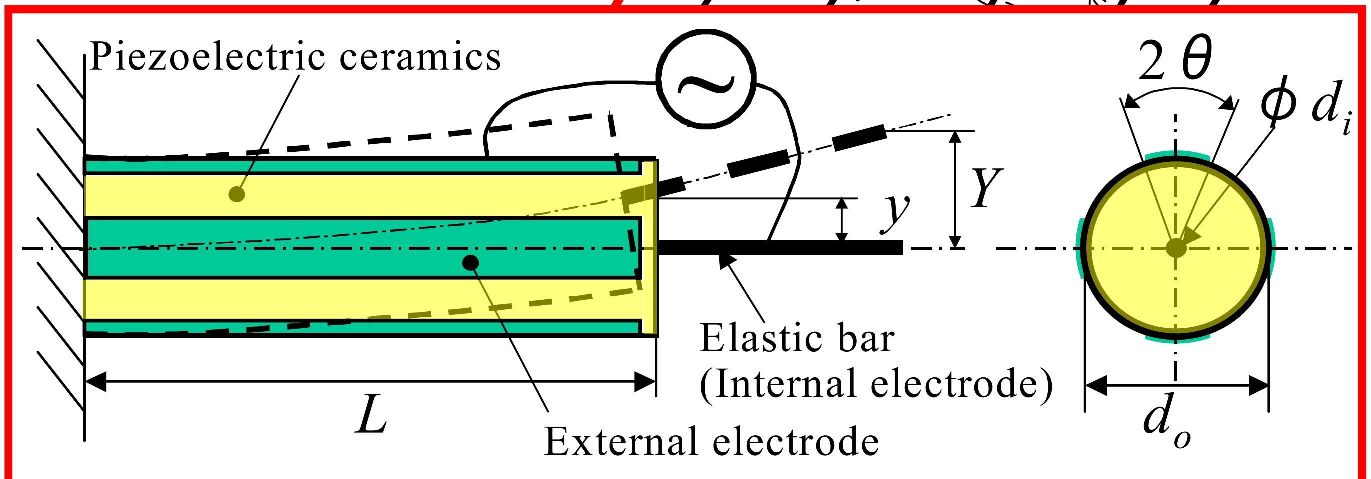


The calculated acceleration ar

(6)Development of Micro Ciliary Actuators in Group

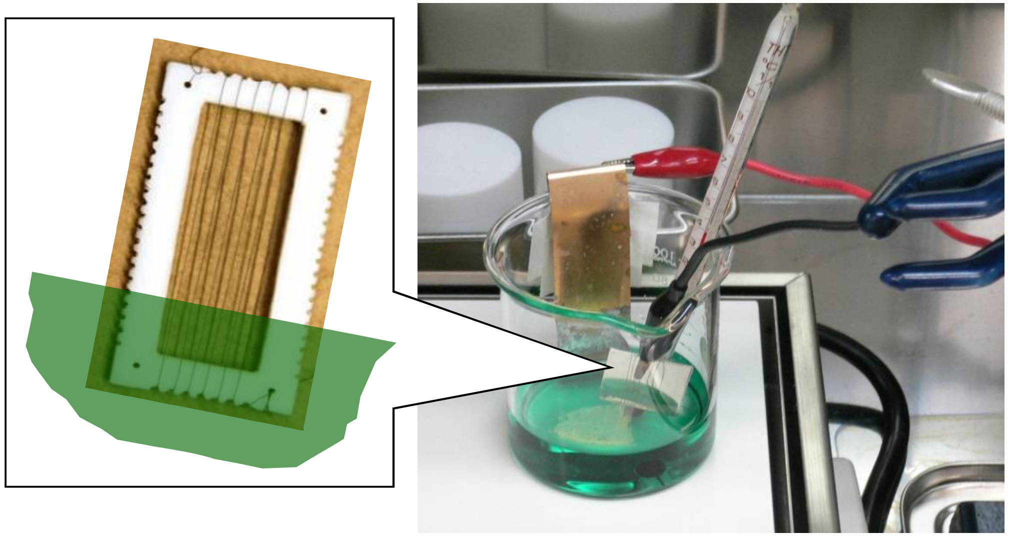
Micro machine driven by many artificial micro cilium actuators

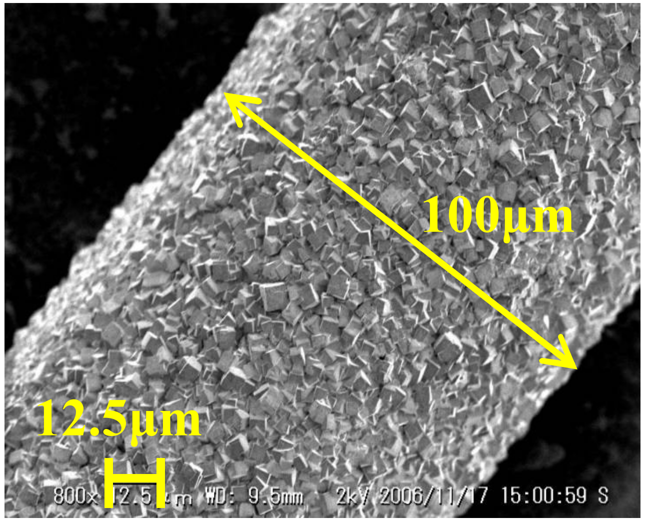




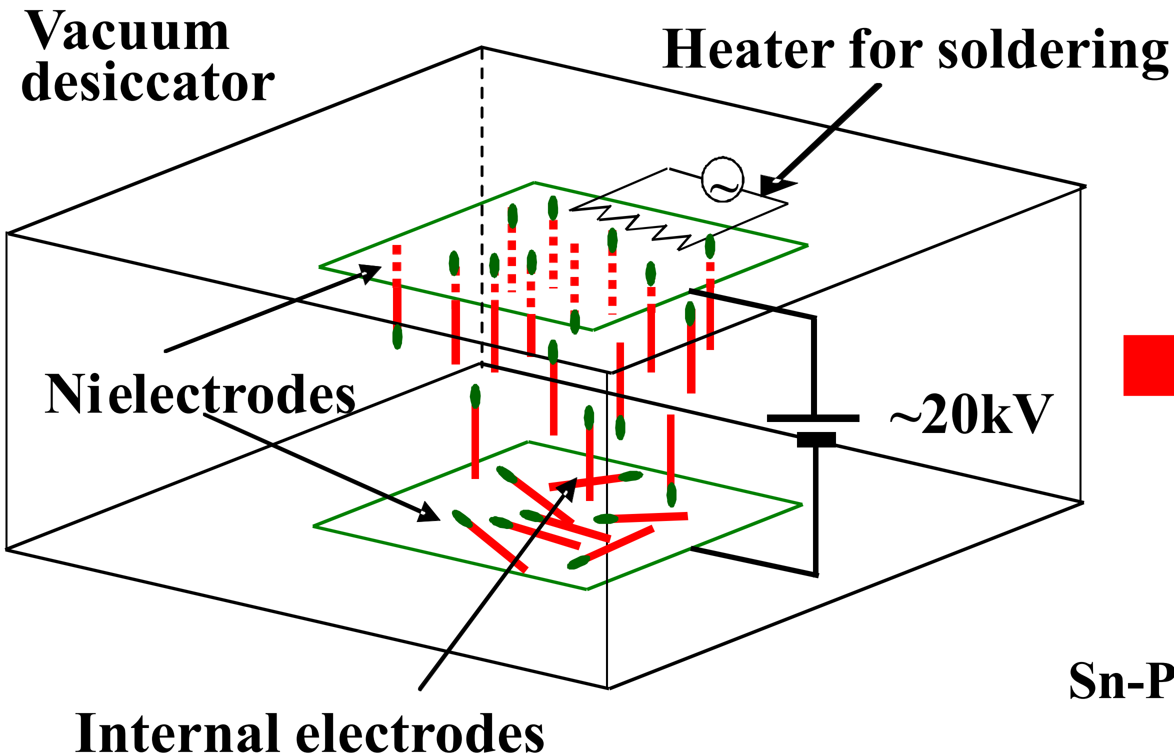
Piezoelectric Pipemorph actuator

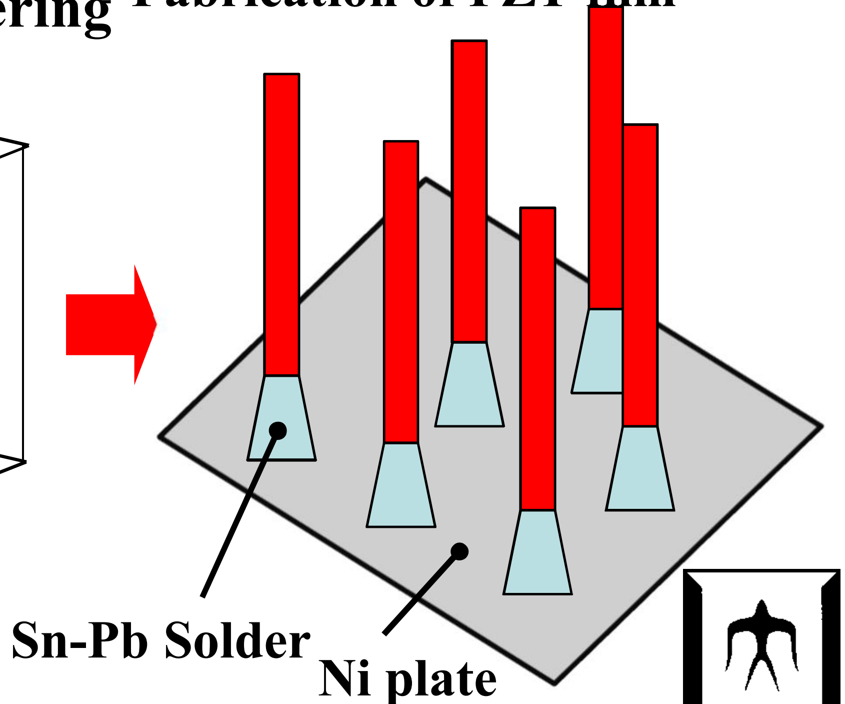
Several fabrication methods:

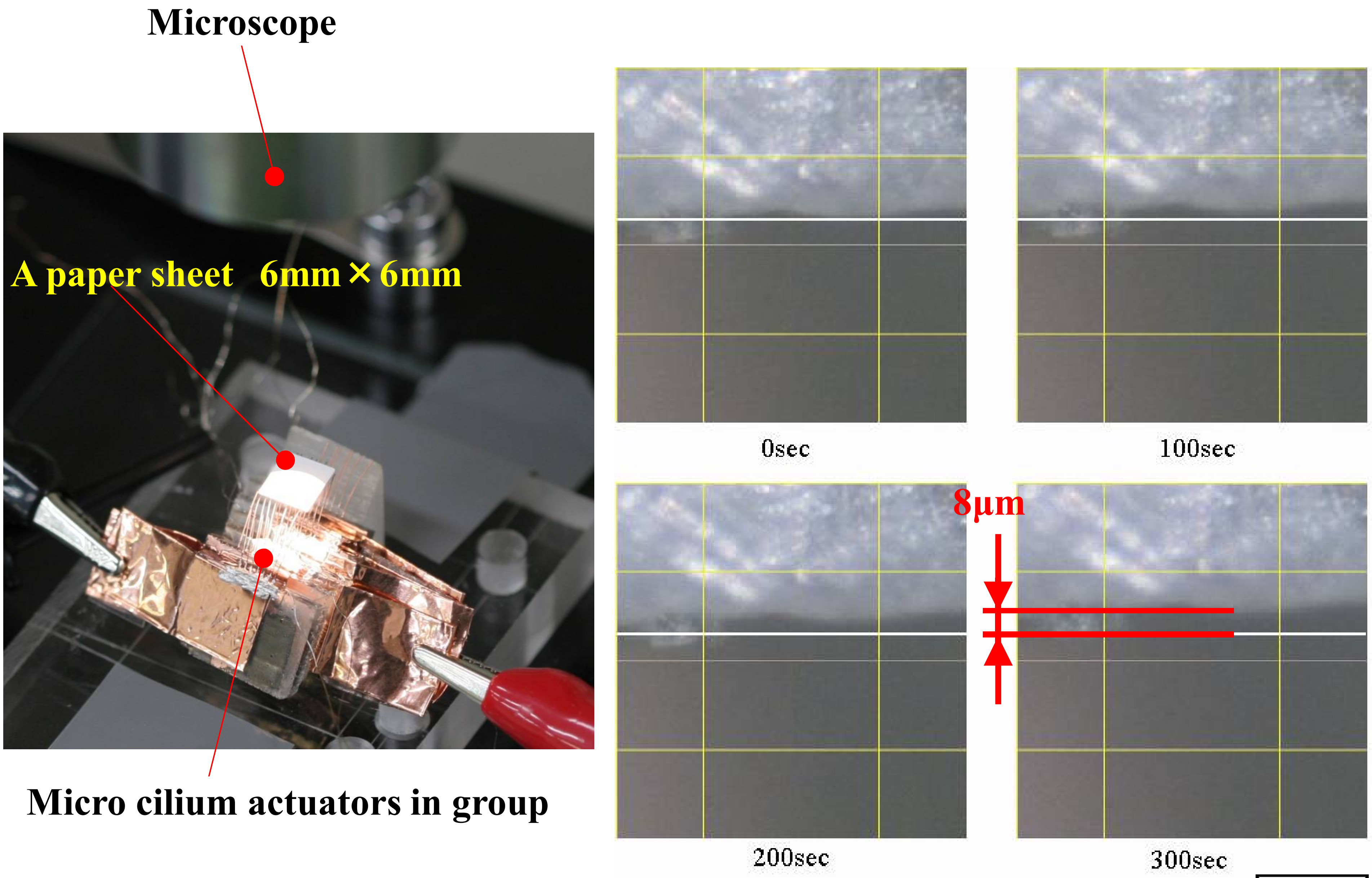




Fabrication of PZT film







Convey of a small sheet with micro cilium actuators in group

I hope you will visit my WEB-site:

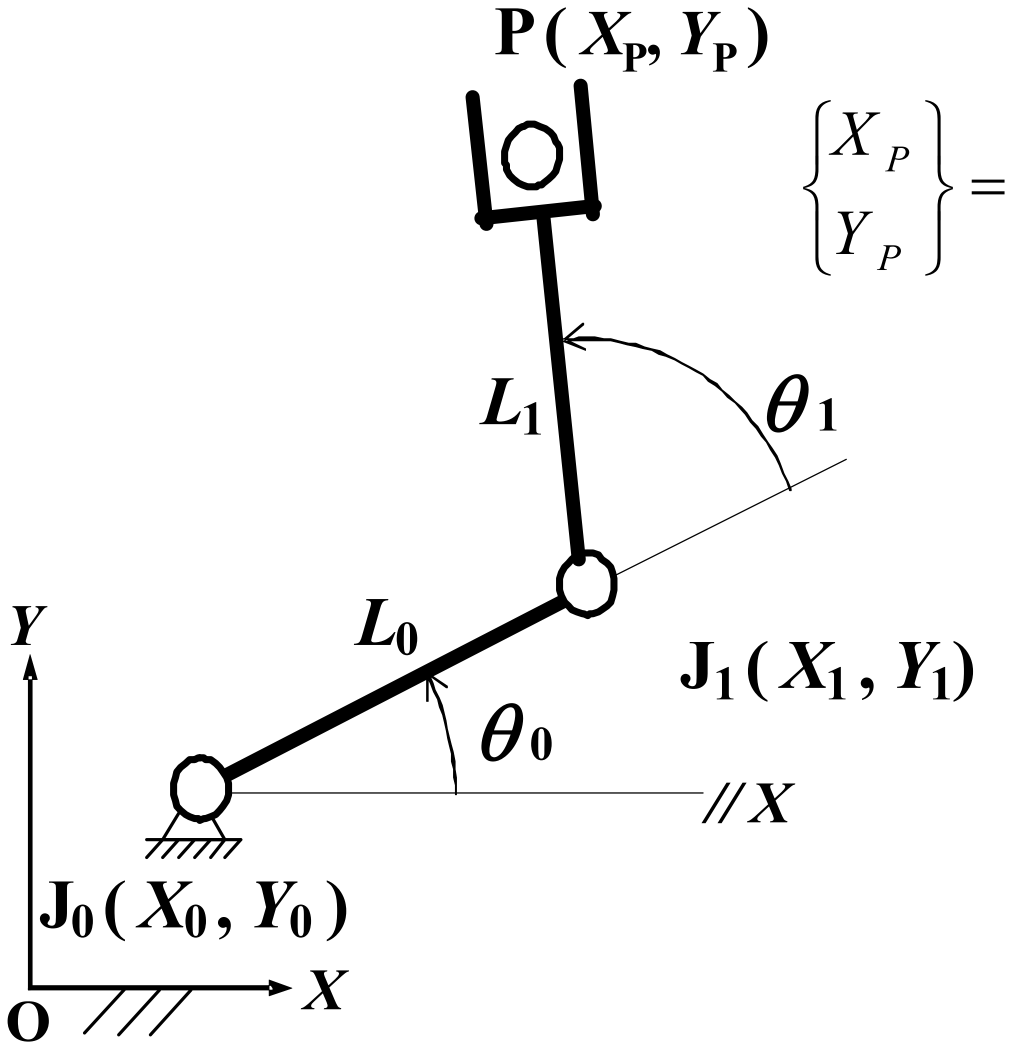
http://www.rmsv.mech.e.titech.ac.jp/index.php

if you are interested in other research works.

Thank you!

2. Kinematics of planar serial link mechanisms

Forward kinematics:



Planar serial 2R manipulator

$$\begin{Bmatrix} X_P \\ Y_P \end{Bmatrix} = \begin{Bmatrix} L_0 \cos\theta_0 + L_1 \cos(\theta_0 + \theta_1) + X_0 \\ L_0 \sin\theta_0 + L_1 \sin(\theta_0 + \theta_1) + Y_0 \end{Bmatrix}$$

It's easy to calculate output motion of the manipulator.

By differentiating the displacement, the output velocity can be calculated as

$$\begin{Bmatrix} \dot{X}_P \\ \dot{Y}_P \end{Bmatrix} = \begin{Bmatrix} -L_0 \dot{\theta}_0 \sin\theta_0 - L_1(\dot{\theta}_0 + \dot{\theta}_1)\sin(\theta_0 + \theta_1) \\ L_0 \dot{\theta}_0 \sin\theta_0 + L_1(\dot{\theta}_0 + \dot{\theta}_1)\cos(\theta_0 + \theta_1) \end{Bmatrix}$$

It can be deformed as

$$\begin{Bmatrix} \dot{X}_P \\ \dot{Y}_P \end{Bmatrix} = \begin{bmatrix} -L_0 \sin\theta_0 - L_1 \sin(\theta_0 + \theta_1) & -L_1 \sin(\theta_0 + \theta_1) \\ L_0 \cos\theta_0 + L_1 \cos(\theta_0 + \theta_1) & L_1 \cos(\theta_0 + \theta_1) \end{bmatrix} \begin{Bmatrix} \dot{\theta}_0 \\ \dot{\theta}_1 \end{Bmatrix}$$

Output velocity vector $\dot{\boldsymbol{r}}$

Jacobian matrix $\boldsymbol{J}(\boldsymbol{\theta})$

Joint velocity vector $\dot{\boldsymbol{\theta}}$

$$\dot{\boldsymbol{r}} = \boldsymbol{J}(\boldsymbol{\theta})\dot{\boldsymbol{\theta}}$$

Linear relationship between output velocity and joint velocity

Output acceleration can also be calculated as

$$\begin{Bmatrix} \ddot{X}_P \\ \ddot{Y}_P \end{Bmatrix} = \begin{Bmatrix} -L_0(\ddot{\theta}_0 \sin\theta_0 + \dot{\theta}_0^2 \cos\theta_0) \\ -L_1[(\ddot{\theta}_0 + \ddot{\theta}_1)\sin(\theta_0 + \theta_1) + (\dot{\theta}_0 + \dot{\theta}_1)^2 \cos(\theta_0 + \theta_1)] \\ L_0(\ddot{\theta}_0 \cos\theta_0 - \dot{\theta}_0^2 \sin\theta_0) \\ +L_1[(\ddot{\theta}_0 + \ddot{\theta}_1)\cos(\theta_0 + \theta_1) - (\dot{\theta}_0 + \dot{\theta}_1)^2 \sin(\theta_0 + \theta_1)] \end{Bmatrix}$$

$$= \dot{\boldsymbol{J}}(\boldsymbol{\theta})\dot{\boldsymbol{\theta}} + \boldsymbol{J}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}}$$

It is easy to analyze output motion with knowledge at high school!

For general planar serial nR manipulator….

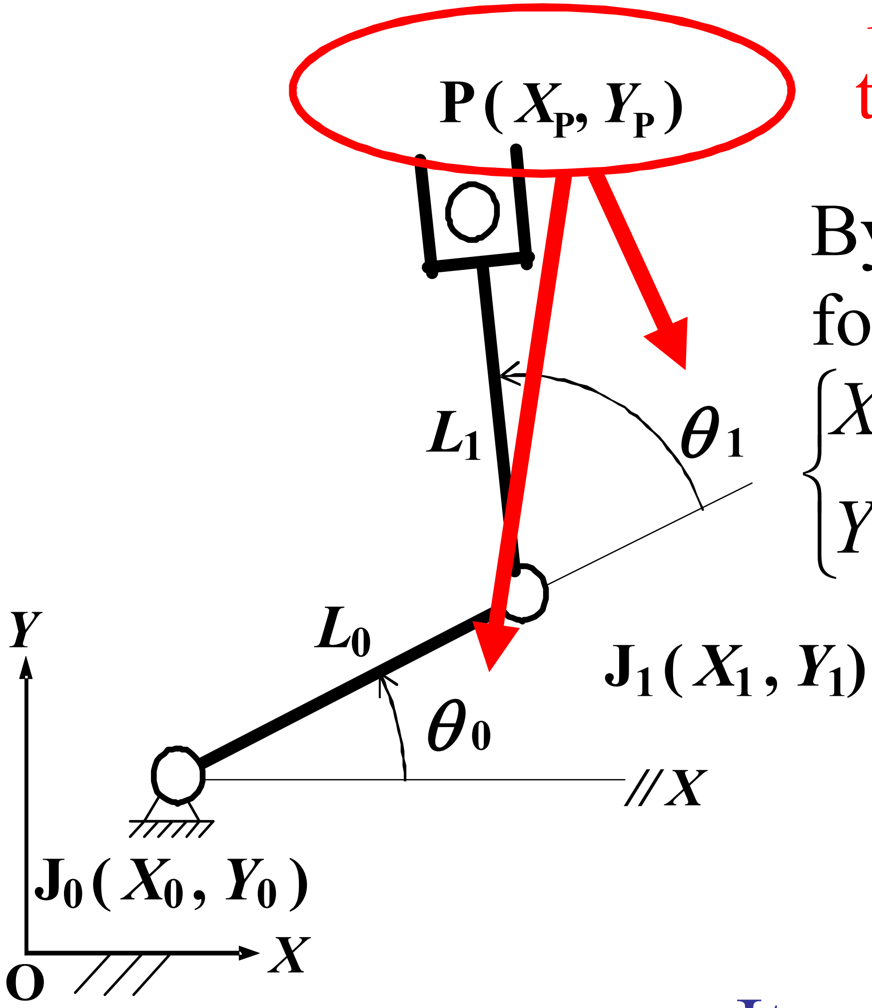
$$\begin{Bmatrix} X_P \\ Y_P \end{Bmatrix} = \begin{Bmatrix} \sum_{i=0}^{n-1} L_i \cos \sum_{k=0}^{i} \theta_k + X_0 \\ \sum_{i=0}^{n-1} L_i \sin \sum_{k=0}^{i} \theta_k + Y_0 \end{Bmatrix}$$

It's also easy to analyze!

$$\begin{Bmatrix} \dot{X}_P \\ \dot{Y}_P \end{Bmatrix} = \begin{Bmatrix} -\sum_{i=0}^{n-1} L_i \sum_{k=0}^{i} \dot{\theta}_k \sin \sum_{k=0}^{i} \theta_k \\ \sum_{i=0}^{n-1} L_i \sum_{k=0}^{i} \dot{\theta}_k \cos \sum_{k=0}^{i} \theta_k \end{Bmatrix}$$

$$\begin{Bmatrix} \ddot{X}_P \\ \ddot{Y}_P \end{Bmatrix} = \begin{Bmatrix} -\sum_{i=0}^{n-1} L_i [\sum_{k=0}^{i} \ddot{\theta}_k \sin \sum_{k=0}^{i} \theta_k + (\sum_{k=0}^{i} \dot{\theta}_k)^2 \cos \sum_{k=0}^{i} \theta_k] \\ \sum_{i=0}^{n-1} L_i [\sum_{k=0}^{i} \ddot{\theta}_k \cos \sum_{k=0}^{i} \theta_k - (\sum_{k=0}^{i} \dot{\theta}_k)^2 \sin \sum_{k=0}^{i} \theta_k] \end{Bmatrix}$$

Inverse kinematics



Planar serial 2R manipulator

To calculate joint input motion for the specified output motion

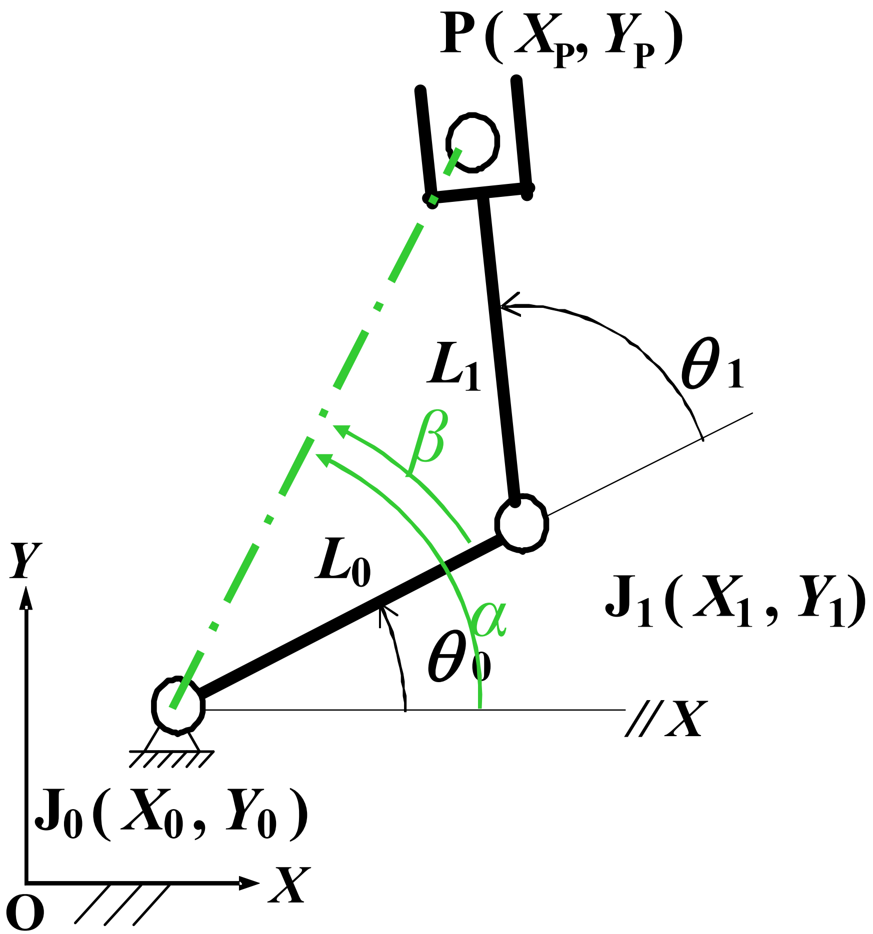
By using the prementioned equation for forward kinematics:

$$\begin{Bmatrix} X_P \\ Y_P \end{Bmatrix} = \begin{Bmatrix} L_0 \cos\theta_0 + L_1 \cos(\theta_0 + \theta_1) + X_0 \\ L_0 \sin\theta_0 + L_1 \sin(\theta_0 + \theta_1) + Y_0 \end{Bmatrix}$$

Can θ_0, θ_1 be calculated as the function of X_P, Y_P?

It will be a little bit complicated.

Let draw an additional straight line!



Planar serial 2R manipulator

By defining angles α, β

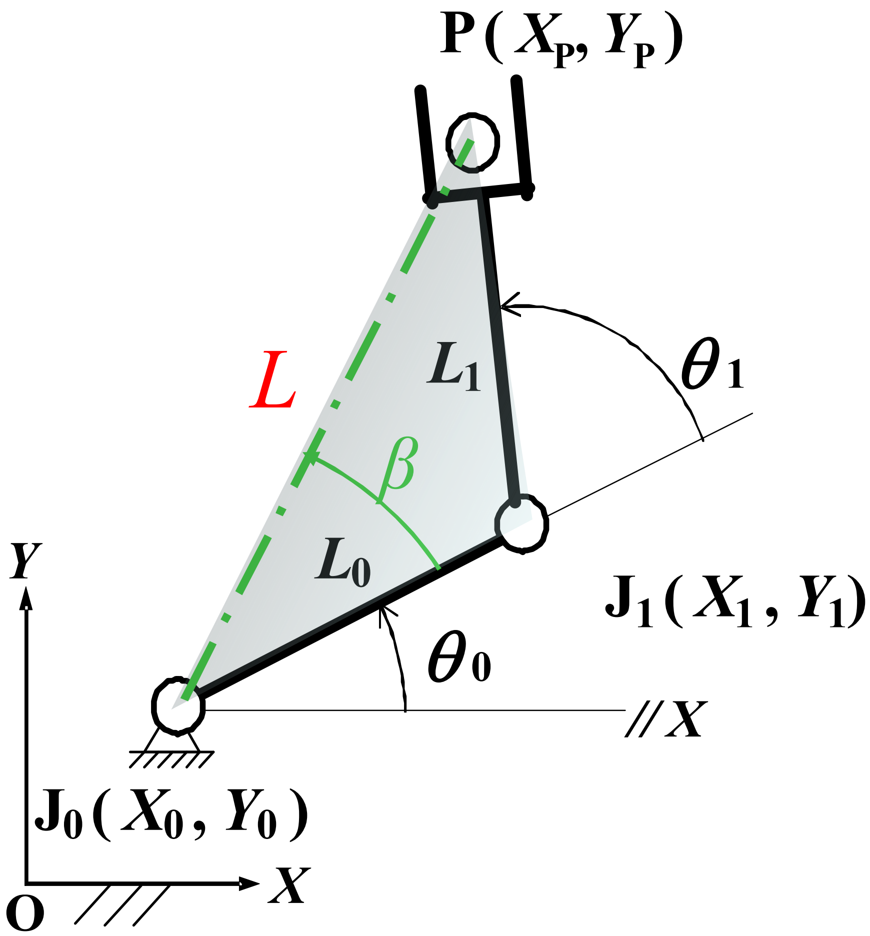
$$\theta_0 = \alpha - \beta$$

where

$$\alpha = \tan^{-1} \frac{Y_P - Y_0}{X_P - X_0}$$

How to derive β ?

Let consider $\triangle PJ_0J_1$.



Planar serial 2R manipulator

By using side lengths L_0, L_1 and distance:

$$L = \sqrt{(X_P - X_0)^2 + (Y_P - Y_0)^2}$$

We can apply cosine theorem:

$$\cos\beta = \frac{L^2 + L_0^2 - L_1^2}{2LL_0}$$

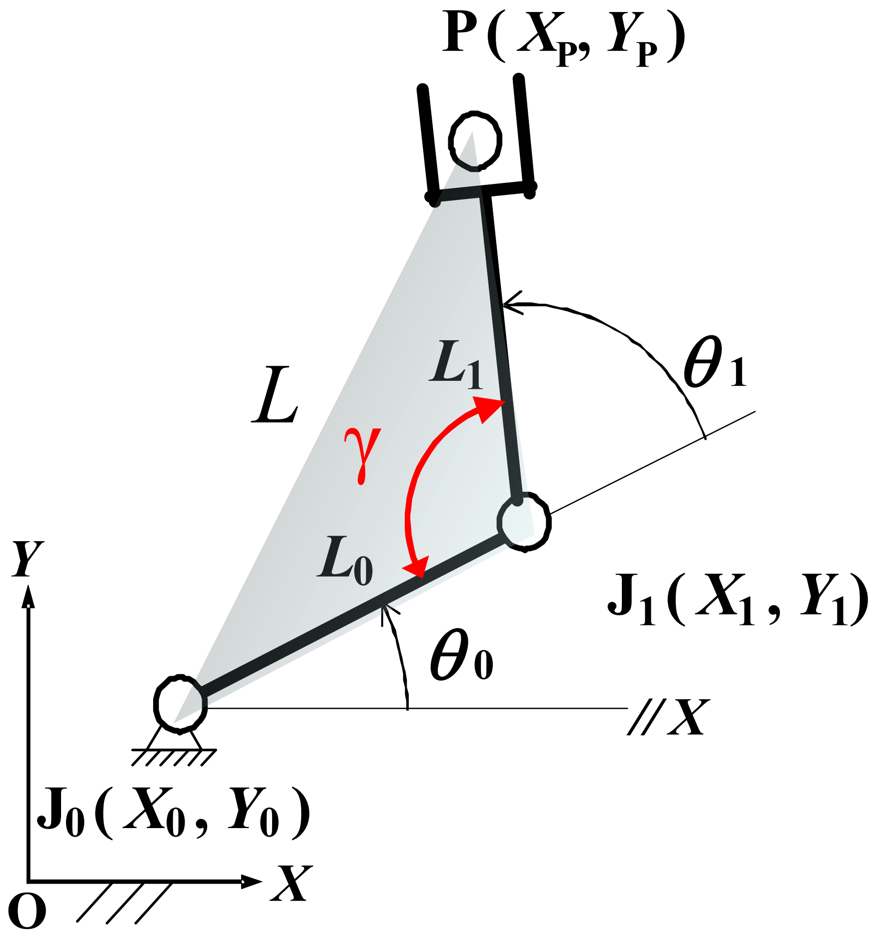
Therefore

$$\beta = \cos^{-1}\frac{L^2 + L_0^2 - L_1^2}{2LL_0}$$

$$\theta_0 = \alpha - \beta$$

$$= \tan^{-1}\frac{Y_P - Y_0}{X_P - X_0} - \cos^{-1}\frac{L^2 + L_0^2 - L_1^2}{2LL_0}$$

θ_1 can then be calculated.



Let consider $\triangle PJ_0J_1$ and an angle γ.

$$\theta_1 = \pi - \gamma$$

From cosine theorem

$$\gamma = \cos^{-1}\frac{L_0^2 + L_1^2 - L^2}{2L_0L_1}$$

$$\theta_1 = \pi - \gamma$$

$$= \pi - \cos^{-1}\frac{L_0^2 + L_1^2 - L^2}{2L_0 L_1}$$

Joint input velocity and acceleration can be derived as

$$\dot{\theta}_0 = \dot{\alpha} - \dot{\beta}$$

$$\ddot{\theta}_0 = \ddot{\alpha} - \ddot{\beta}$$

$$\dot{\theta}_1 = -\dot{\gamma}$$

$$\ddot{\theta}_1 = -\ddot{\gamma}$$

$$\dot{\gamma} = \frac{\dot{L}L}{L_0 L_1 \sin\gamma}$$

$$\ddot{\gamma} = \frac{(\ddot{L}L + \dot{L}^2)\sin\gamma - \dot{L}L\dot{\gamma}\cos\gamma}{L_0 L_1 \sin^2\gamma}$$

$$\dot{L} = \frac{\dot{X}_P X_P + \dot{Y}_P Y_P}{L}$$

$$\ddot{L} = \frac{(\ddot{X}_P X_P + \dot{X}_P^2 + \ddot{Y}_P Y_P + \dot{Y}_P^2)L - (\dot{X}_P X_P + \dot{Y}_P Y_P)\dot{L}}{L^2}$$

They can also be derived as

$$\dot{\boldsymbol{\theta}} = \boldsymbol{J}(\boldsymbol{\theta})^{-1}\dot{\boldsymbol{r}}$$

$$= \begin{bmatrix} -L_0 \sin\theta_0 - L_1 \sin(\theta_0+\theta_1) & -L_1 \sin(\theta_0+\theta_1) \\ L_0 \cos\theta_0 + L_1 \cos(\theta_0+\theta_1) & L_1 \cos(\theta_0+\theta_1) \end{bmatrix}^{-1} \begin{Bmatrix} \dot{X}_P \\ \dot{Y}_P \end{Bmatrix}$$

$$= \frac{1}{L_0 L_1 \sin\theta_1} \begin{bmatrix} L_1 \cos(\theta_0+\theta_1) & L_1 \sin(\theta_0+\theta_1) \\ -L_0 \cos\theta_0 - L_1 \cos(\theta_0+\theta_1) & -L_0 \sin\theta_0 - L_1 \sin(\theta_0+\theta_1) \end{bmatrix} \begin{Bmatrix} \dot{X}_P \\ \dot{Y}_P \end{Bmatrix}$$

$$\ddot{\boldsymbol{\theta}} = \boldsymbol{J}(\boldsymbol{\theta})^{-1}[\ddot{\boldsymbol{r}} - \dot{\boldsymbol{J}}(\boldsymbol{\theta})\dot{\boldsymbol{\theta}}]$$

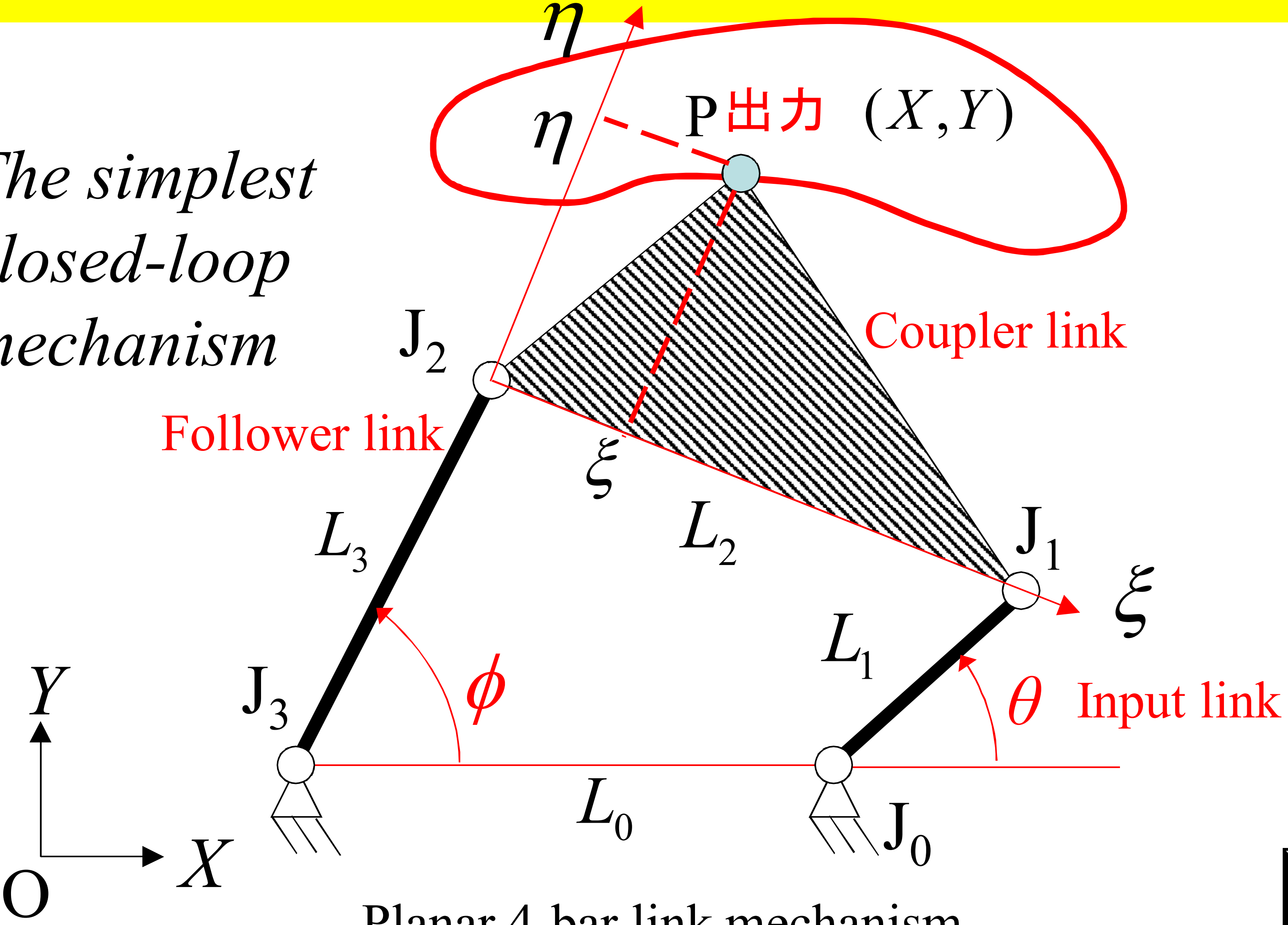
$$= \frac{1}{L_0 L_1 \sin\theta_1} \begin{bmatrix} L_1 \cos(\theta_0+\theta_1) & L_1 \sin(\theta_0+\theta_1) \\ -L_0 \cos\theta_0 - L_1 \cos(\theta_0+\theta_1) & -L_0 \sin\theta_0 - L_1 \sin(\theta_0+\theta_1) \end{bmatrix}$$

$$\times \left[\begin{Bmatrix} \ddot{X}_P \\ \ddot{Y}_P \end{Bmatrix} - \begin{bmatrix} -L_0\dot{\theta}_0 \cos\theta_0 - L_1(\dot{\theta}_0+\dot{\theta}_1)\cos(\theta_0+\theta_1) & -L_1(\dot{\theta}_0+\dot{\theta}_1)\cos(\theta_0+\theta_1) \\ -L_0\dot{\theta}_0 \sin\theta_0 - L_1(\dot{\theta}_0+\dot{\theta}_1)\sin(\theta_0+\theta_1) & -L_1(\dot{\theta}_0+\dot{\theta}_1)\sin(\theta_0+\theta_1) \end{bmatrix} \begin{Bmatrix} \dot{\theta}_0 \\ \dot{\theta}_1 \end{Bmatrix} \right]$$

Anyway, it is easy to analyze kinematics of serial mechanisms.

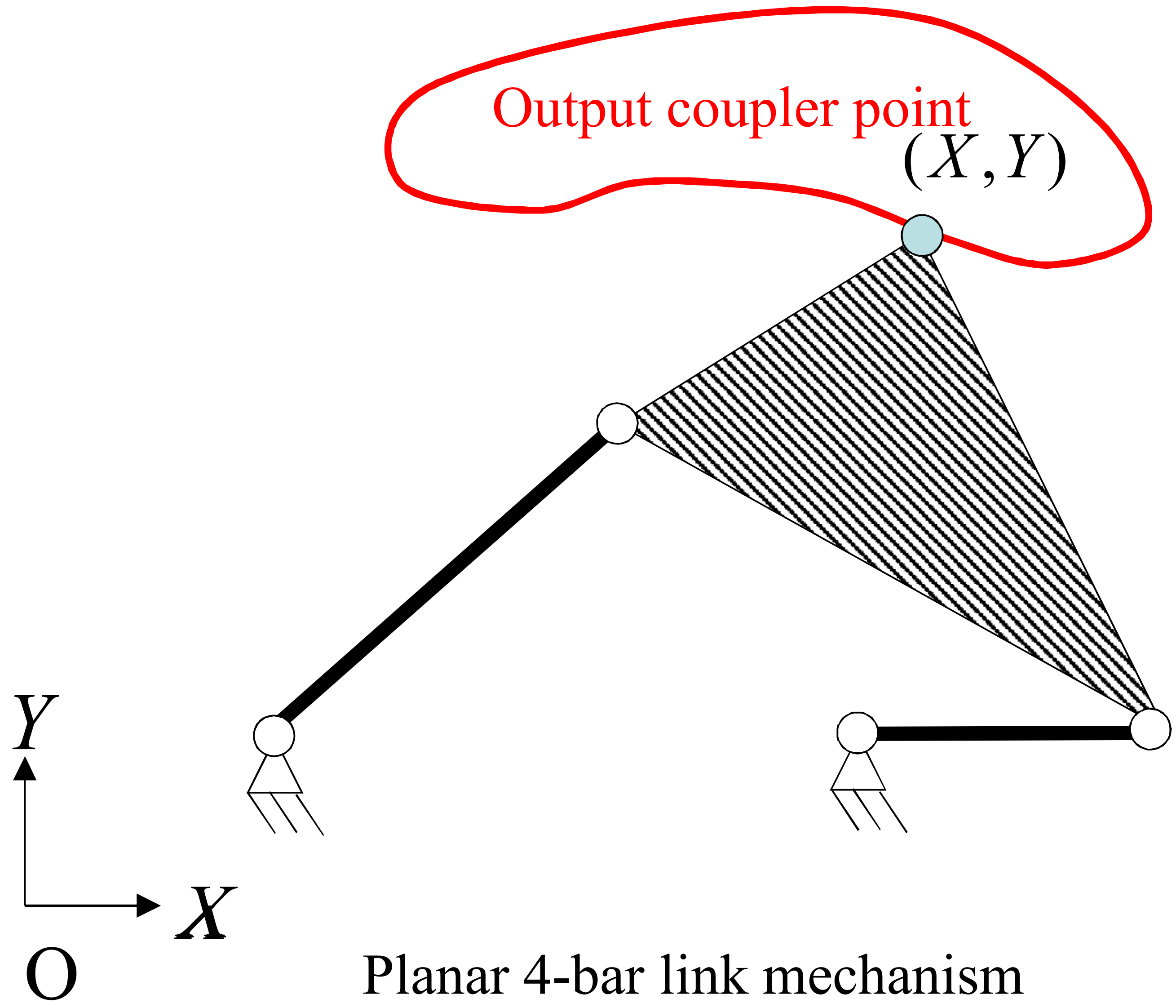
3. Kinematics of planar closed-loop link mechanisms

The simplest closed-loop mechanism



Planar 4-bar link mechanism

Crank-rocker motion:



Planar 4-bar link mechanism

Let analyze output angular motion.

(1)From closed-loop equation :

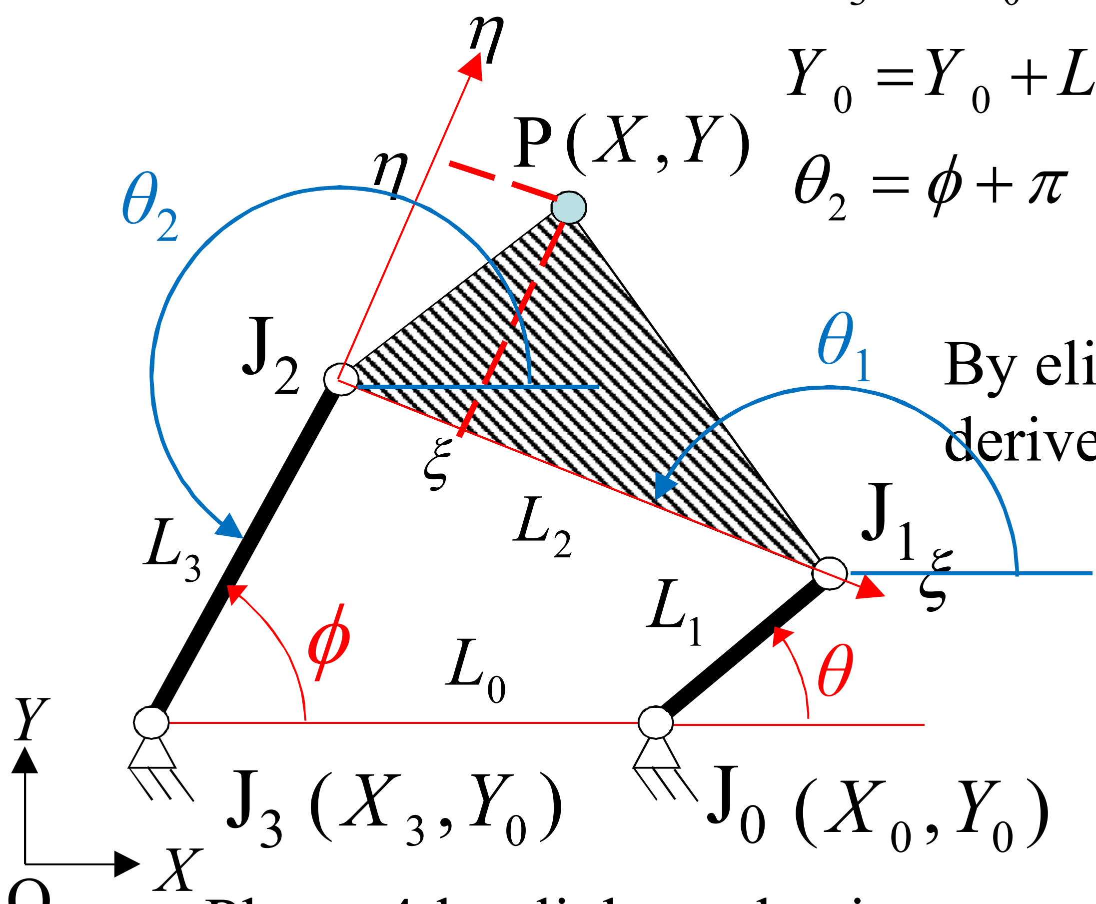
$$X_3 = X_0 + L_1 \cos\theta + L_2 \cos\theta_1 + L_3 \cos\theta_2$$

$$Y_0 = Y_0 + L_1 \sin\theta + L_2 \sin\theta_1 + L_3 \sin\theta_2$$

$$\theta_2 = \phi + \pi$$

By eliminating θ_1, θ_2, $\phi = f(\theta)$ may be derived

Complicated!



Planar 4-bar link mechanism

Let analyze output angular motion.

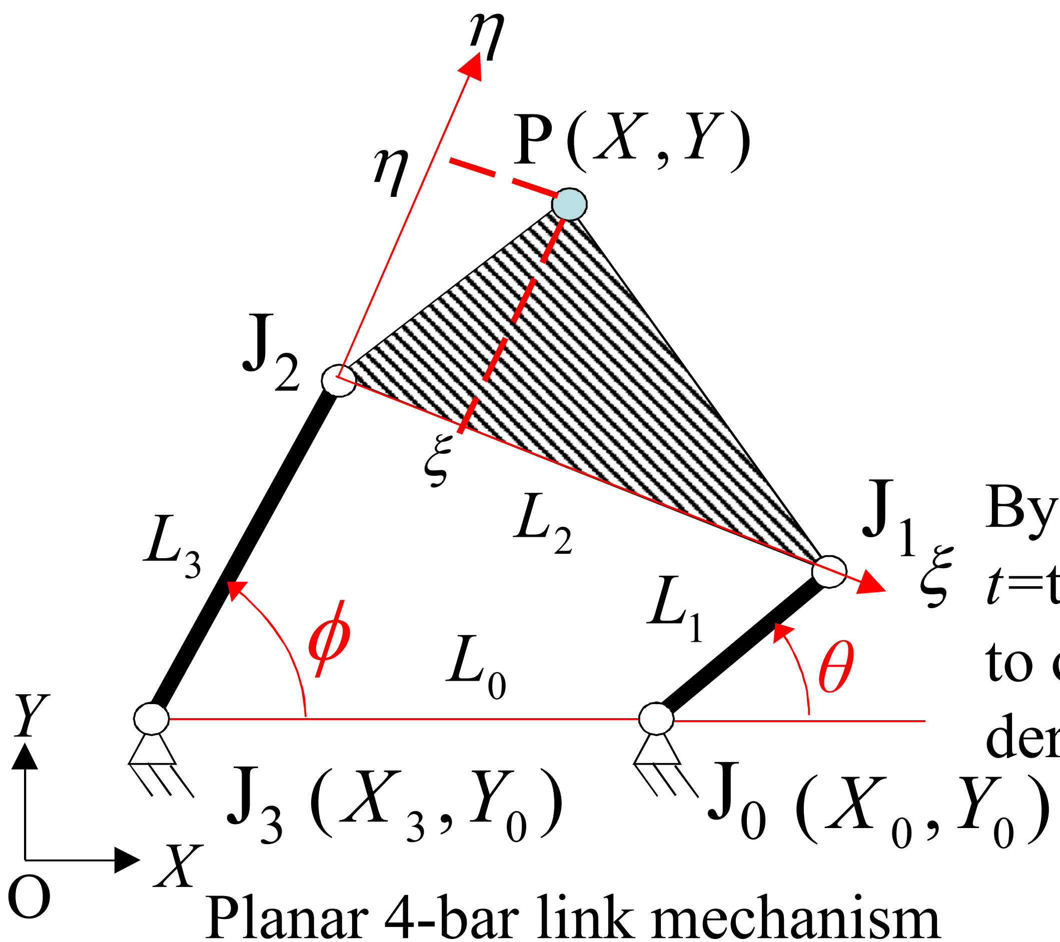
(2) From length of coupler link:

$$(X_1 - X_2)^2 + (Y_1 - Y_2)^2 = L_2^2$$

$$\begin{Bmatrix} X_1 \\ Y_1 \end{Bmatrix} = \begin{Bmatrix} L_1 \cos\theta + X_0 \\ L_1 \sin\theta + Y_0 \end{Bmatrix}$$

$$\begin{Bmatrix} X_2 = L_3 \cos\phi + X_3 \\ Y_2 = L_3 \sin\phi + Y_3 \end{Bmatrix}$$

By deforming equation with t=tan $\phi/2$, and solving the equation to obtain t, then $\phi = f(\theta)$ may be derived.



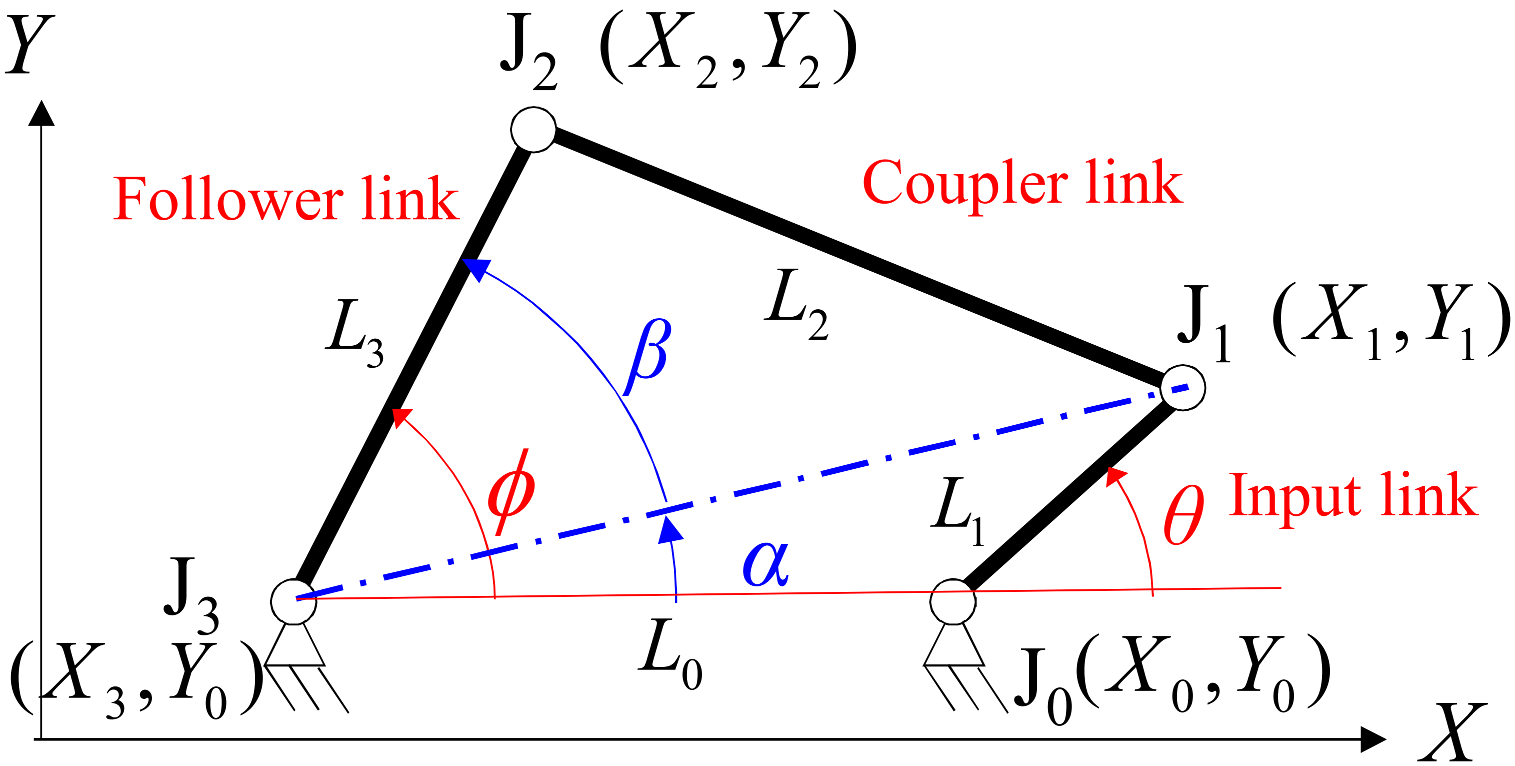
Planar 4-bar link mechanism

Complicated!

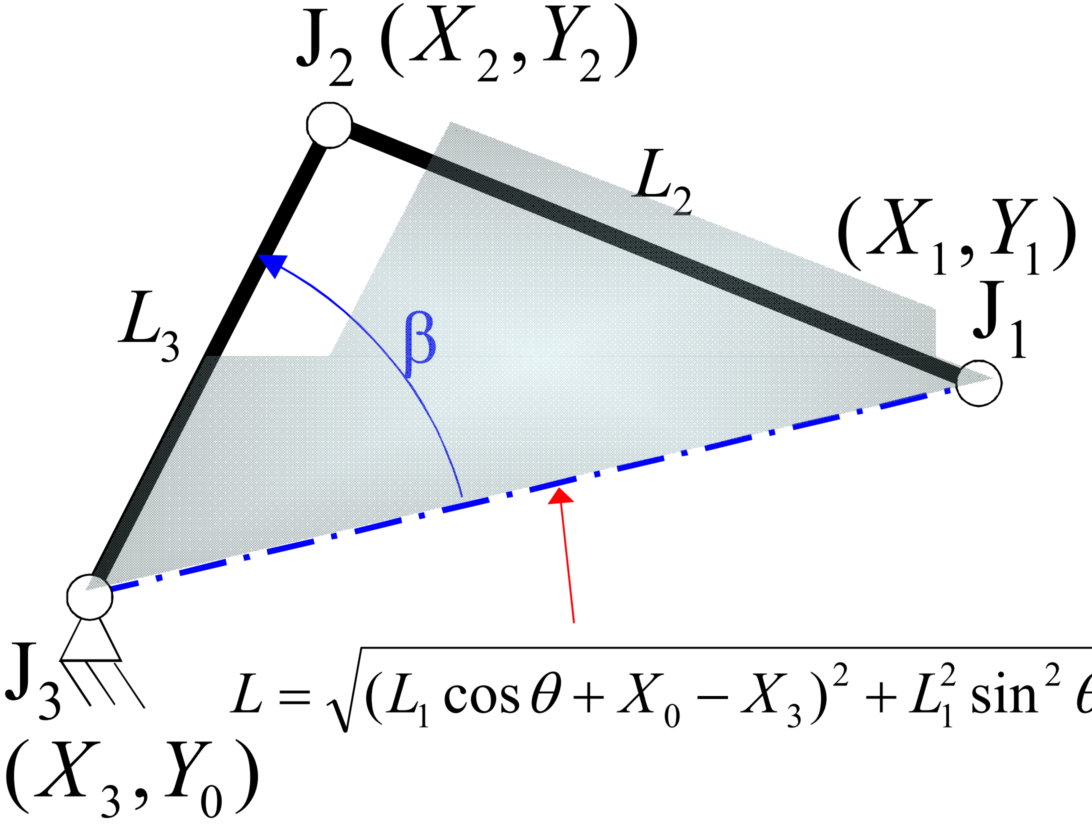
(3)Let draw an additional straight line

$$\phi = \alpha + \beta$$

Let derive α and β, respectively.



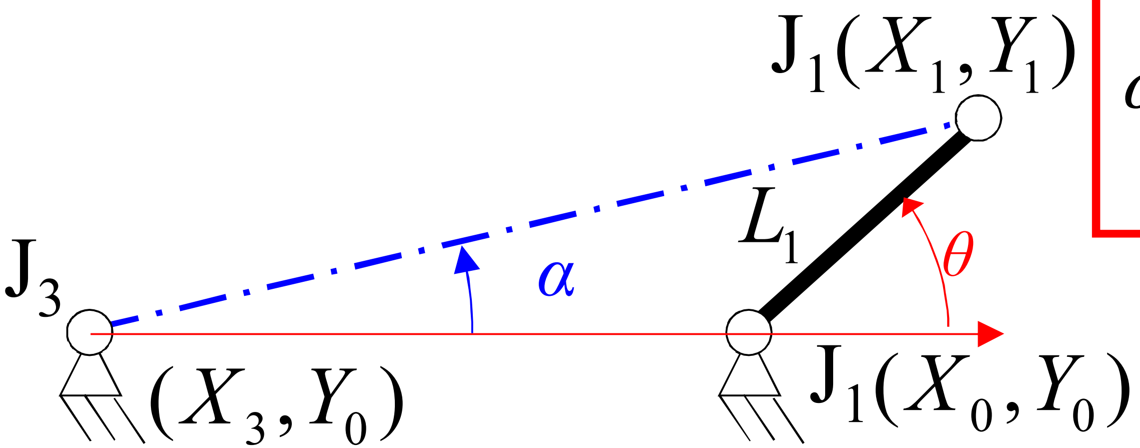
Planar 4-bar link mechanism



$$L=\sqrt{(L_1\cos\theta+X_0-X_3)^2+L_1^2\sin^2\theta}$$

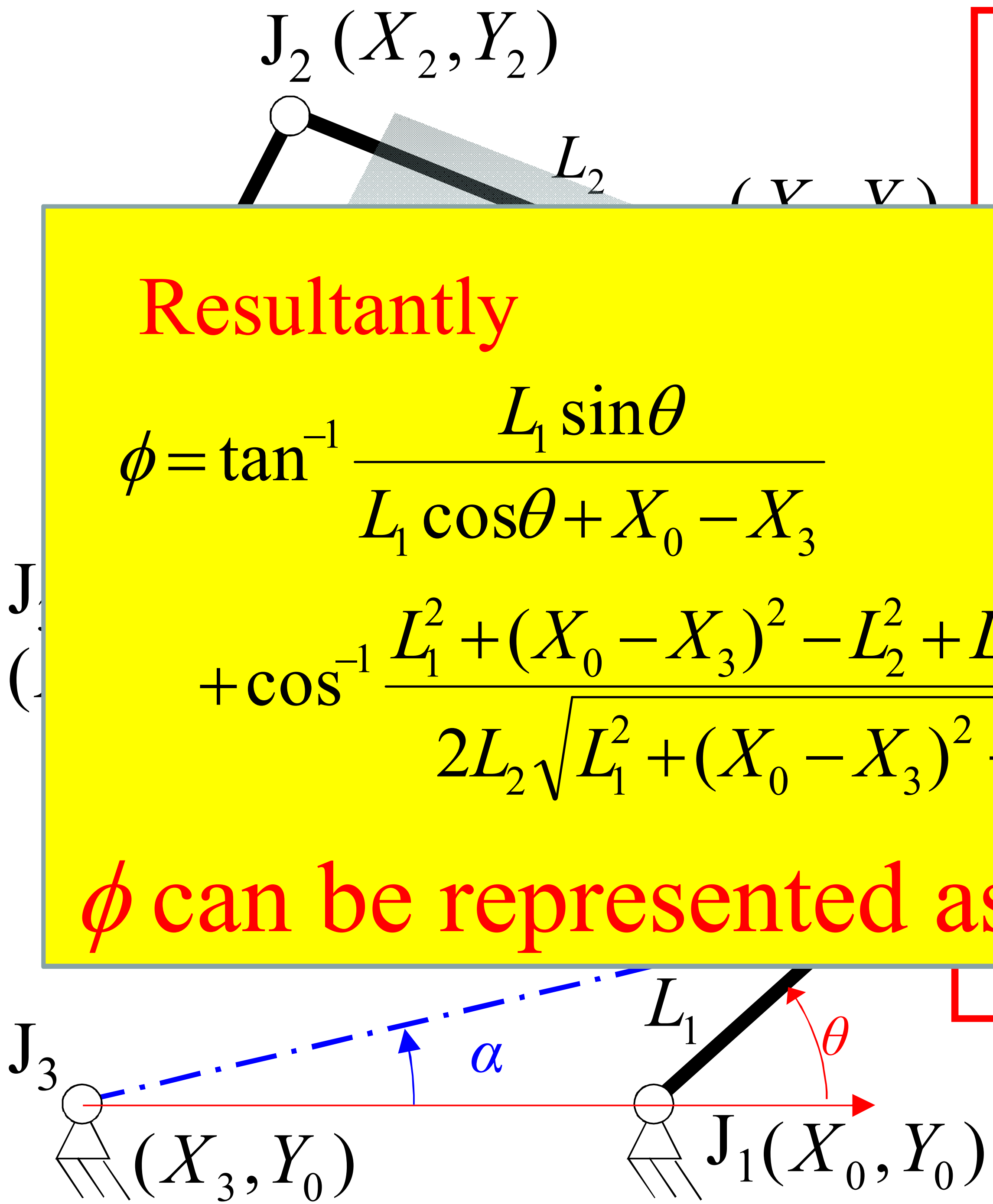
$$\beta=\cos^{-1}\frac{L^2+L_3^2-L_2^2}{2LL_3}$$

Cosine theorem



$$\alpha=\tan^{-1}\frac{L_1\sin\theta}{L_1\cos\theta+X_0-X_3}$$

Easy!



$$\beta = \cos^{-1}\frac{L^2 + L_3^2 - L_2^2}{2LL}$$

Resultantly

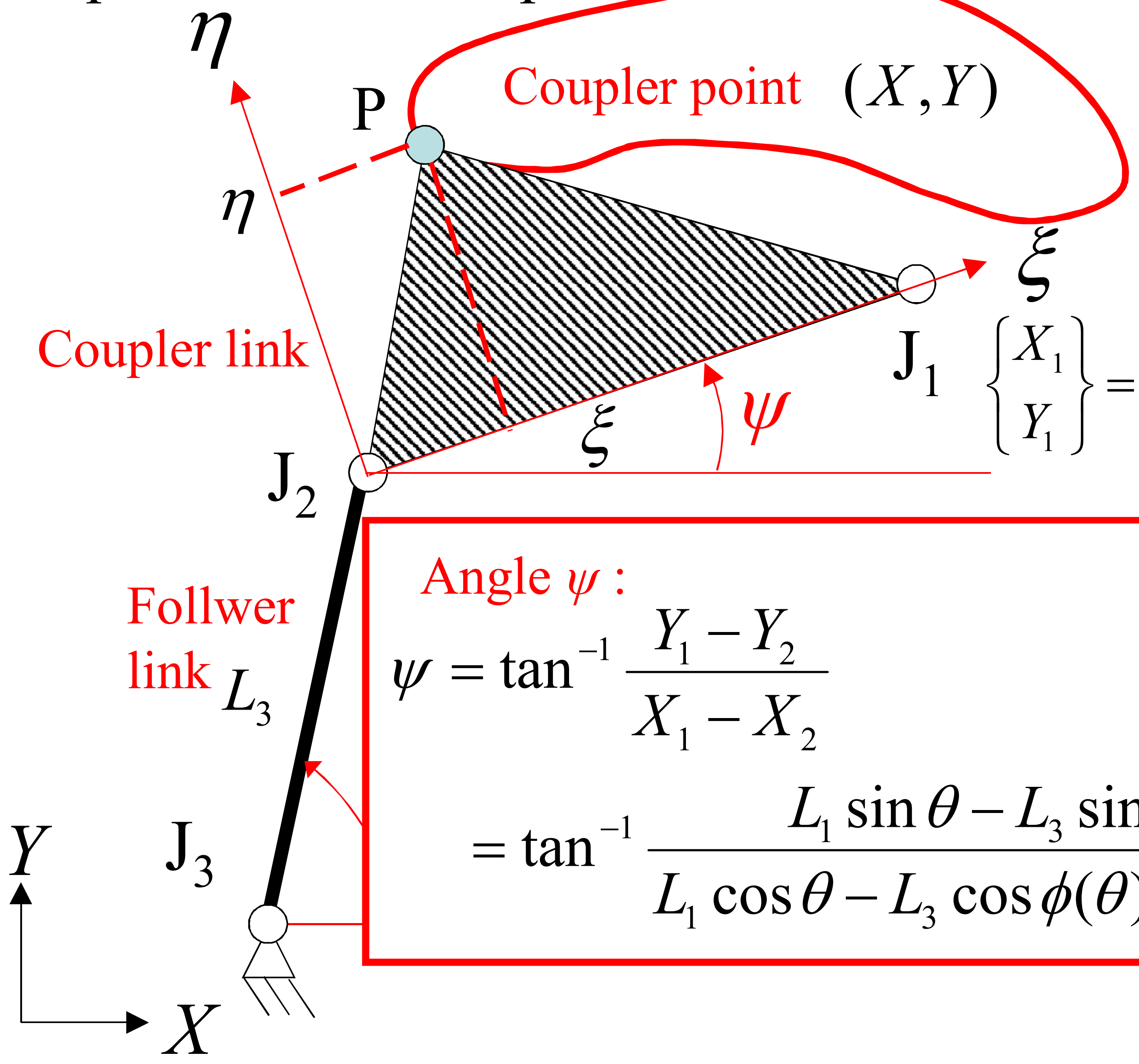
$$\phi = \tan^{-1}\frac{L_1\sin\theta}{L_1\cos\theta + X_0 - X_3}$$

$$+\cos^{-1}\frac{L_1^2 + (X_0 - X_3)^2 - L_2^2 + L_3^2 + 2L_1(X_0 - X_3)\cos\theta}{2L_2\sqrt{L_1^2 + (X_0 - X_3)^2 + 2L_1(X_0 - X_3)\cos\theta}}$$

ϕ can be represented as function of θ.

Easy!

Displacement of coupler point :

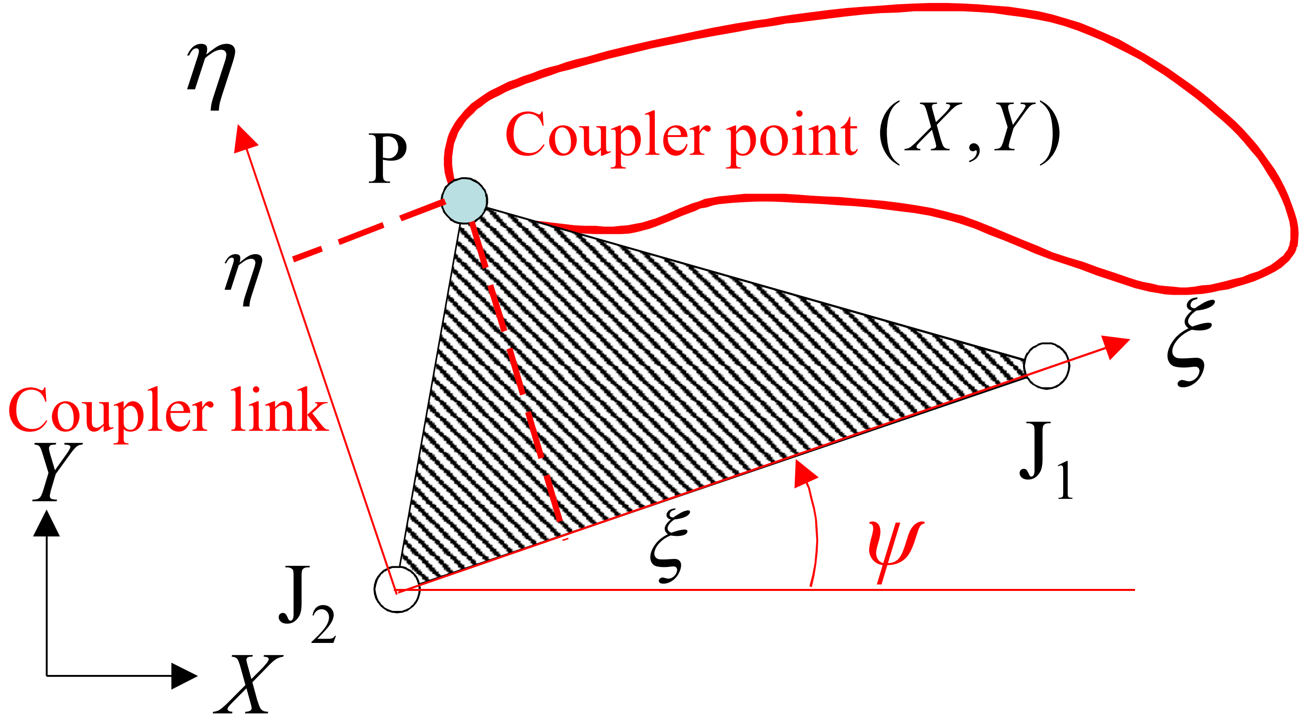


$$\begin{Bmatrix} X_1 \\ Y_1 \end{Bmatrix} = \begin{Bmatrix} L_1 \cos\theta + X_0 \\ L_1 \sin\theta + Y_0 \end{Bmatrix}$$

Angle ψ :

$$\psi = \tan^{-1} \frac{Y_1 - Y_2}{X_1 - X_2}$$

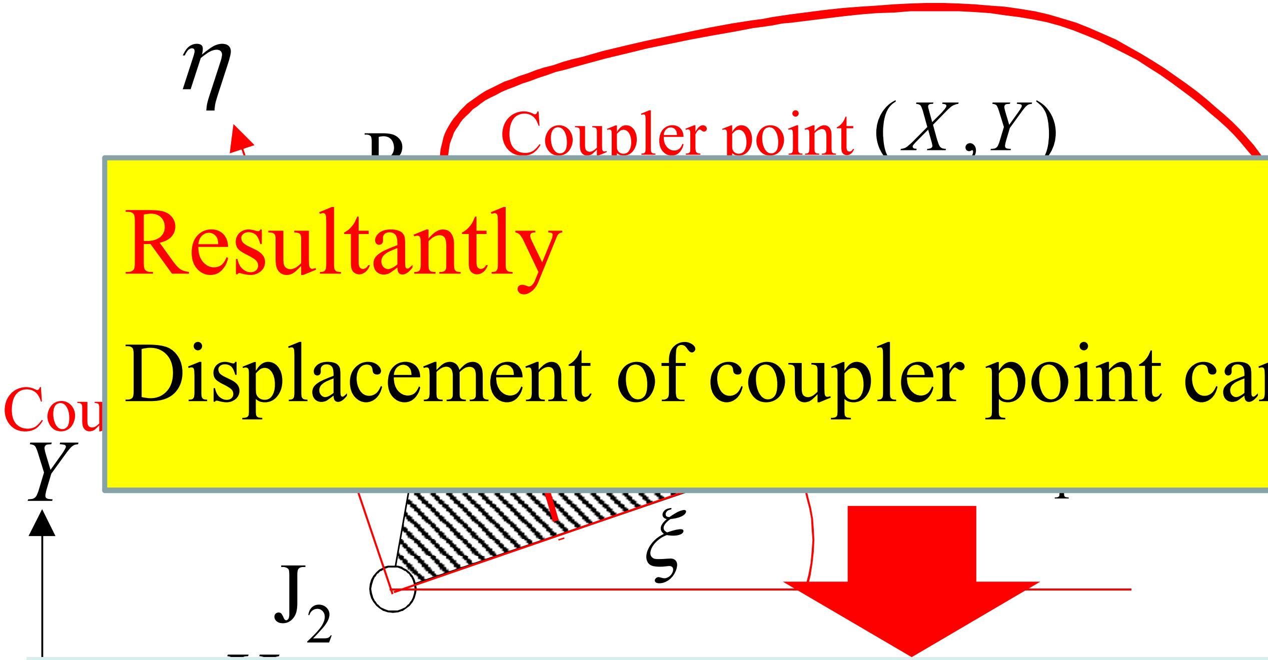
$$= \tan^{-1} \frac{L_1 \sin\theta - L_3 \sin\phi(\theta)}{L_1 \cos\theta - L_3 \cos\phi(\theta) + X_0 - X_3}$$



Coordinate transformation:

$$\begin{Bmatrix} X(\theta) \\ Y(\theta) \end{Bmatrix} = \begin{bmatrix} \cos\psi & -\sin\psi \\ \sin\psi & \cos\psi \end{bmatrix} \begin{Bmatrix} \xi \\ \eta \end{Bmatrix} + \begin{Bmatrix} X_2 \\ Y_2 \end{Bmatrix}$$

$$= \begin{Bmatrix} \xi\cos\psi - \eta\sin\psi + X_2 \\ \xi\sin\psi + \eta\cos\psi + Y_2 \end{Bmatrix}$$



Resultantly

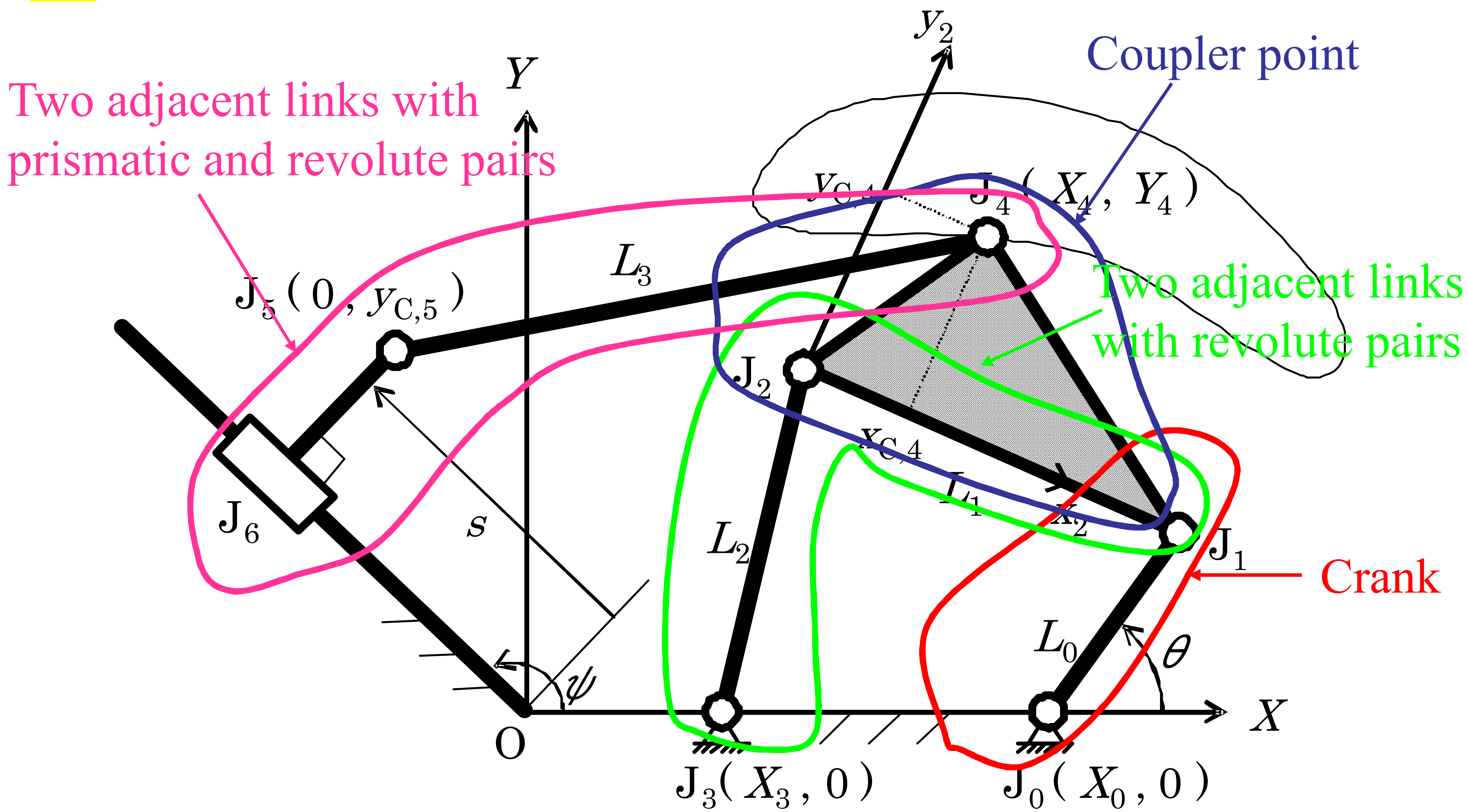
Displacement of coupler point can be derived.

However…

- ▲**Peculiar to a mechanism**
- ▲**It will be complicated to analyze velocity and acceleration.**

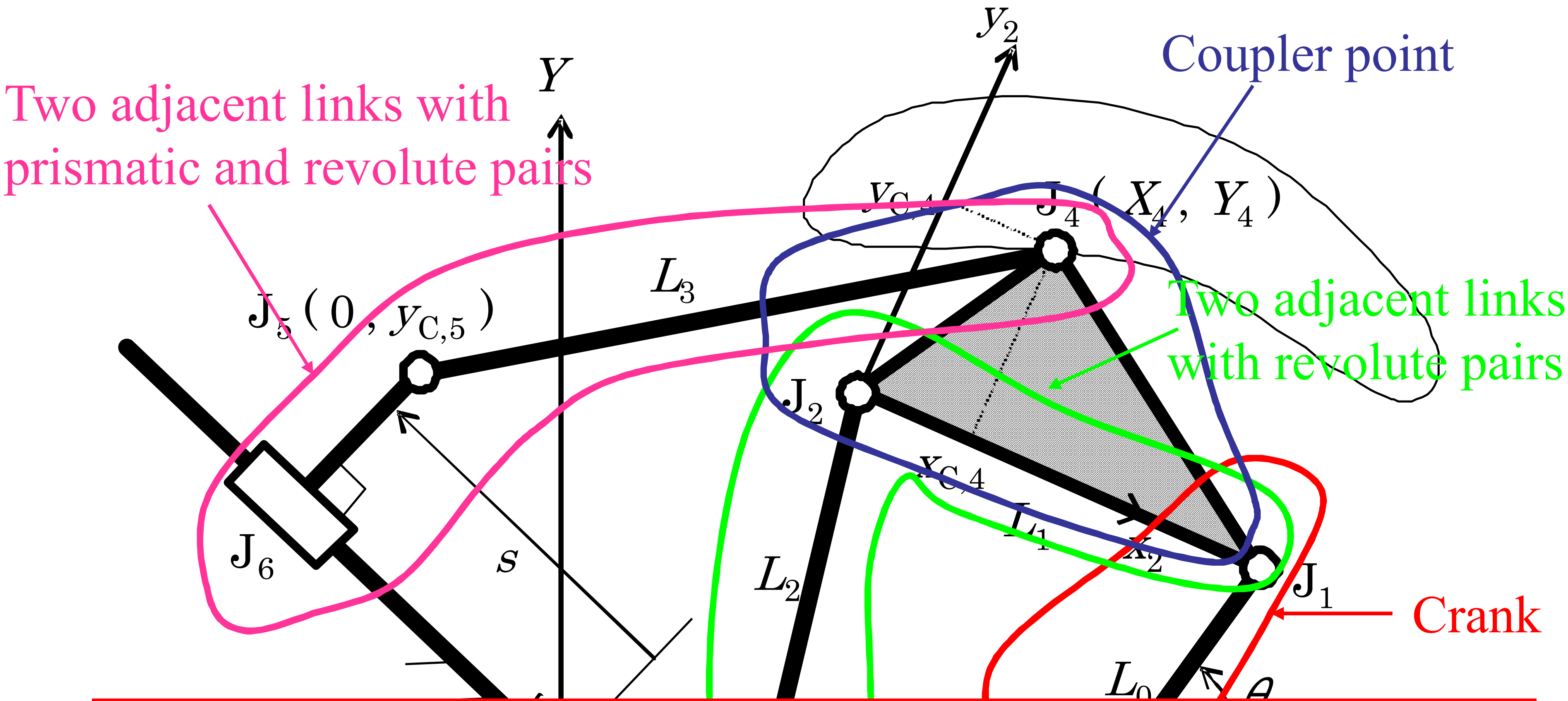
"General and systematic method is strongly required!"

4. Systematic kinematics analysis method



Crank input-Slider output planar 6-bar link mechanism

4. Systematic kinematics analysis method



A mechanism is divided into several units and each of them will be analyzed with systematic calculation.

Feature of the systematic kinematics analysis:

(1)It is easy to understand because motion of kinematic pair can be calculated in order.

(2)Applicable to various planar mechanisms with revolute and prismatic pairs

(3)Velocity and acceleration can be calculated.

(4)Fundamentals to analyze dynamics of mechanism

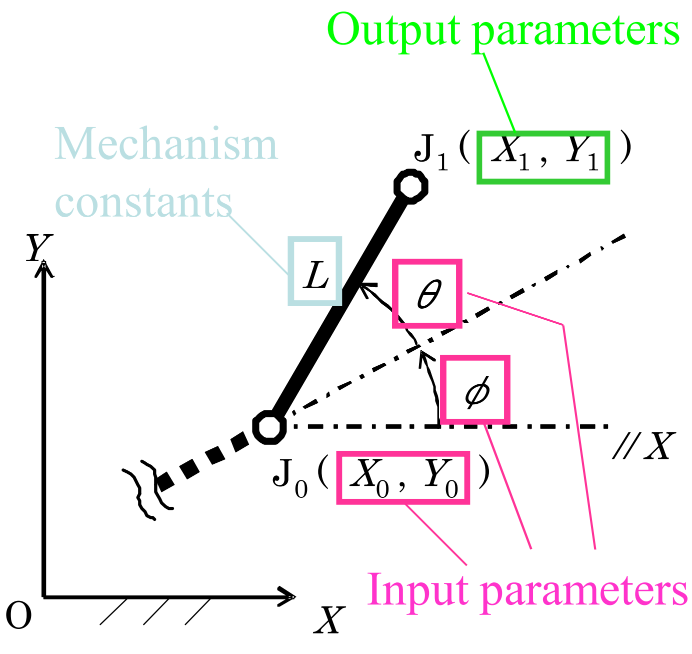
"General, systematic and available method"

→***Calculation program can be offered.***

4.1 Derivation of systematic kinematic analysis

(1)Motion of crank link : crank_input

To consider the posture of front link



Crank link

$$\begin{Bmatrix} X_1 \\ Y_1 \end{Bmatrix} = \begin{Bmatrix} L\cos(\theta+\phi)+X_0 \\ L\sin(\theta+\phi)+Y_0 \end{Bmatrix}$$

Differentiating

$$\begin{Bmatrix} \dot{X}_1 \\ \dot{Y}_1 \end{Bmatrix} = \begin{Bmatrix} -L(\dot\theta+\dot\phi)\sin(\theta+\phi)+\dot{X}_0 \\ L(\dot\theta+\dot\phi)\cos(\theta+\phi)+\dot{Y}_0 \end{Bmatrix}$$

$$\begin{Bmatrix} \ddot{X}_1 \\ \ddot{Y}_1 \end{Bmatrix} = \begin{Bmatrix} -L[(\ddot\theta+\ddot\phi)\sin(\theta+\phi)+(\dot\theta+\dot\phi)^2\cos(\theta+\phi)]+\ddot{X}_0 \\ L[(\ddot\theta+\ddot\phi)\cos(\theta+\phi)-(\dot\theta+\dot\phi)^2\sin(\theta+\phi)]+\ddot{Y}_0 \end{Bmatrix}$$

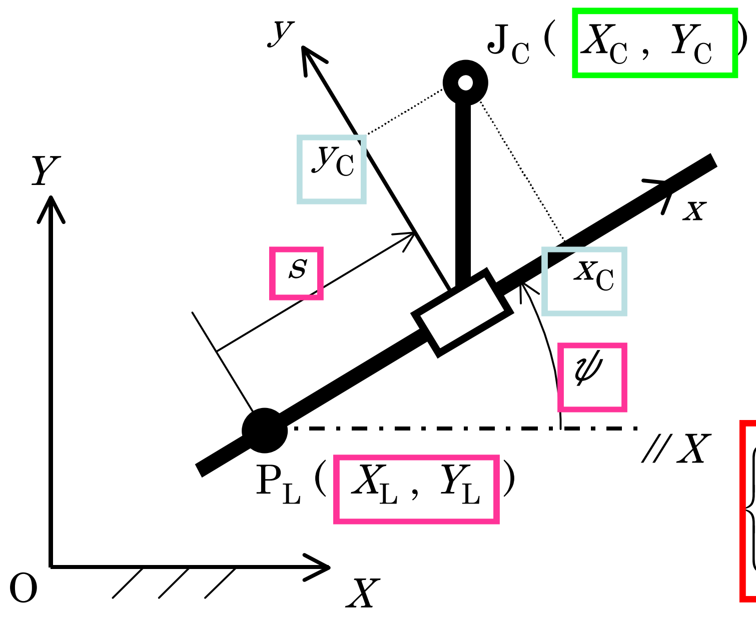
Position and posture of pair J_0 →Crank angle →Position of pair J_1

(X_0, Y_0) , ϕ → θ → (X_1, Y_1)

From a pair to a pair

Velocity and acceleration

(2)Motion of input slider : slider_input



Input slider link

Posture of slider axis

$$\begin{Bmatrix} X_S \\ Y_S \end{Bmatrix} = \begin{Bmatrix} s\cos\psi + X_L \\ s\sin\psi + Y_L \end{Bmatrix}$$

$$\begin{Bmatrix} \dot{X}_S \\ \dot{Y}_S \end{Bmatrix} = \begin{Bmatrix} \dot{s}\cos\psi - s\dot{\psi}\sin\psi + \dot{X}_L \\ \dot{s}\sin\psi + s\dot{\psi}\cos\psi + \dot{Y}_L \end{Bmatrix}$$

$$\begin{Bmatrix} \ddot{X}_S \\ \ddot{Y}_S \end{Bmatrix} = \begin{Bmatrix} \ddot{s}\cos\psi - 2\dot{s}\dot{\psi}\sin\psi - s\ddot{\psi}\sin\psi - s\dot{\psi}^2\cos\psi + \ddot{X}_L \\ \ddot{s}\sin\psi + 2\dot{s}\dot{\psi}\cos\psi + s\ddot{\psi}\cos\psi - s\dot{\psi}^2\sin\psi + \ddot{Y}_L \end{Bmatrix}$$

$$\begin{Bmatrix} X_C \\ Y_C \end{Bmatrix} = \begin{Bmatrix} x_C\cos\psi - y_C\sin\psi + X_S \\ x_C\sin\psi + y_C\cos\psi + Y_S \end{Bmatrix}$$

Position of pair J_C

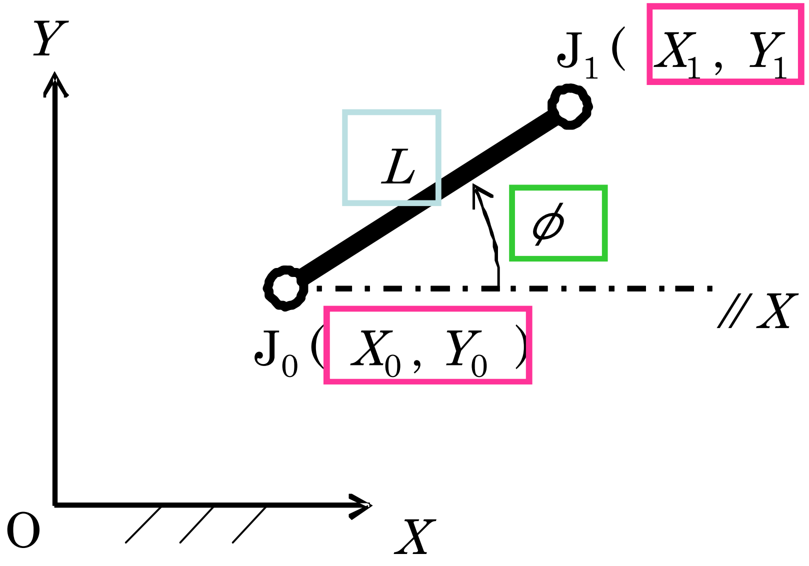
$$\begin{Bmatrix} \dot{X}_C \\ \dot{Y}_C \end{Bmatrix} = \begin{Bmatrix} -x_C\dot{\psi}\sin\psi - y_C\dot{\psi}\cos\psi + \dot{X}_S \\ x_C\dot{\psi}\cos\psi - y_C\dot{\psi}\sin\psi + \dot{Y}_S \end{Bmatrix}$$

$$\begin{Bmatrix} \ddot{X}_C \\ \ddot{Y}_C \end{Bmatrix} = \begin{Bmatrix} -x_C(\ddot{\psi}\sin\psi + \dot{\psi}^2\cos\psi) - y_C(\ddot{\psi}\cos\psi - \dot{\psi}^2\sin\psi) + \ddot{X}_S \\ x_C(\ddot{\psi}\cos\psi - \dot{\psi}^2\sin\psi) - y_C(\ddot{\psi}\sin\psi + \dot{\psi}^2\cos\psi) + \ddot{Y}_S \end{Bmatrix}$$

Position and posture of slider axis→Translational input →Position of pair J_C

(X_L, Y_L), ψ → s → (X_C, Y_C)

(3)Angular motion of link : link_angle



Angular motion of link

$$\phi = \tan^{-1}\frac{\Delta Y}{\Delta X}$$

Posture angle Is represented with position of pairs

$$\dot{\phi} = \frac{\Delta\dot{Y}\Delta X - \Delta Y\Delta\dot{X}}{L^2}$$

$$\ddot{\phi} = \frac{\Delta\ddot{Y}\Delta X - \Delta Y\Delta\ddot{X}}{L^2}$$

where

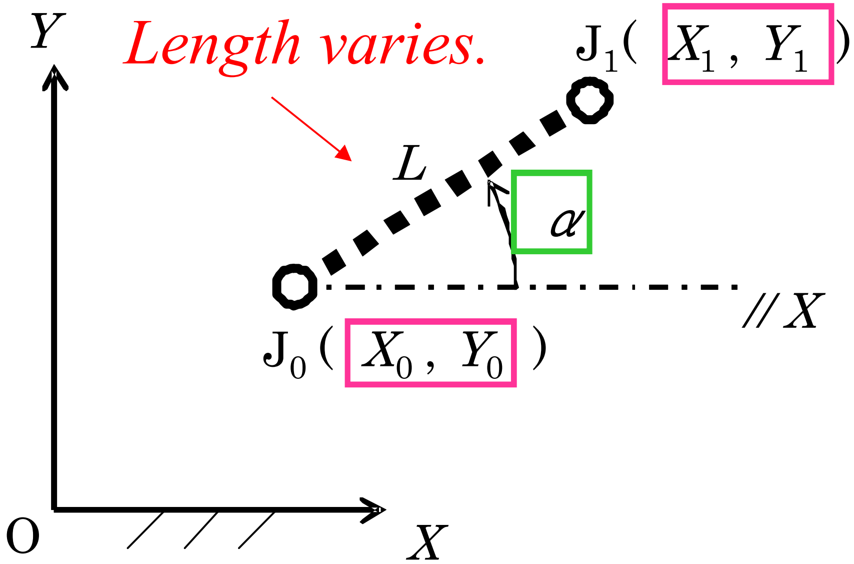
$$\Delta X = X_1 - X_0, \Delta\dot{X} = \dot{X}_1 - \dot{X}_0, \Delta\ddot{X} = \ddot{X}_1 - \ddot{X}_0,$$

$$\Delta Y = Y_1 - Y_0, \Delta\dot{Y} = \dot{Y}_1 - \dot{Y}_0, \Delta\ddot{Y} = \ddot{Y}_1 - \ddot{Y}_0$$

Position of pairs at both ends → Posture of link

(X_0, Y_0), (X_1, Y_1) → ϕ

(4)Angular motion of segment : segment_angle



Segment connecting two points

$$\alpha = \tan^{-1}\frac{\Delta Y}{\Delta X}$$

Let consider change of distance

$$\dot{\alpha} = \frac{\Delta\dot{Y}\Delta X - \Delta Y\Delta\dot{X}}{\Delta X^2}\cos^2\alpha$$

$$\ddot{\alpha} = \{[\Delta X(\Delta\ddot{Y}\Delta X - \Delta Y\Delta\ddot{X}) - 2\Delta\dot{X}(\Delta\dot{Y}\Delta X - \Delta Y\Delta\dot{X})]\cos\alpha - 2\dot{\alpha}(\Delta\dot{Y}\Delta X - \Delta Y\Delta\dot{X})\Delta X\sin\alpha\} \cdot \cos\alpha / \Delta X^3$$

where

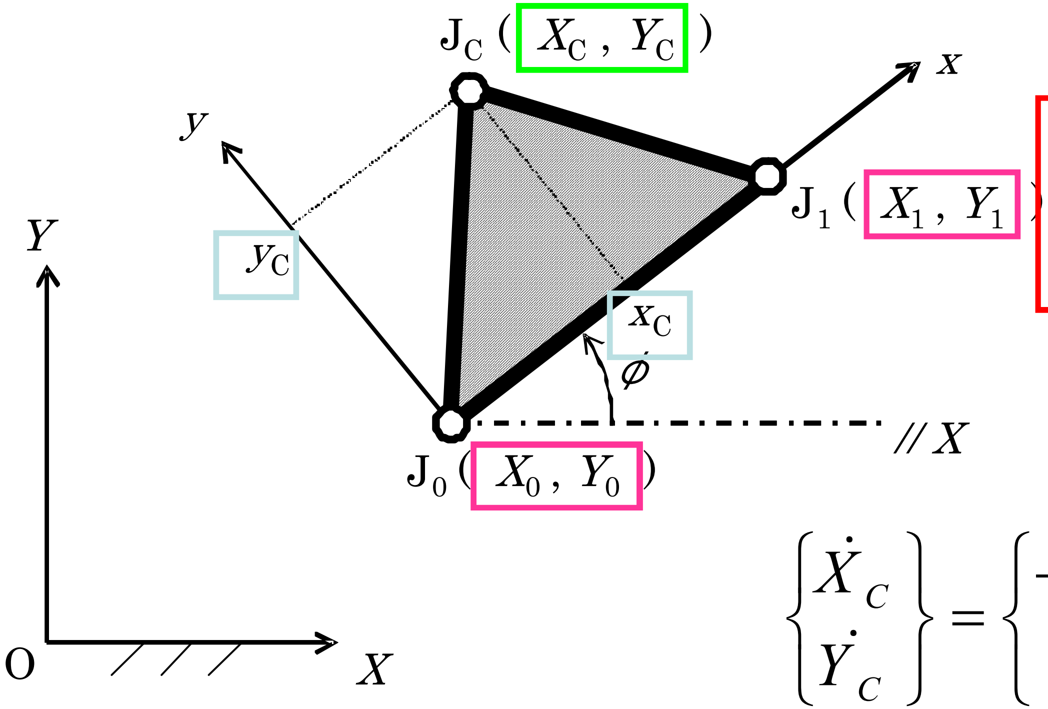
$$\Delta X = X_1 - X_0, \Delta\dot{X} = \dot{X}_1 - \dot{X}_0, \Delta\ddot{X} = \ddot{X}_1 - \ddot{X}_0,$$

$$\Delta Y = Y_1 - Y_0, \Delta\dot{Y} = \dot{Y}_1 - \dot{Y}_0, \Delta\ddot{Y} = \ddot{Y}_1 - \ddot{Y}_0$$

Position of two points → Posture of segment

(X_0, Y_0), (X_1, Y_1) → α

(5)Motion of coupler point : coupler_point



Point on a link

Coordinate transformation due to rotation

$$\begin{Bmatrix} X_C \\ Y_C \end{Bmatrix} = \begin{Bmatrix} x_C \cos\phi - y_C \sin\phi + X_0 \\ x_C \sin\phi + y_C \cos\phi + Y_0 \end{Bmatrix}$$

ϕ can be calculated with link_angle

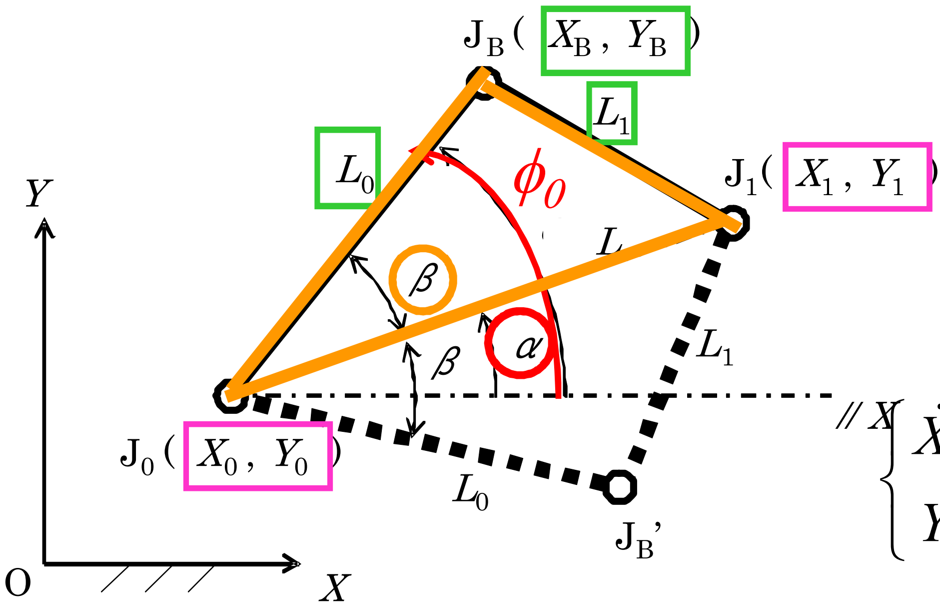
$$\begin{Bmatrix} \dot{X}_C \\ \dot{Y}_C \end{Bmatrix} = \begin{Bmatrix} -\dot{\phi}(x_C \sin\phi + y_C \cos\phi) + \dot{X}_0 \\ \dot{\phi}(x_C \cos\phi - y_C \sin\phi) + \dot{Y}_0 \end{Bmatrix}$$

$$\begin{Bmatrix} \ddot{X}_C \\ \ddot{Y}_C \end{Bmatrix} = \begin{Bmatrix} -\ddot{\phi}(x_C \sin\phi + y_C \cos\phi) - \dot{\phi}^2(x_C \cos\phi - y_C \sin\phi) + \ddot{X}_0 \\ \ddot{\phi}(x_C \cos\phi - y_C \sin\phi) - \dot{\phi}^2(x_C \sin\phi + y_C \cos\phi) + \ddot{Y}_0 \end{Bmatrix}$$

Position of pairs J_0, J_1 → Position of pair J_C

(X_0, Y_0), (X_1, Y_1) → (X_C, Y_C)

(6)Motion of two adjacent links with only revolute pairs：RRR_links



Two adjacent links with only revolute pai…

$$\begin{Bmatrix} X_B \\ Y_B \end{Bmatrix} = \begin{Bmatrix} L_0 \cos\phi_0 + X_0 \\ L_0 \sin\phi_0 + Y_0 \end{Bmatrix}$$

By giving posture angle ϕ_0 position of pair J_B can be calculated.

$$\begin{Bmatrix} \dot X_B \\ \dot Y_B \end{Bmatrix} = \begin{Bmatrix} -L_0\dot\phi_0 \sin\phi_0 + \dot X_0 \\ L_0\dot\phi_0 \cos\phi_0 + \dot Y_0 \end{Bmatrix}$$

$$\begin{Bmatrix} \ddot X_B \\ \ddot Y_B \end{Bmatrix} = \begin{Bmatrix} \cdots + \ddot X_0 \\ \cdots + \ddot Y_0 \end{Bmatrix}$$

Sign corresponds to mechanical inversion.

Calculated with segment angle

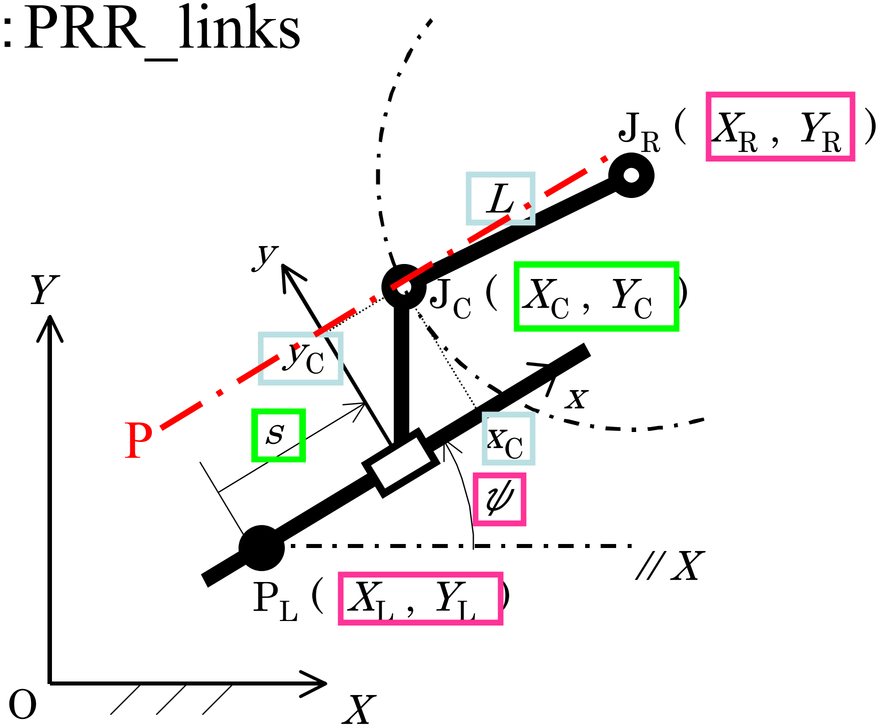
where

$$\phi_0 = \alpha \pm \beta, \quad \dot\phi_0 = \dot\alpha \pm \dot\beta, \quad \ddot\phi_0 = \ddot\alpha \pm \ddot\beta$$

Position of pairs J_0, J_1 → Position of pair J_B

(X_0, Y_0), (X_1, Y_1) → (X_B, Y_B)

(7)Motion of two adjacent links with prismatic and revolute pairs :PRR_links



Two adjacent links with prismatic and revolute pairs

$$\begin{Bmatrix} X_C \\ Y_C \end{Bmatrix} = \begin{Bmatrix} (s + x_C)\cos\psi - y_C \sin\psi + X_L \\ (s + x_C)\sin\psi + y_C \cos\psi + Y_L \end{Bmatrix}$$

$$(X_C - X_R)^2 + (Y_C - Y_R)^2 = L^2$$

Pair J_C is located at the intersection between straight line P and a circle with radius L

Displacement of slider:

$$s = -A\cos\psi - B\sin\psi - x_C \pm \sqrt{D}$$

where

$$A = -y_C \sin\psi + X_L - X_R$$

$$B = y_C \cos\psi + Y_L - Y_R$$

$$D = L^2 - (A\sin\psi - B\cos\psi)^2$$

where

$$\dot{s} = -\dot{A}\cos\psi + A\dot{\psi}\sin\psi - \dot{B}\sin\psi - B\dot{\psi}\cos\psi \pm \frac{\dot{D}}{2\sqrt{D}}$$

$$\ddot{s} = -\ddot{A}\cos\psi + 2\dot{A}\dot{\psi}\sin\psi + A\ddot{\psi}\sin\psi + A\dot{\psi}^2\cos\psi$$

$$-\ddot{B}\sin\psi - 2\dot{B}\dot{\psi}\cos\psi - B\ddot{\psi}\cos\psi + B\dot{\psi}^2\sin\psi \pm \frac{2\ddot{D}D - \dot{D}^2}{4D\sqrt{D}}$$

$$\dot{D} = -2(\dot{A}\sin\psi + A\dot{\psi}\cos\psi - \dot{B}\cos\psi + B\dot{\psi}\sin\psi)(A\sin\psi - B\cos\psi)$$

$$\dot{A} = -y_C\dot{\psi}\cos\psi + \dot{X}_L - \dot{X}_R \ , \dot{B} = -y_C\dot{\psi}\sin\psi + \dot{Y}_L - \dot{Y}_R \ , \ldots$$

Velocity and acceleration can be calculated with slider_input

Position and posture of slider axis, position of pair

J_R (X_L, Y_L), ψ, (X_R, Y_R)

↓

Position of pair J_C, Displacement of slider

(X_C, Y_C) , s

```
//------------------------------------------------------------
//
//        Kinematics Calculation
//
//          < mech.cpp >
//
//
//      2002. 8.14 Created by D. Matsuuura
//      2003.10.13 Improved by D. Matusuura
//      2008. 1.12 Improved by D. Matsuuura
//      2008. 1.14 Improved by N. Iwatsuki
//      2010. 8.21 Improved by N. Iwatsuki
//      2011.10.10 Improved by N. Iwatsuki
//      2018. 4.15 Improved by N. Iwatsuki
//
//------------------------------------------------------------

#include <stdafx.h>
#include <stdio.h>
#include <math.h>
#include "mech.h"
#include "matrix.h"

//------------------------------------------------------------
//
//    Input slider motion
//
//   pg       : Displacement, velocity and acceleration of a point on
//              a linear guide to describe position of the linear guide
//   psig     : Angular displacement, velocity and acceleration of linear guide
//   s        : Displacement, velocity and acceleration of an input slider
//   gsi, eta : Moving coordinate of a pair on a slider
//   jp       : Displacement, velocity and acceleration of a slider
//   jc       : Displacement, velocity and acceleration of a pair on a slider
//
//------------------------------------------------------------

void crank_input( POINT j1, double l, ANGLE theta, ANGLE phi, POINT *j2 )
{
        double thephi, dthephi, ddthephi;
        double c,s;

        thephi = theta.the + phi.the;
        dthephi = theta.dthe + phi.dthe;
        ddthephi = theta.ddthe + phi.ddthe;

        c = cos( thephi );
        s = sin( thephi );

        j2->P.x   = j1.P.x + l*c;
        j2->DP.x  = -l*s*dthephi;
        j2->DDP.x = -l*( s*ddthephi + c*dthephi*dthephi );

        j2->P.y   = j1.P.y + l*s;
        j2->DP.y  = l*c*dthephi;
        j2->DDP.y = l*( c*ddthephi  - s*dthephi*dthephi );
}
```

```
//------------------------------------------------------------
//
//      Input crank motion
//
//   j1      : Displacement, velocity and acceleration of root pair
//   l       : Link length
//   theta   : Angular displacement, velocity and acceleration of crank
//   phi     : Angular displacement, velocity and acceleration of root link
//   j2      : Displacement, velocity and acceleration of tip pair
//
//------------------------------------------------------------

void slider_input( POINT pg, double l, ANGLE psig,  VARIABLE s,
                        double gsi, double eta,  POINT *jp, POINT *jc )
{
        double cp, sp;

        cp = cos( psig.the );
        sp = sin( psig.the );

        jp->P.x   = s.v*cp + pg.P.x;
        jp->DP.x  = s.dv*cp - s.v*sp*psig.dthe + pg.DP.x;
        jp->DDP.x = s.ddv*cp - 2.0*s.dv*sp*psig.dthe - s.v*sp*psig.ddthe
                              - s.v*cp*psig.dthe*psig.dthe + pg.DDP.x;

        jp->P.y   = s.v*sp + pg.P.y;
        jp->DP.y  = s.dv*sp + s.v*cp*psig.dthe + pg.DP.y;
        jp->DDP.y = s.ddv*sp + 2.0*s.dv*cp*psig.dthe + s.v*cp*psig.ddthe
                              - s.v*sp*psig.dthe*psig.dthe + pg.DDP.y;

        jc->P.x   =  gsi*cp - eta*sp + jp->P.x;
        jc->DP.x  = -gsi*sp*psig.dthe - eta*cp*psig.dthe + jp->DP.x;
        jc->DDP.x = -gsi*( sp*psig.ddthe + cp*psig.dthe*psig.dthe )
                         - eta*( cp*psig.ddthe - sp*psig.dthe*psig.dthe ) + jp->DDP.x;

        jc->P.y   =  gsi*sp + eta*cp + jp->P.y;
        jc->DP.y  =  gsi*cp*psig.dthe - eta*sp*psig.dthe + jp->DP.y;
        jc->DDP.y =  gsi*( cp*psig.ddthe - sp*psig.dthe*psig.dthe )
                         - eta*( sp*psig.ddthe + cp*psig.dthe*psig.dthe ) + jp->DDP.y;
}
```

C language program of systematic kinematics analysis

```c
//------------------------------------------------------------
//
//     Angular motion of a segment connecting two points
//
//   j1, j2    : Displacement, velocity and acceleration of each point
//   *angle    : Angular displacement, velocity and acceleration of
//              a segment connecting two points
//   Function value: Complete/incomplete(Two points are quite same)
//
//------------------------------------------------------------

int segment_angle( POINT j1, POINT j2, ANGLE *angle )
{
        POINT diff;
        double c, s;
        double absx, absy;
        double q1, q2;
        double r, dr;

        get_diff( j1, j2, &diff );
        absx = fabs( diff.P.x );
        absy = fabs( diff.P.y );
        if( absx < 10e-20 && absy < 10e-20 ){
                angle->the   = 0.0;
                angle->dthe  = 0.0;
                angle->ddthe = 0.0;
                printf("++ Error in segment_angle ¥n");
                return ERROR;
        }
        angle->the = atan2( diff.P.y, diff.P.x );
                c = cos( angle->the );
                s = sin( angle->the );

                if( absy > 10e-20 && absx/absy < 10e-10 ){
                        r   = sqrt( diff.P.x*diff.P.x + diff.P.y*diff.P.y );
                        dr  = ( diff.DP.x*diff.P.x + diff.DP.y*diff.P.y )/r;
                        if( s >= 0.0 ){
                                angle->dthe  = -diff.DP.x/r;
                                angle->ddthe = ( -diff.DDP.x - 2.0*dr*angle->dthe )/r;
                                return SUCCESS;
                        }
                        else{
                                angle->dthe  = diff.DP.x/r;
                                angle->ddthe = ( diff.DDP.x - 2.0*dr*angle->dthe )/r;
                                return SUCCESS;
                        }
                }
                else{
                        q1 = diff.DP.y*diff.P.x - diff.P.y*diff.DP.x;
                        q2 = diff.DDP.y*diff.P.x - diff.P.y*diff.DDP.x;
                        angle->dthe = q1*c*c/(diff.P.x*diff.P.x);
                        angle->ddthe = ( (diff.P.x*q2 - 2.0*diff.DP.x*q1)*c
                         - 2.0*diff.P.x*angle->dthe*q1*s)*c/( diff.P.x*diff.P.x*diff.P.x );
                        return SUCCESS;
                }
}
```

```c
//------------------------------------------------------------
//
//     Motion of Adjacent Two links Connecting Revolute Pairs
//
//   j1, j2    : Displacement, velocity and acceleration of pairs
//               at both ends
//   l1, l2    : Link lengths
//   minv      : Indicator for mechanical inversion
//   *jb       : Displacement, velocity and acceleration of a pair which connects
//               two links
//   Function value: Complete/incomplete(Link chain cannot be formed)
//
//------------------------------------------------------------

int RRR_links( POINT j1, POINT j2, double l1, double l2, int minv, POINT *jb )
{
        POINT diff;
        double q2;
        double q, dq, ddq;

        double cb, sb, ss;
        ANGLE angle;
        double phi, dphi, ddphi;
        double bet, dbet, ddbet;
        double cp, sp;
        int ier;

        get_diff( j1, j2, &diff );
        q2 = diff.P.x*diff.P.x + diff.P.y*diff.P.y;

        if( q2 <= 10e-30 ){
                printf("++ Error in RRR_links (2対偶が一致してます.) ¥n");
                return ERROR;
        }

        q = sqrt( q2 );
        dq = ( diff.DP.x*diff.P.x + diff.DP.y*diff.P.y )/q;
        ddq = ( ( diff.DDP.x*diff.P.x + diff.DP.x*diff.DP.x
             + diff.DDP.y*diff.P.y + diff.DP.y*diff.DP.y )*q
             - ( diff.DP.x*diff.P.x + diff.DP.y*diff.P.y )*dq )/q2;

        cb = ( l1*l1 + q2 - l2*l2 )/( 2.0*l1*q );
        if( fabs( cb ) > 1.0 ){
                printf("++ Error in RRR_links (連鎖が構成できません. )¥n");
                return ERROR;
        }
        sb = sqrt( 1.0 - cb*cb );
        ss = l1*l1 - l2*l2 - q2;

        ier = segment_angle( j1, j2, &angle );
        if( ier != SUCCESS ){
                printf("++ Error in RRR_links (2対偶が一致してます.) ¥n");
                return ERROR;
        }
        bet = acos( cb );
        dbet = ss*dq/( 2.0*l1*q2*sb );
```

C language program of systematic kinematics analysis

```
        ddbet = ( ( ss*( ddq*q - 2.0*dq*dq )
                - 2.0*( dq*q )*( dq*q) )*sb - ss*dq*q*dbet*cb )/( 2.0*l1*q2*q*sb*sb );

        if( minv > 0 ) {
                phi = angle.the + bet;
                dphi = angle.dthe + dbet;
                ddphi = angle.ddthe + ddbet;
        }
        else {
                phi = angle.the - bet;
                dphi = angle.dthe - dbet;
                ddphi = angle.ddthe - ddbet;
        }

        cp = cos( phi );
        sp = sin( phi );

        jb->P.x = j1.P.x + l1*cp;
        jb->P.y = j1.P.y + l1*sp;
        jb->DP.x = j1.DP.x - l1*dphi*sp;
        jb->DP.y = j1.DP.y + l1*dphi*cp;
        jb->DDP.x = j1.DDP.x - l1*( ddphi*sp + dphi*dphi*cp );
        jb->DDP.y = j1.DDP.y + l1*( ddphi*cp - dphi*dphi*sp );

        return SUCCESS;
}
```

```
//-----------------------------------------------------------
//
//    Motion of Adjacent Two links with a Prismatic Pair and
//       Two Revolute Pairs
//
//   jr       : Displacement, velocity and acceleration of revolute
//              pair at end
//   l        : Link legth between two revolute pairs
//   pg       : Displacement, velocity and acceleration of a point on
//              a linear guide to describe position of the linear guide
//   psig     : Angular displacement, velocity and acceleration of linear guide
//   gsi, eta : Moving coordinate of a pair on a slider
//   minv     : Indicator for mechanical inversion
//   *s       : Displacement, velocity and acceleration of a slider on moving
//              coordinate system
//   *jp      : Displacement, velocity and acceleration of a slider on fixed
//              coordinate system
//   *jb      : Displacement, velocity and acceleration of a revolute pair connecting
//              two linksr
//   Function value: Complete/incomplete(Link chain cannot be formed)
//
//-----------------------------------------------------------

int PRR_links( POINT jr, double l, POINT pg, ANGLE psig, double gsi, double eta,
                int minv,   VARIABLE *s, POINT *jp, POINT *jb )
{
        double cp, sp;
        double a, da, dda, b, db, ddb, d, dd, ddd, q, dq, ddq, s0, ds0, dds0;

        cp = cos( psig.the );
        sp = sin( psig.the );

        a = -eta*sp + pg.P.x - jr.P.x;
        b =  eta*cp + pg.P.y - jr.P.y;

        d = l*l - ( a*sp - b*cp )*( a*sp - b*cp );
        if( d<0.0 )  return ERROR;

        da = -eta*cp*psig.dthe + pg.DP.x - jr.DP.x;
        dda = -eta*( cp*psig.ddthe - sp*psig.dthe*psig.dthe ) + pg.DDP.x - jr.DDP.x;

        db = -eta*sp*psig.dthe + pg.DP.y - jr.DP.y;
        ddb = -eta*( sp*psig.ddthe + cp*psig.dthe*psig.dthe ) + pg.DDP.y - jr.DDP.y;

        dd = -2.0*( da*sp + a*cp*psig.dthe - db*cp + b*sp*psig.dthe )*( a*sp - b*cp );
        ddd = -2.0*( ( dda*sp + 2.0*da*cp*psig.dthe + a*cp*psig.ddthe
                - a*sp*psig.dthe*psig.dthe
                - ddb*cp + 2.0*db*sp*psig.dthe + b*sp*psig.ddthe
                + b*cp*psig.dthe*psig.dthe )*( a*sp - b*cp )
                + ( da*sp + a*cp*psig.dthe - db*cp + b*sp*psig.dthe )
                  *( da*sp + a*cp*psig.dthe - db*cp + b*sp*psig.dthe ) );

        q = sqrt( d );
        dq = dd/( 2.0*q );
        ddq = ( 2.0*ddd*d - dd*dd )/( 4.0*d*q );
```

C language program of systematic kinematics analysis

```
        dds0 = -dda*cp + 2.0*da*sp*psig.dthe + a*sp*psig.ddthe
                + a*cp*psig.dthe*psig.dthe - ddb*sp - 2.0*db*cp*psig.dthe
                 - b*cp*psig.ddthe + b*sp*psig.dthe*psig.dthe;

        if( minv>0 ) {
                s->v = s0 + q;
                s->dv = ds0 + dq;
                s->ddv = dds0 + ddq;
        }
        else {
                s->v = s0 - q;
                s->dv = ds0 - dq;
                s->ddv = dds0 - ddq;
        }

        jp->P.x = s->v*cp + pg.P.x;
        jp->P.y = s->v*sp + pg.P.y;

        jp->DP.x = s->dv*cp - s->v*sp*psig.dthe + pg.DP.x;
        jp->DP.y = s->dv*sp + s->v*cp*psig.dthe + pg.DP.y;

        jp->DDP.x = s->ddv*cp - 2.0*s->dv*sp*psig.dthe - s->v*sp*psig.ddthe
                         - s->v*cp*psig.dthe*psig.dthe + pg.DDP.x;
        jp->DDP.y = s->ddv*sp + 2.0*s->dv*cp*psig.dthe + s->v*cp*psig.ddthe
                       - s->v*sp*psig.dthe*psig.dthe + pg.DDP.y;

        jb->P.x = gsi*cp - eta*sp + jp->P.x;
        jb->P.y = gsi*sp + eta*cp + jp->P.y;

        jb->DP.x = -gsi*sp*psig.dthe - eta*cp*psig.dthe + jp->DP.x;
        jb->DP.y =  gsi*cp*psig.dthe - eta*sp*psig.dthe + jp->DP.y;

        jb->DDP.x = -gsi*( sp*psig.ddthe + cp*psig.dthe*psig.dthe )
                         - eta*( cp*psig.ddthe - sp*psig.dthe*psig.dthe ) + jp->DDP.x;
        jb->DDP.y =  gsi*( cp*psig.ddthe - sp*psig.dthe*psig.dthe )
                         - eta*( sp*psig.ddthe + cp*psig.dthe*psig.dthe ) + jp->DDP.y;

        return SUCCESS;
}
```

```
//-----------------------------------------------------------
//
//    節の角変位（節上の2対偶間を結ぶ線分の成す角）
//
//   j1, j2   : 位置ベクトルとその微分
//   *angle   : 線分の成す角
//   関数値   : 0 : 正常終了
//              1 : 異常終了（2点が一致）
//
//-----------------------------------------------------------

int link_angle( POINT j1, POINT j2, ANGLE *angle )
{
        POINT diff;
        double q;

        get_diff( j1, j2, &diff );

        q = diff.P.x*diff.P.x + diff.P.y*diff.P.y;
        if( q <= 10e-30 ){
                printf("++ Error in link_angle（2対偶が一致してます）¥n");
                return ERROR;
        }

        angle->the   = atan2( diff.P.y, diff.P.x );
        angle->dthe  = ( diff.DP.y*diff.P.x - diff.P.y*diff.DP.x )/q;
        angle->ddthe = ( diff.DDP.y*diff.P.x - diff.P.y*diff.DDP.x )/q;

        return SUCCESS;
}
```

C language program of systematic kinematics analysis

```
dds0 = -dda*cp + 2.0*da*sp*psig.dthe + a*sp*psig.ddthe
           + a*cp*psig.dthe*psig.dthe - ddb*sp - 2.0*db*cp*psig.dthe
            - b*cp*psig.ddthe + b*sp*psig.dthe*psig.dthe;

if( minv>0 ) {
        s->v = s0 + q;
        s->dv = ds0 + dq;
        s->ddv = dds0 + ddq;
}
else {
        s->v = s0 - q;
        s->dv = ds0 - dq;
        s->ddv = dds0 - ddq;
}

jp->P.x = s->v*cp + pg.P.x;
jp->P.y = s->v*sp + pg.P.y;

jp->DP.x = s->dv*cp - s->v*sp*psig.dthe + pg.DP.x;
jp->DP.y = s->dv*sp + s->v*cp*psig.dthe + pg.DP.y;

jp->DDP.x = s->ddv*cp - 2.0*s->dv*sp*psig.dthe - s->v*sp*psig.ddthe
                  - s->v*cp*psig.dthe*psig.dthe + pg.DDP.x;
jp->DDP.y = s->ddv*sp + 2.0*s->dv*cp*psig.dthe + s->v*cp*psig.ddthe
                - s->v*sp*psig.dthe*psig.dthe + pg.DDP.y;

jb->P.x = gsi*cp - eta*sp + jp->P.x;
jb->P.y = gsi*sp + eta*cp + jp->P.y;

jb->DP.x = -gsi*sp*psig.dthe - eta*cp*psig.dthe + jp->DP.x;
jb->DP.y =  gsi*cp*psig.dthe - eta*sp*psig.dthe + jp->DP.y;
```

```
}
```

```
//---------------------------------------------------------------
//
//    Angular Motion of Link
//
//   j1, j2   : Displacement, velocity and acceleration of two points
//              on a link
//   *angle   : Angular displacement, velocity and acceleration of link
//   Function value: Complete/incomplete(Two points are quite same)
//
//---------------------------------------------------------------

int link_angle( POINT j1, POINT j2, ANGLE *angle )
{
        POINT diff;
        double q;

        get_diff( j1, j2, &diff );

        q = diff.P.x*diff.P.x + diff.P.y*diff.P.y;
        if( q <= 10e-30 ){
                printf("++ Error in link_angle (2対偶が一致してます)¥n");
                return ERROR;
        }

        angle->the   = atan2( diff.P.y, diff.P.x );
        angle->dthe  = ( diff.DP.y*diff.P.x - diff.P.y*diff.DP.x )/q;
        angle->ddthe = ( diff.DDP.y*diff.P.x - diff.P.y*diff.DDP.x )/q;

        return SUCCESS;
}
```

You can download these programs from WEB-site:

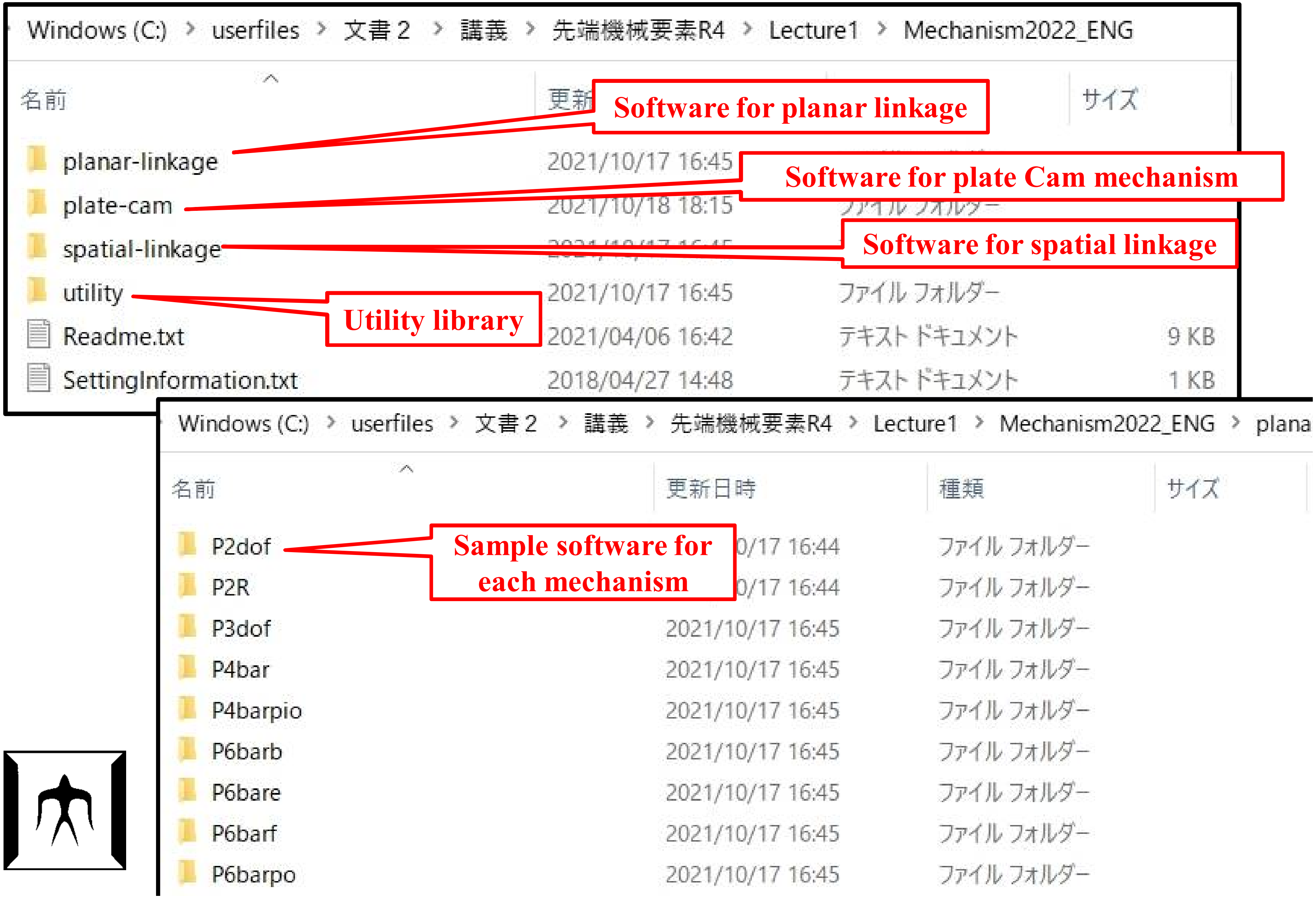
http://www.rmsv.mech.e.titech.ac.jp/japanese/download.html

or from T2Box which URL is shown in T2SCHOLA

C language program of systematic kinematics analysis

How to use offered programs:

Folders



Windows (C:) > userfiles > 文書 2 > 講義 > 先端機械要素R4 > Lecture1 > Mechanism2022_ENG > planar-linkage > P6barpo

Subprogram library → P6barpo_subfunctions

Program for each analysis → Pmp6barpo

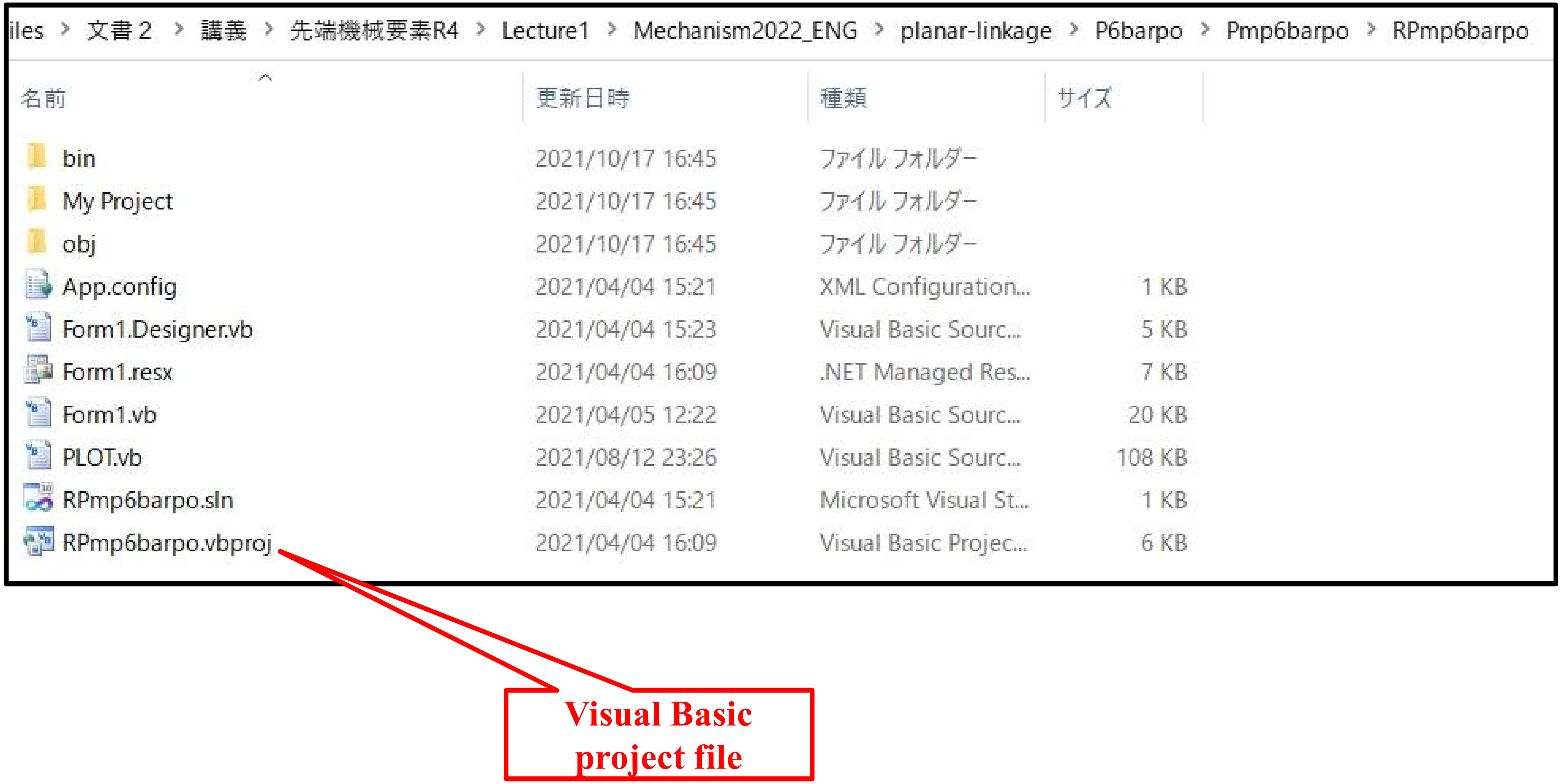
名前	更新日時	種類	サイズ
P6barpo_subfunctions	2021/10/17 16:45	ファイル フォルダー	
Pidp6barpo	2021/10/17 16:45	ファイル フォルダー	
Pmp6barpo		ファイル フォルダー	
IGDATA.dat		DAT ファイル	1 KB
IMDATA.dat	2011/10/14 16:52	DAT ファイル	1 KB

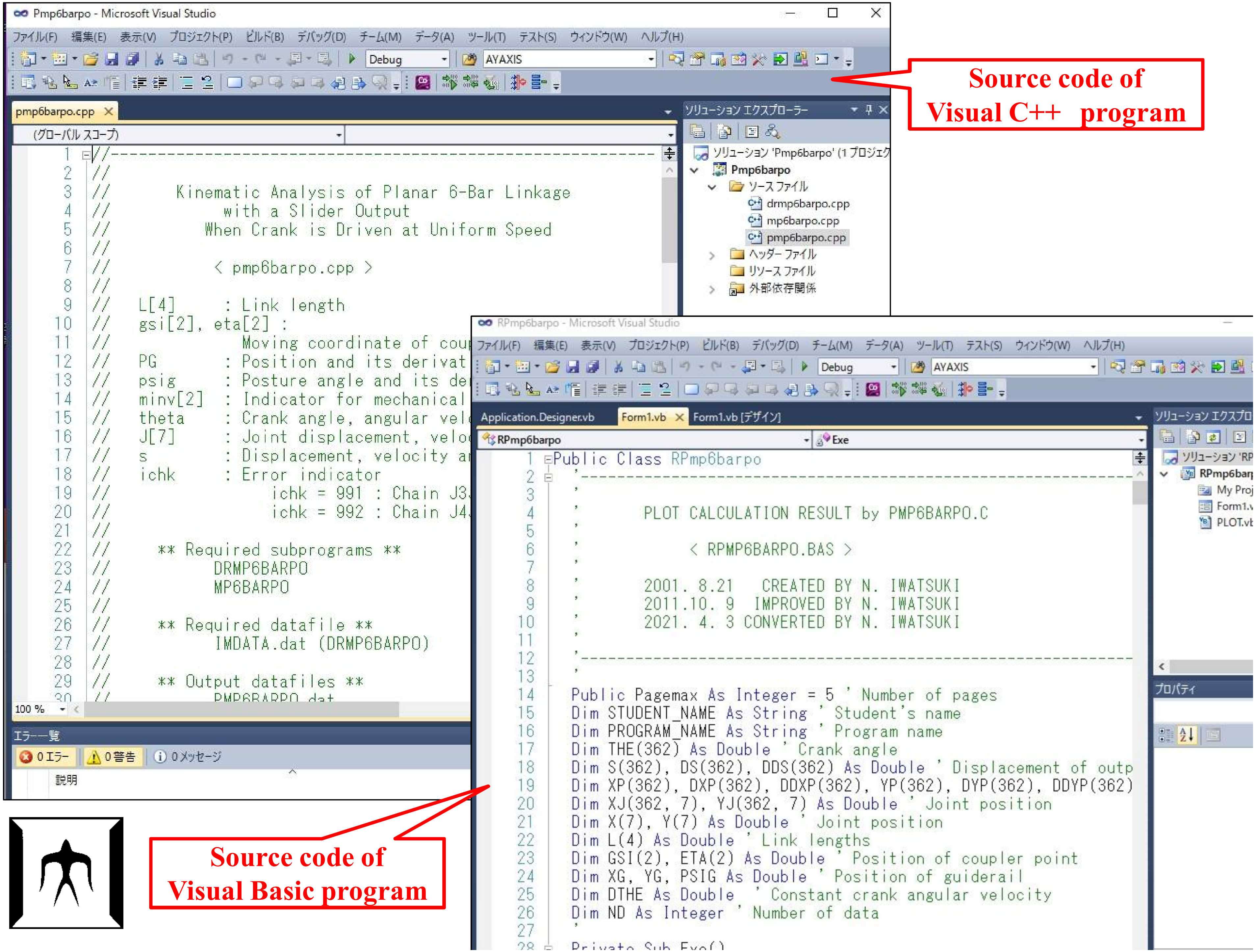
lows (C:) > userfiles > 文書 2 > 講義 > 先端機械要素R4 > Lecture1 > Mechanism2022_ENG > planar-linkage > P6bar

Program folder for graphics → RPmp6barpo

Visual C++ project file → Pmp6barpo.vcxproj

名前	更新日時	種類	サイズ
Debug	21/10/17 16:45	ファイル フォルダー	
ipch	21/10/17 16:45	ファイル フォルダー	
RPmp6barpo	2021/10/17 16:45	ファイル フォルダー	
MSSCCPRJ.SCC	2012/05/27 13:51	Microsoft SourceS...	1 KB
pmp6barpo.cpp	2018/04/22 9:22	C++ Source	5 KB
PMP6BARPO.CSV	2021/04/15 12:18	Microsoft Excel CS...	51 KB
PMP6BARPO.DAT	2021/04/15 12:18	DAT ファイル	48 KB
Pmp6barpo.sdf	:26	SQL Server Compa...	1,876 KB
Pmp6barpo.sln	:47	Microsoft Visual St...	1 KB
PMP6BARPO.TRJ	2021/04/15 12:18	TRJ ファイル	73 KB
Pmp6barpo.vcxproj	2018/04/25 20:09	VC++ Project	4 KB
Pmp6barpo.vcxproj.filters	2012/05/27 13:50	VC++ Project Filte...	2 KB
Pmp6barpo.vcxproj.user	2012/05/27 13:47	Visual Studio Proje...	1 KB
pmp6barpo_subfunctions.h	2011/10/14 16:36	C/C++ Header	1 KB


iles > 文書 2 > 講義 > 先端機械要素R4 > Lecture1 > Mechanism2022_ENG > planar-linkage > P6barpo > Pmp6barpo > RPmp6barpo
名前 更新日時 種類 サイズ
bin 2021/10/17 16:45 ファイル フォルダー
My Project 2021/10/17 16:45 ファイル フォルダー
obj 2021/10/17 16:45 ファイル フォルダー
App.config 2021/04/04 15:21 XML Configuration... 1 KB
Form1.Designer.vb 2021/04/04 15:23 Visual Basic Sourc... 5 KB
Form1.resx 2021/04/04 16:09 .NET Managed Res... 7 KB
Form1.vb 2021/04/05 12:22 Visual Basic Sourc... 20 KB
PLOT.vb 2021/08/12 23:26 Visual Basic Sourc... 108 KB
RPmp6barpo.sln 2021/04/04 15:21 Microsoft Visual St... 1 KB
RPmp6barpo.vbproj 2021/04/04 16:09 Visual Basic Projec... 6 KB
Visual Basic project file


Source code of Visual C++ program
Pmp6barpo - Microsoft Visual Studio
pmp6barpo.cpp
//---
//
// Kinematic Analysis of Planar 6-Bar Linkage
// with a Slider Output
// When Crank is Driven at Uniform Speed
//
// < pmp6barpo.cpp >
//
// L[4] : Link length
// gsi[2], eta[2] :
// Moving coordinate of cou
// PG : Position and its derivat
// psig : Posture angle and its de
// minv[2] : Indicator for mechanica
// theta : Crank angle, angular vel
// J[7] : Joint displacement, velo
// s : Displacement, velocity a
// ichk : Error indicator
// ichk = 991 : Chain J3
// ichk = 992 : Chain J4
//
// ** Required subprograms **
// DRMP6BARPO
// MP6BARPO
//
// ** Required datafile **
// IMDATA.dat (DRMP6BARPO)
//
// ** Output datafiles **
drmp6barpo.cpp
mp6barpo.cpp
pmp6barpo.cpp
RPmp6barpo - Microsoft Visual Studio
Application.Designer.vb
Form1.vb
Form1.vb [デザイン]
Public Class RPmp6barpo
'---
'
'
' PLOT CALCULATION RESULT by PMP6BARPO.C
'
' < RPMP6BARPO.BAS >
'
' 2001. 8.21 CREATED BY N. IWATSUKI
' 2011.10. 9 IMPROVED BY N. IWATSUKI
' 2021. 4. 3 CONVERTED BY N. IWATSUKI
'
'---
'
Public Pagemax As Integer = 5 ' Number of pages
Dim STUDENT_NAME As String ' Student's name
Dim PROGRAM_NAME As String ' Program name
Dim THE(362) As Double ' Crank angle
Dim S(362), DS(362), DDS(362) As Double ' Displacement of outp
Dim XP(362), DXP(362), DDXP(362), YP(362), DYP(362), DDYP(362)
Dim XJ(362, 7), YJ(362, 7) As Double ' Joint position
Dim X(7), Y(7) As Double ' Joint position
Dim L(4) As Double ' Link lengths
Dim GSI(2), ETA(2) As Double ' Position of coupler point
Dim XG, YG, PSIG As Double ' Position of guiderail
Dim DTHE As Double ' Constant crank angular velocity
Dim ND As Integer ' Number of data
Source code of Visual Basic program

Readme.txt file

Readme.txt - メモ帳

ファイル(F) 編集(E) 書式(O) 表示(V) ヘルプ(H)

These files are sample programs for kinematics and dynamics
analyses of link mechanisms, which will be introduced in the
lecture, 'Advanced Mechanical Elements' for graduate students
of Tokyo Institute of Technology, Japan.It's my pleasure that
students will try to analyze and synthesize link mechanisms
by utilizing these sample programs.Please note the followings.

1.Language
All programs are coded as Visual C++ language in the environment
of Microsoft Visial Studio 2010. Because all source files are opened
anyone can export them to another C language environment.

****

Everyone can freely download and use the Microsoft Visual
Studio 2019 Environment in the followinf WEB-site:

https://visualstudio.microsoft.com/ja/vs/

****

2. Copyright
The author keep copyright of all programs.Anyone are strictly
prohibited to resell them for counter value.

3. How to use
After downloading and def
linkage_ENG_new.zip,many fi
project file ***.vcxproj, he
his/her PC has Microsoft Vis
results are outputed as a te

4. Confirmation of programs
All programs are confirmed
environment with Microsoft W
mistakes occur through these
for them.

5. How to use each program
There are many comment se
read and understand them.

6. Graphics programs to plot
In order to visualize the
a data plotting program for
program is coded as Microsof
executing R**

Readme.txt - メモ帳

ファイル(F) 編集(E) 書式(O) 表示(V) ヘルプ(H)

7. Version up
In order to correct bugs or to add new functions,these programs will be
updated without a preliminary announcement.

8.Contents of programs

(1)Subfolder 'utility'
General and systematic subprogram to calculate kinematics and dynamics of
planar and spatial link mechanisms, subprograms for numerical calculation,
subprograms for console or file interface, and ther library files and header
files are installed.

(2)Subfolder 'planar-linkage'
Sample program to analyze and synthesize planar link mechanisms

(2-1)Subfolder 'P2R'
Sample program to analyze planar serial 2R manipualtor
(2-1-a)Subfolder 'Pfkp2r'
Forward kinematics analysis of planar serial 2R manipualtor
(2-1-
2R manipualtor
R manipualtor

mechanism
mechanism
ar link mechanism
nk mechanism

mechanism type-B
mechanism type-B
nk mechanism type-B

mechanism type-E
mechanism type-E
nk mechanism type-E

SettingInformation.txt file

SettingInformation.txt - メモ帳

ファイル(F) 編集(E) 書式(O) 表示(V) ヘルプ(H)

2018. 4.19

Environment Setting Information :

Project TAB

Property TAB

C/C++
General
Additional Include Directory: ..¥..¥..¥utility

Linker
Genral
Additional Librarye Directory: ..¥..¥..¥utility
Input
Additional Library Files : utility.lib

4.2 Procedure of the systematic kinematics analysis

(1)Give position of fixed pairs
Give position of fixed revolute pairs and position and posture of fixed prismatic pairs.

(2)Calculate motion of input links
By using crank_input or slider_input, motion of input link can be calculated.

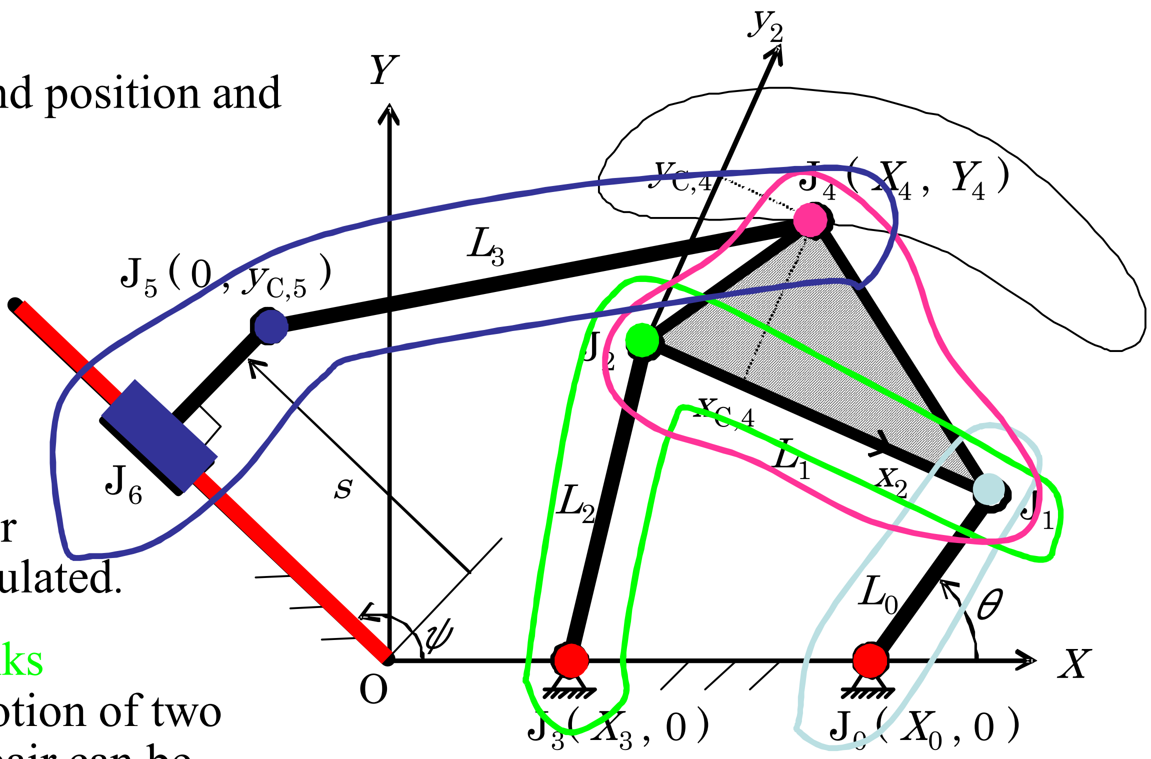
(3)Calculate motion of moving pair on input links.
By using coupler_point, motion of other moving pairs on input links can be calculated.

(4)Calculate motion of two adjacent links
By using RRR_links or PRR_links, motion of two adjacent links with revolute/prismatic pair can be calculated.

(5)Calculate motion of pairs on coupler links
By using coupler_point, motion of moving pairs on coupler link can be calculated.

(6)Repeat (4) and (5) and then calculate all moving pairs

END

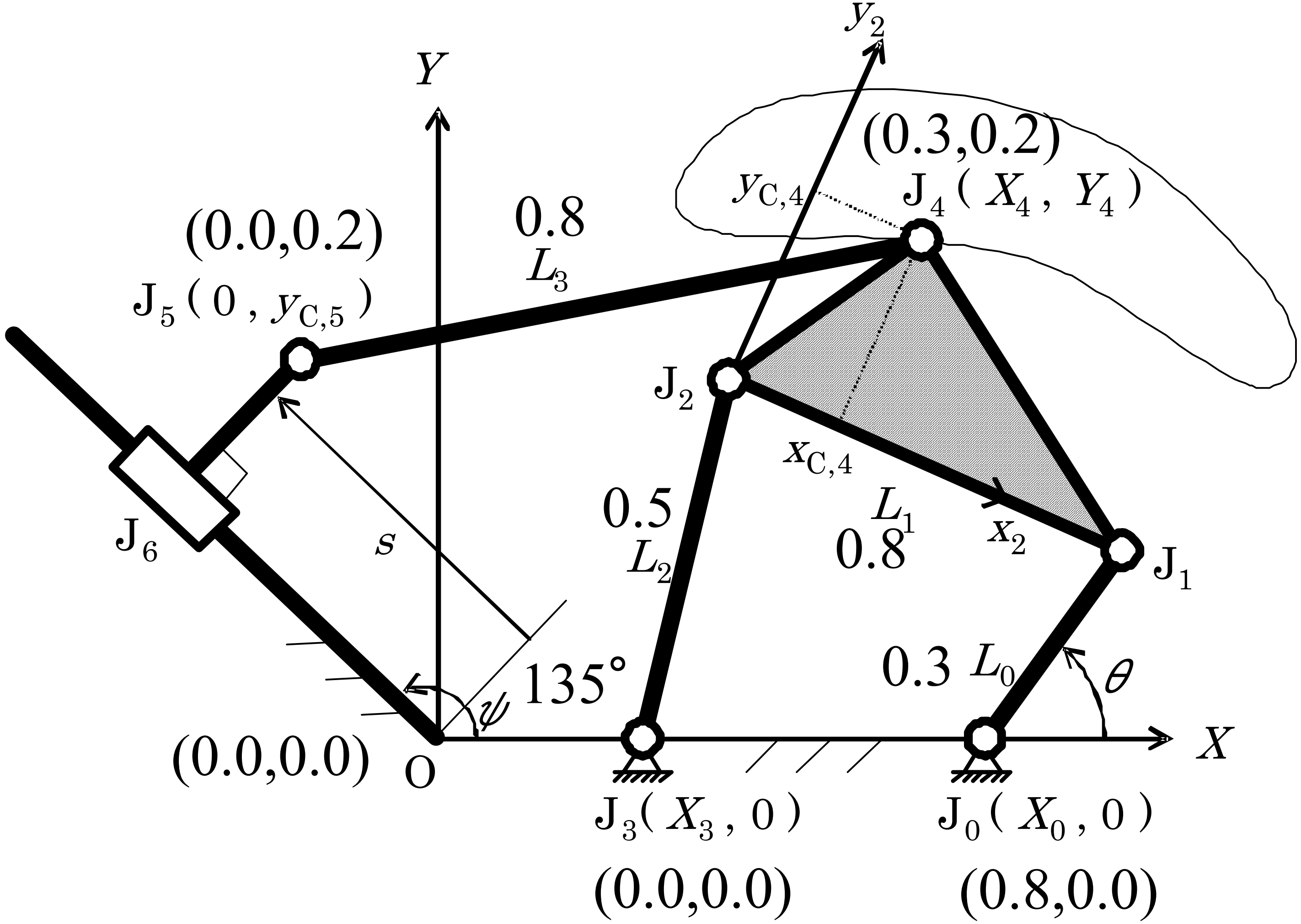


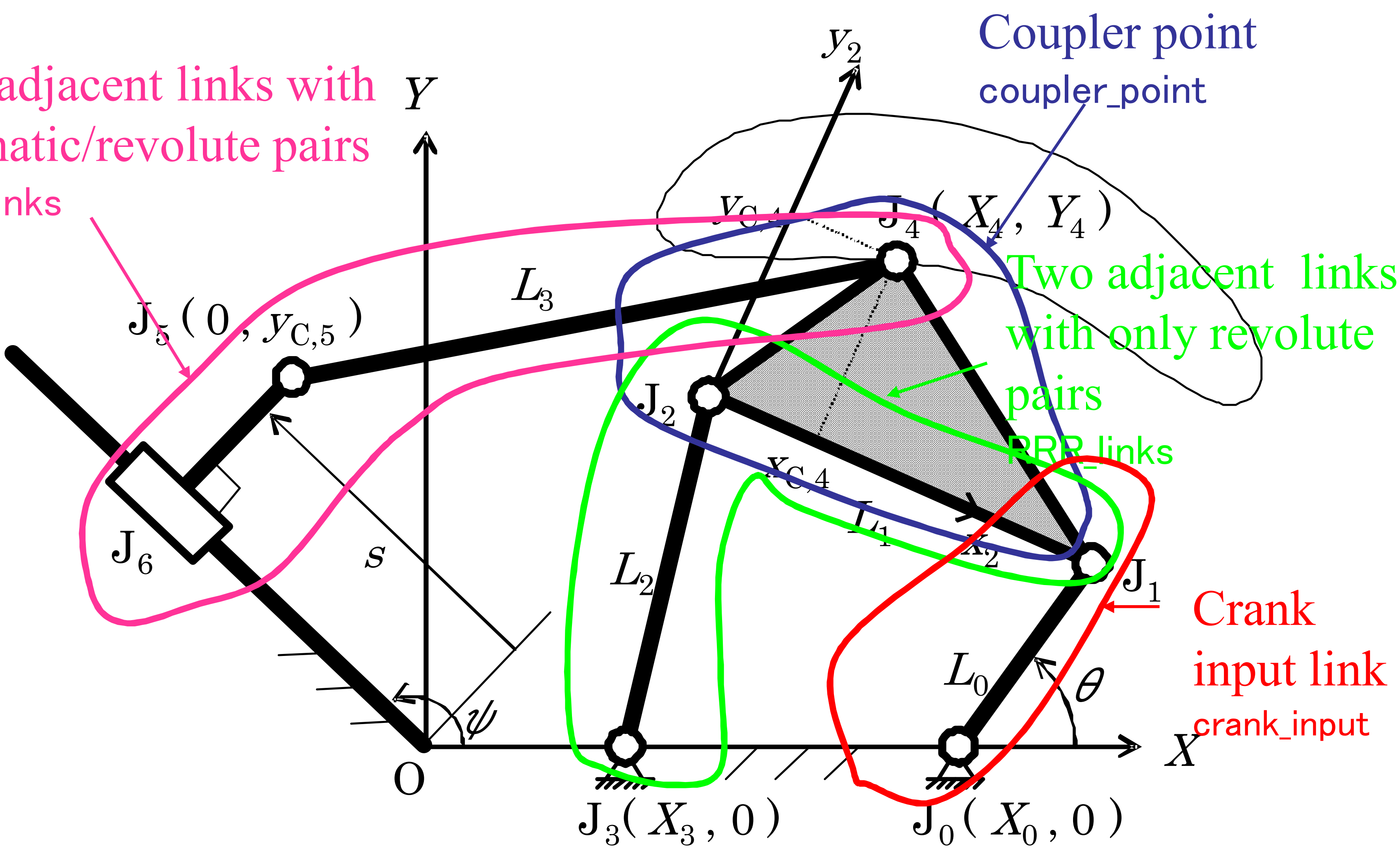
Ex. Crank input-Slider output planar 6-bar link mechanism

"It is important to find two adjacent links to be analyzed."

Examples of analysis：

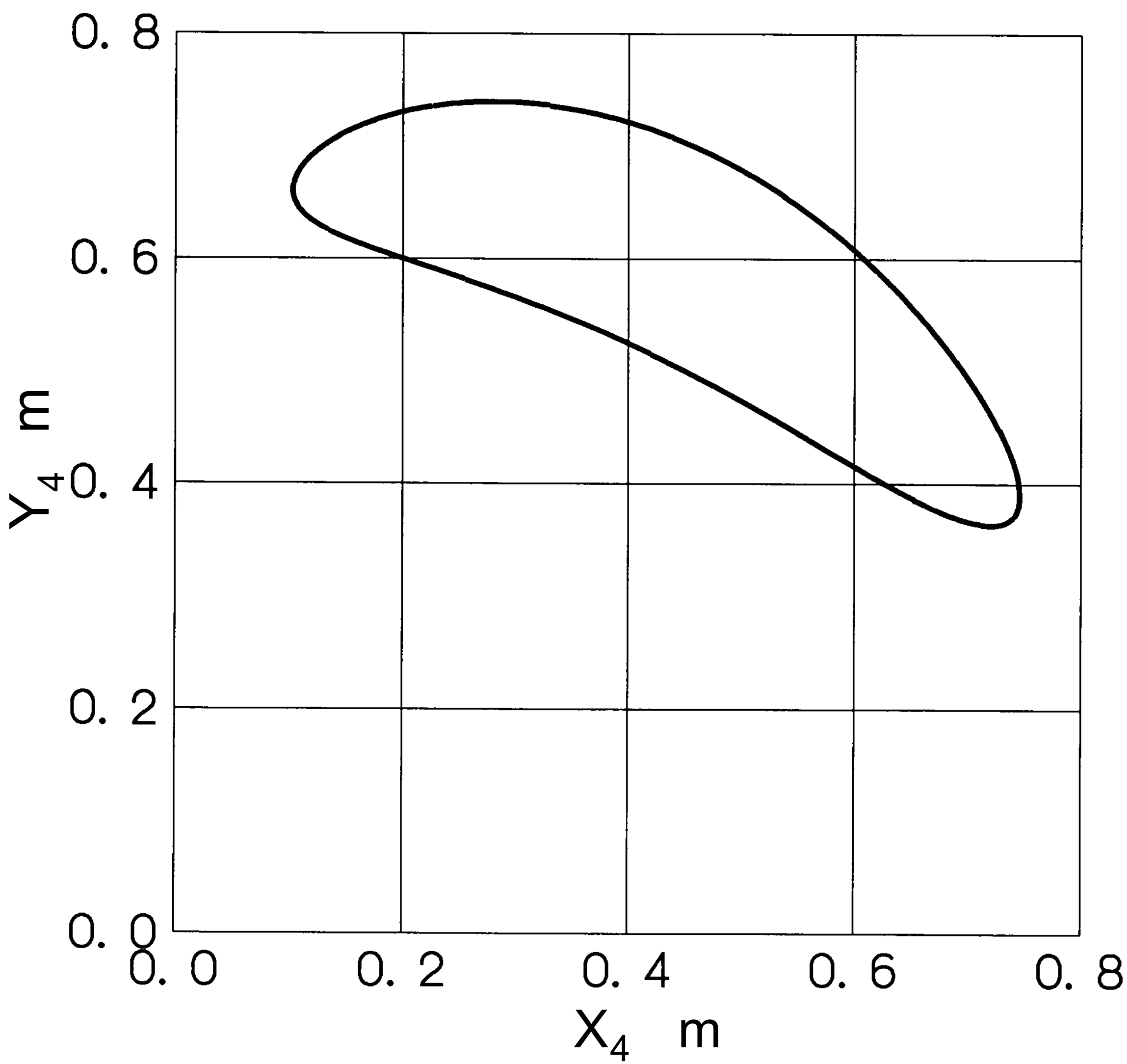
(1)Crank input-Slider output planar 6-bar link mechanism



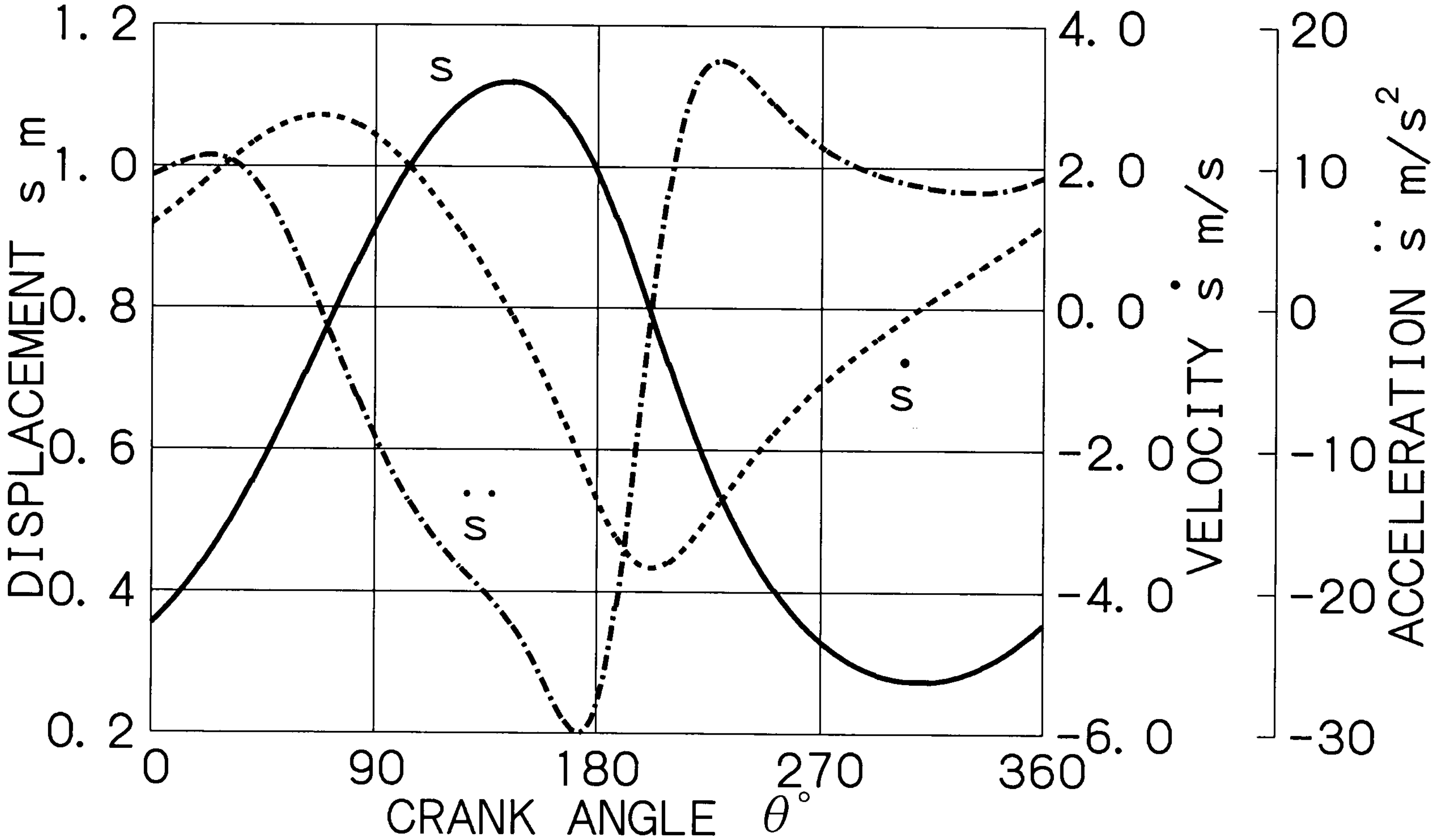


1)Crank input-Slider output planar 6-bar link mechanism

Procedure …

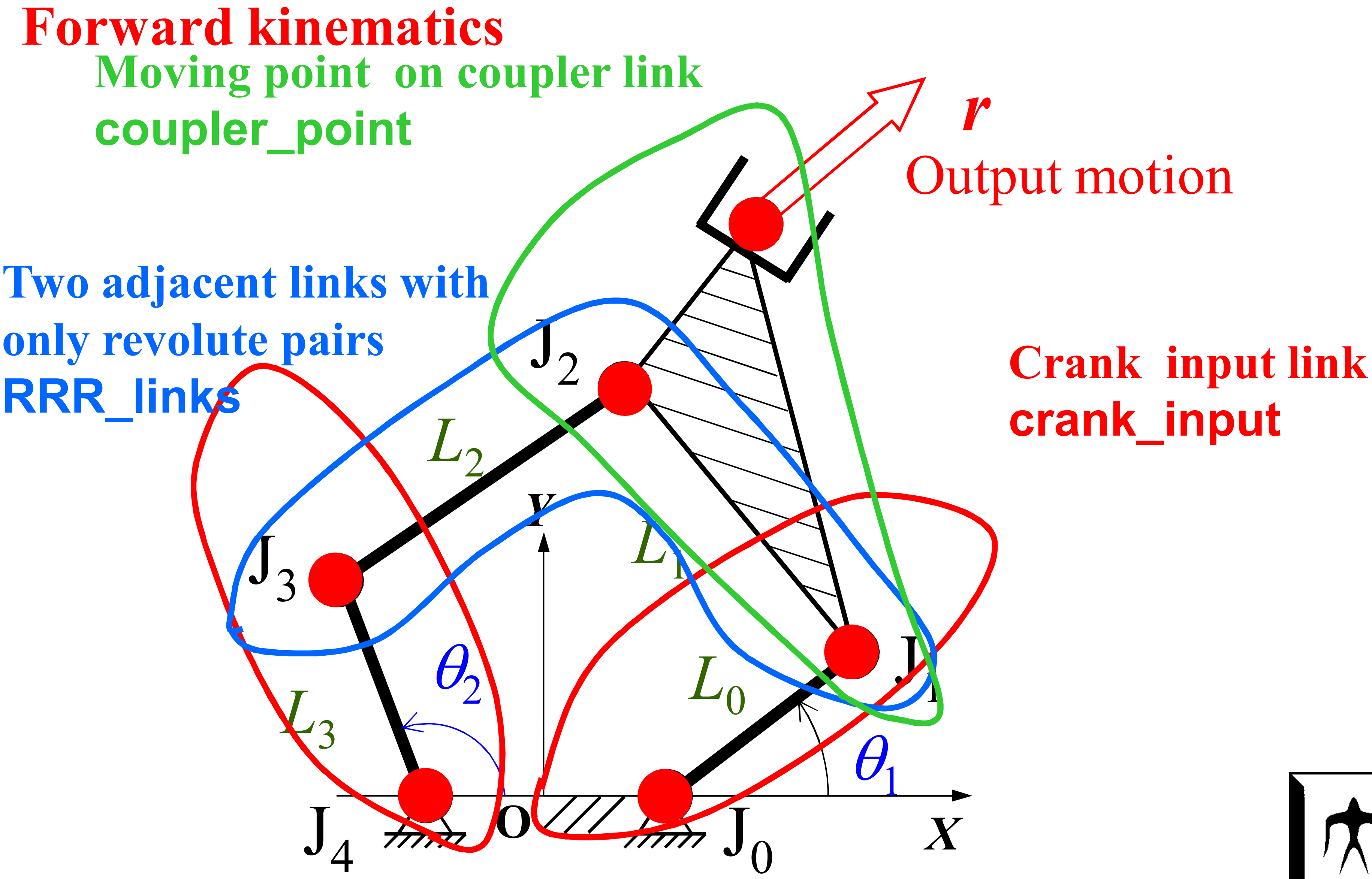


Locus of J4 of the 6-bar mechanism

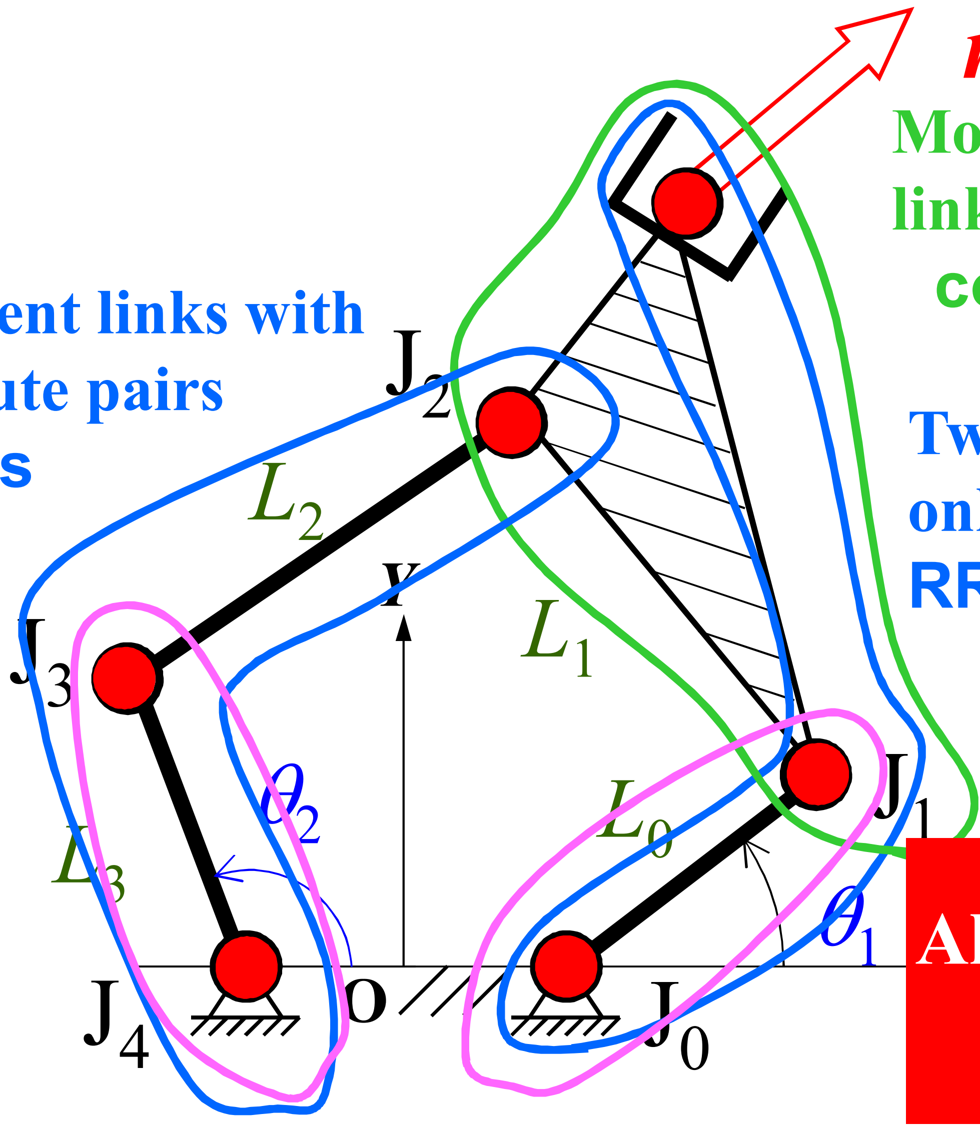


Displacement, velocity and acceleration of output slider of the 6-bar mechanism

(2)A planar closed-loop manipulator with 2 DOF



Inverse kinematics



Moving point on coupler link
coupler_point

Two adjacent links with only revolute pairs
RRR_links

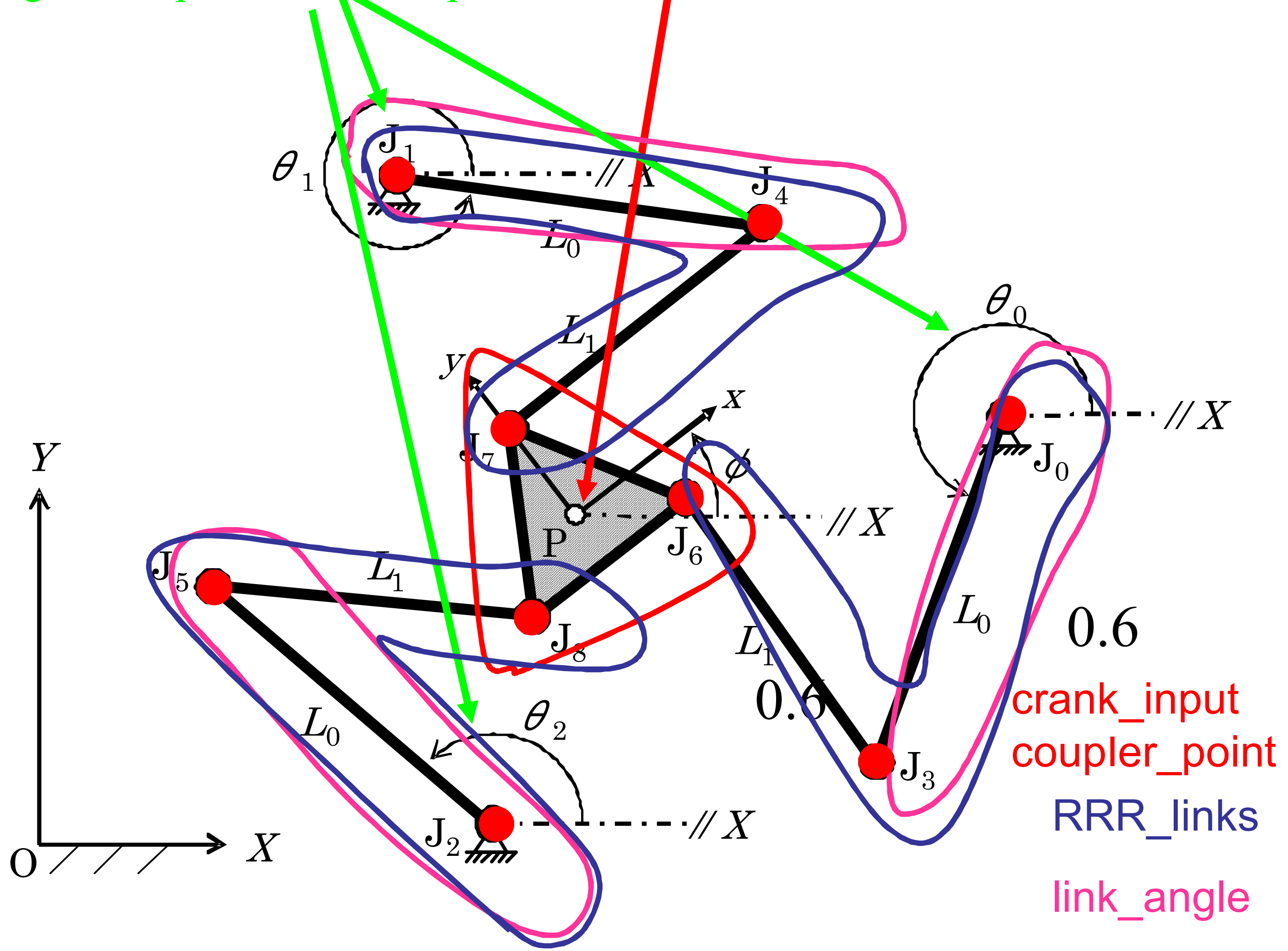
Two adjacent links with only revolute pairs
RRR_links

Input angle
link_angle

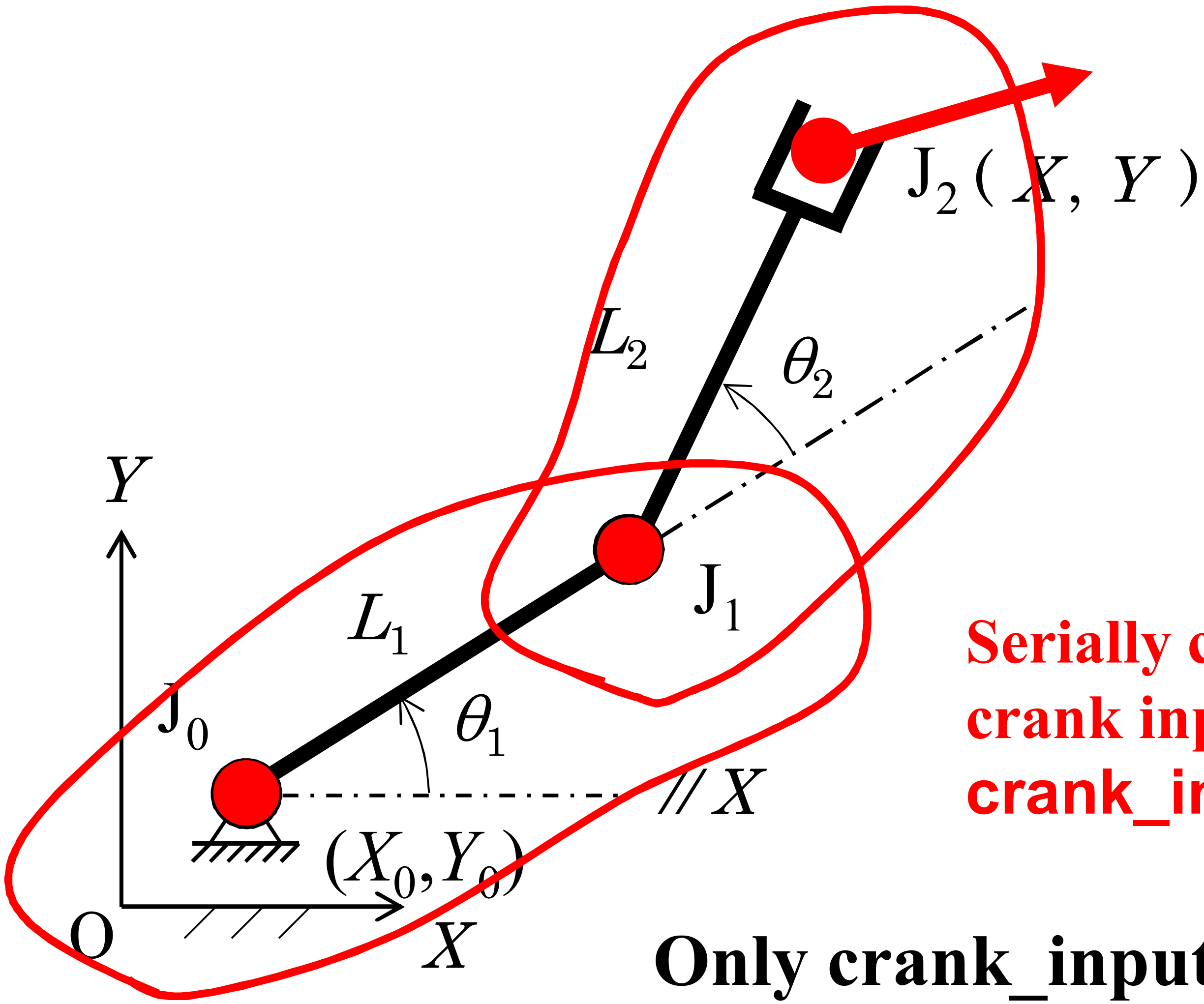
All motion can easily be calculated.

(3)Inverse kinematics of a planar closed-loop manipulator with 3 DOF

By specifying the position and motion of output platform,
Angular displacement of input link can be calculated.

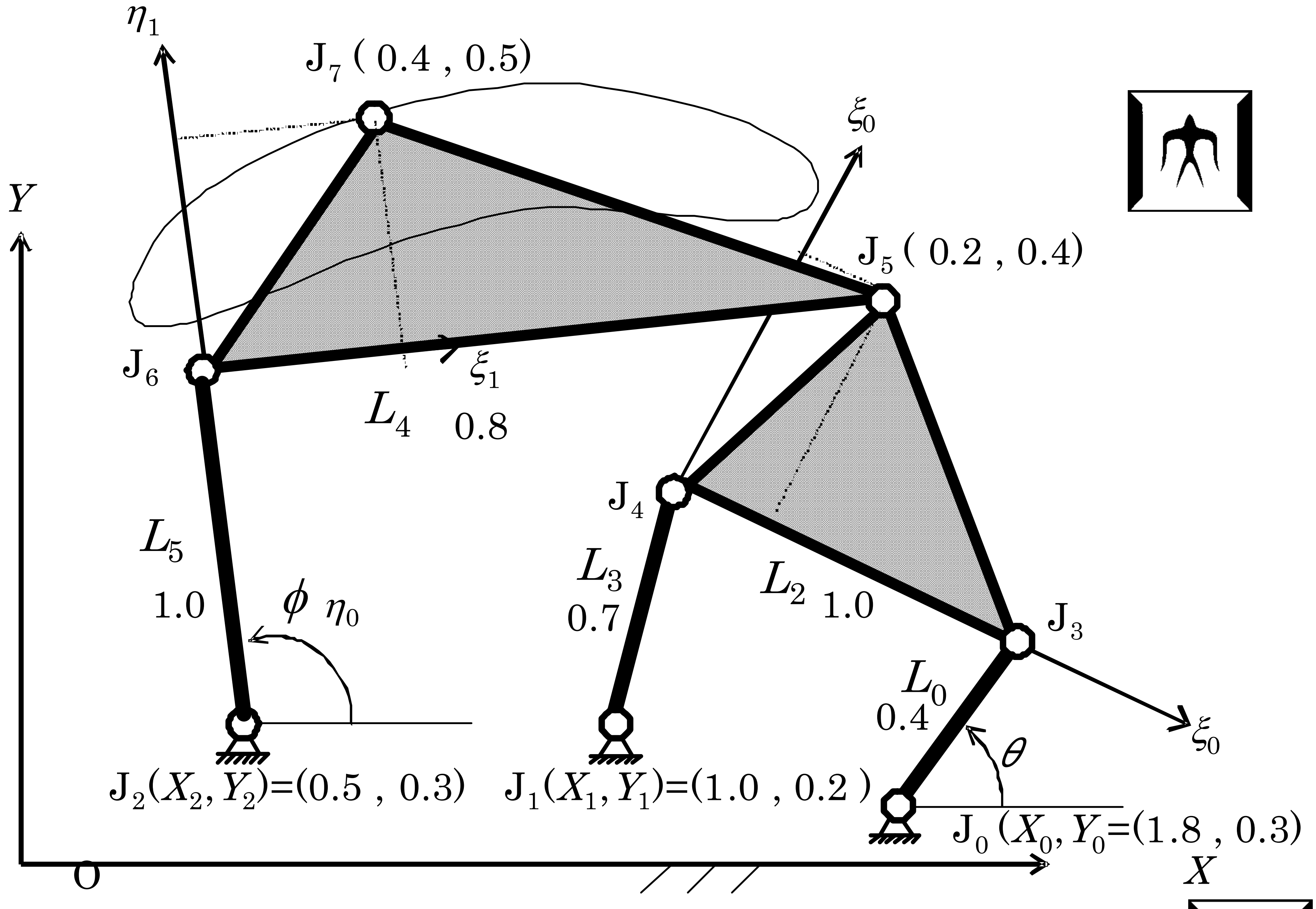


(4)Planar serial 2R manipulator

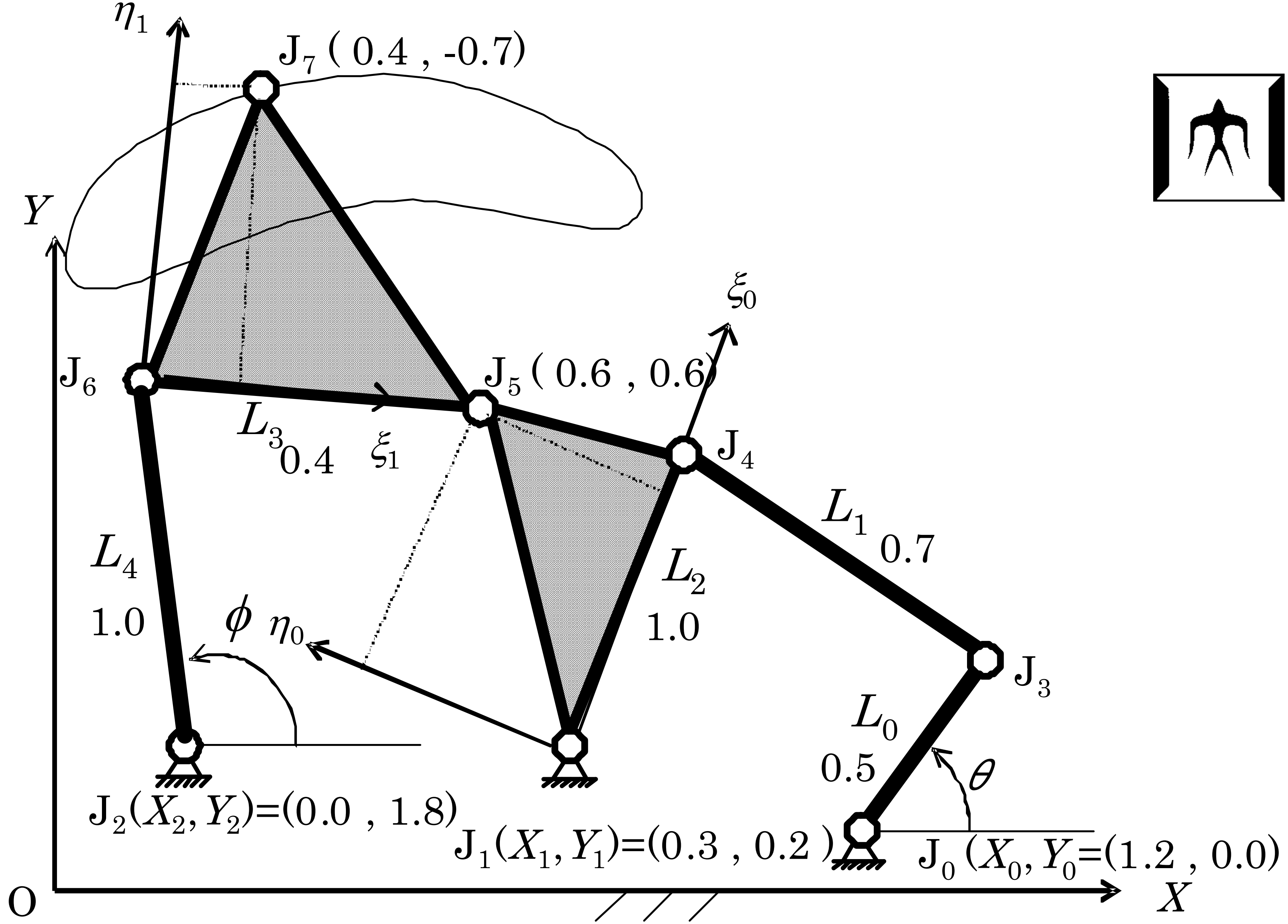


Serially connected two crank input link
crank_input

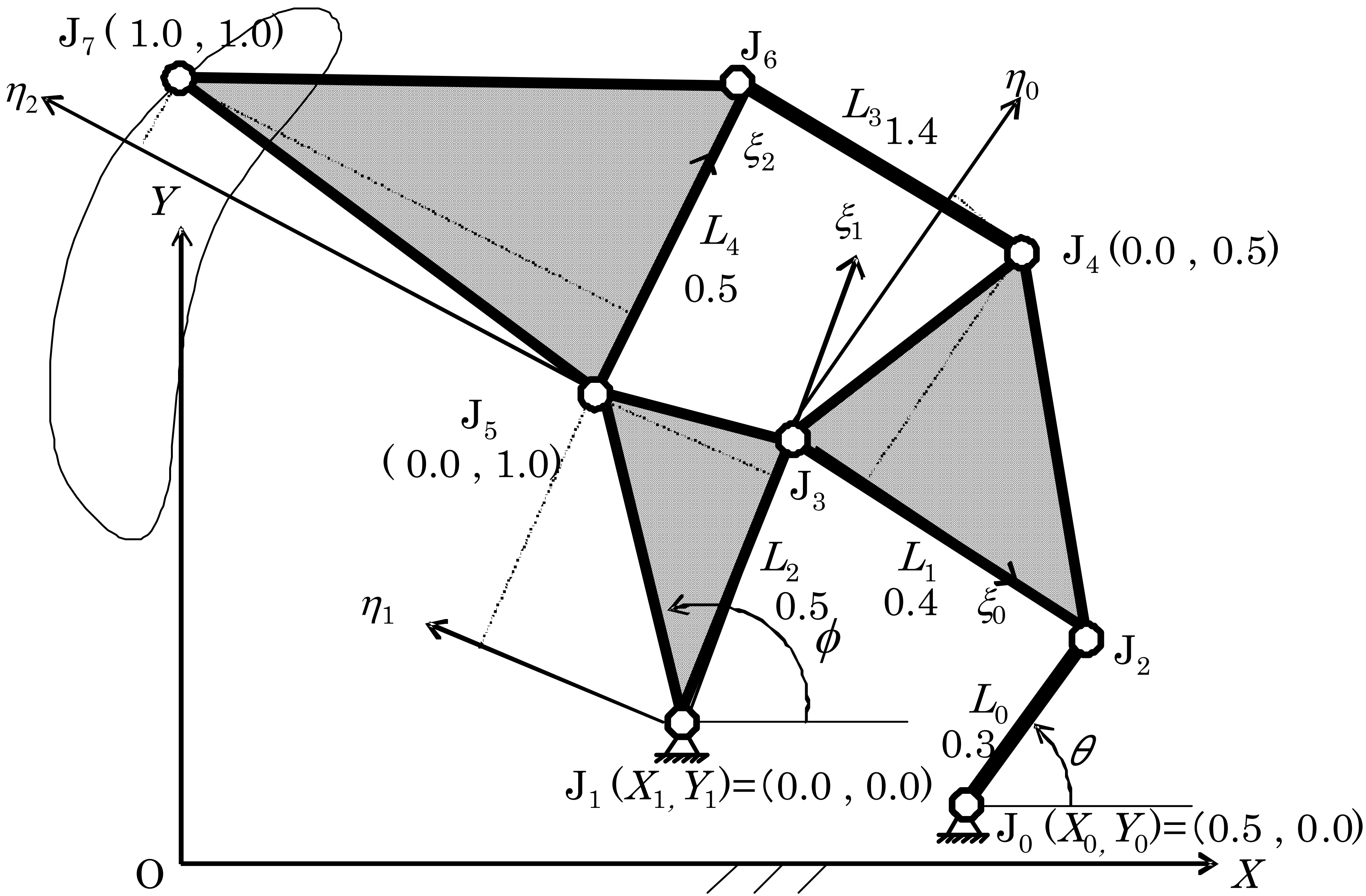
Only crank_input



(4)Planar 6-bar link mechanism type-E

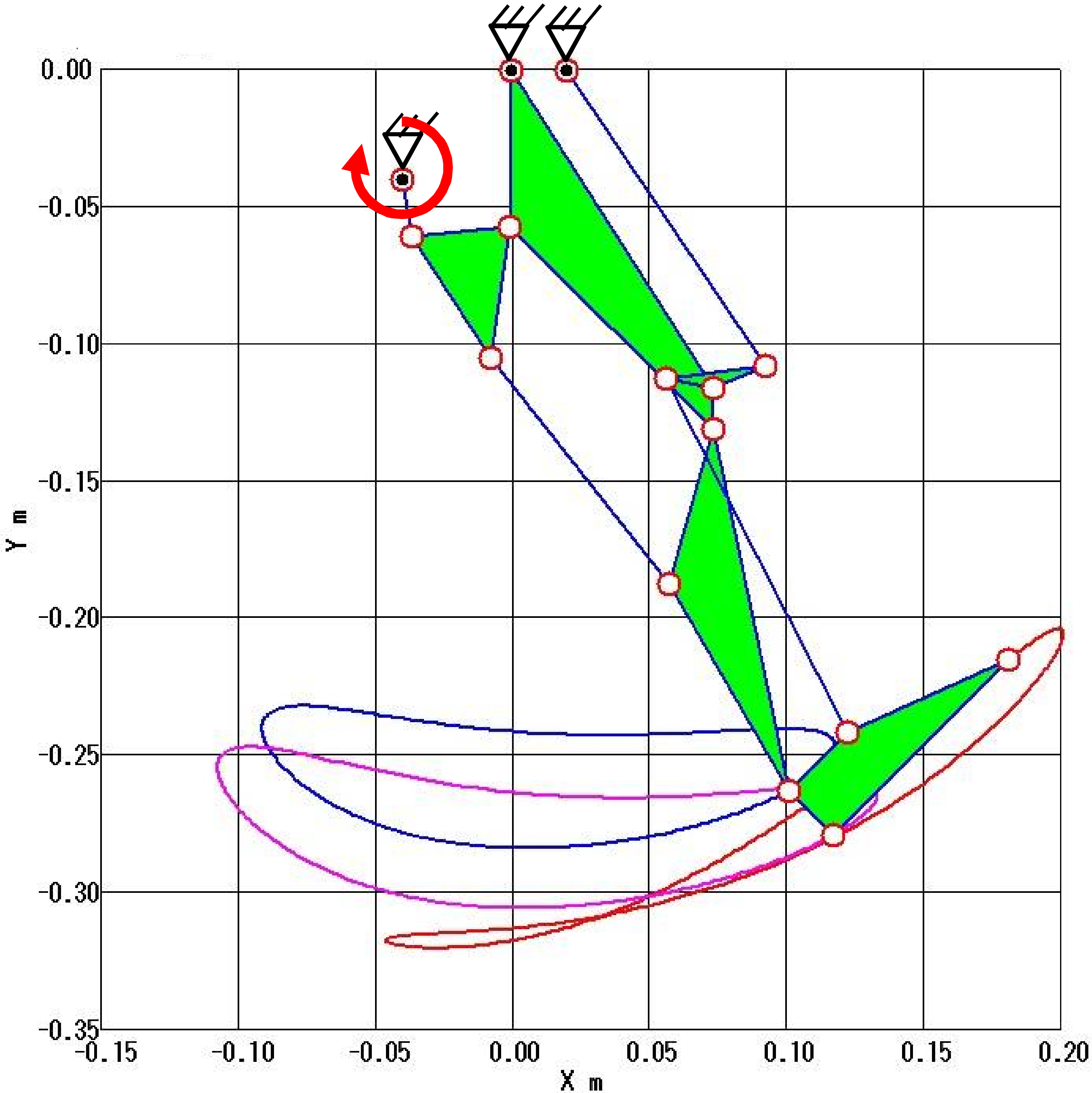


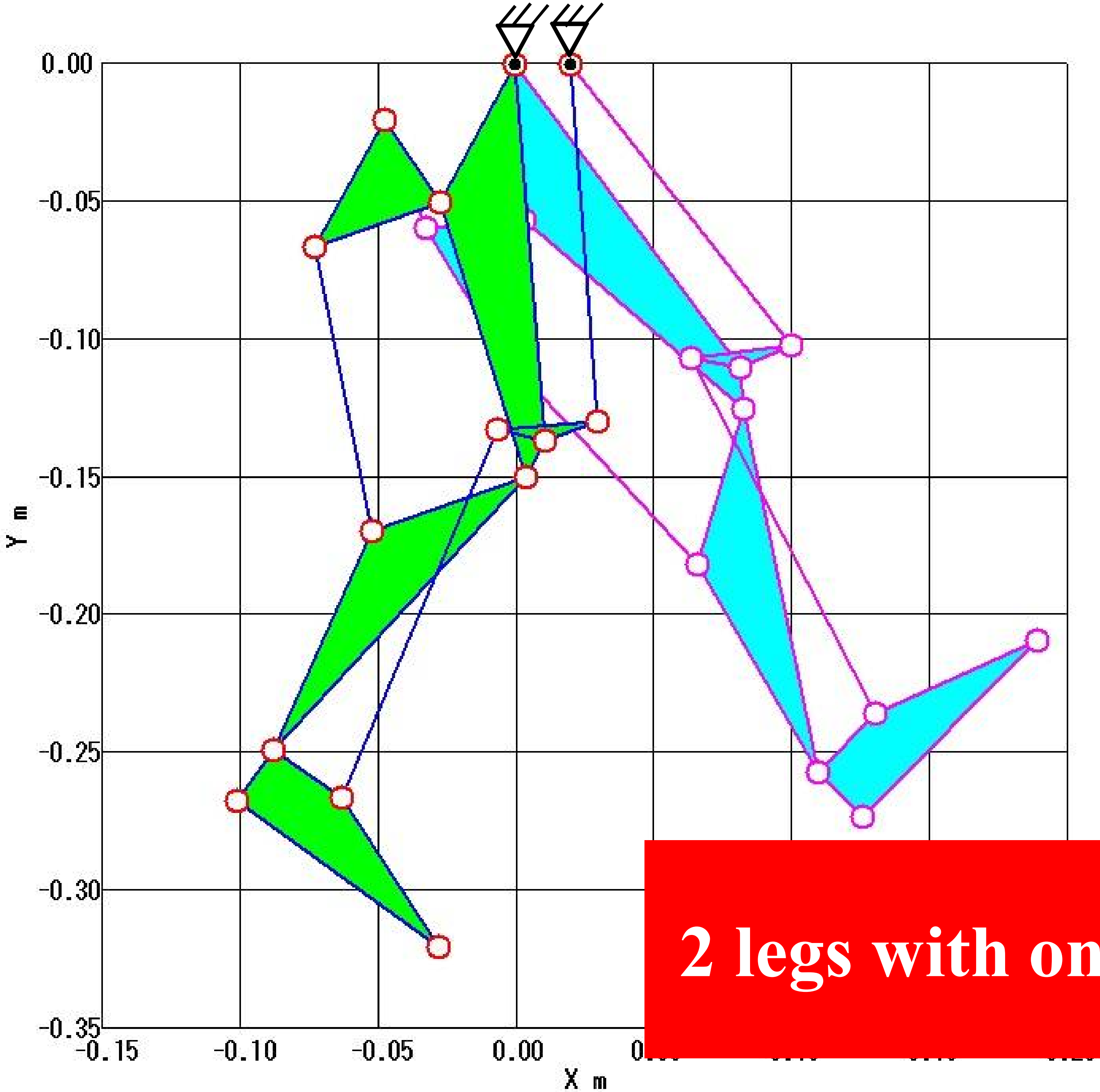
(5)Planar 6-bar link mechanism type-B



(6)Planar 6-bar link mechanism type-F

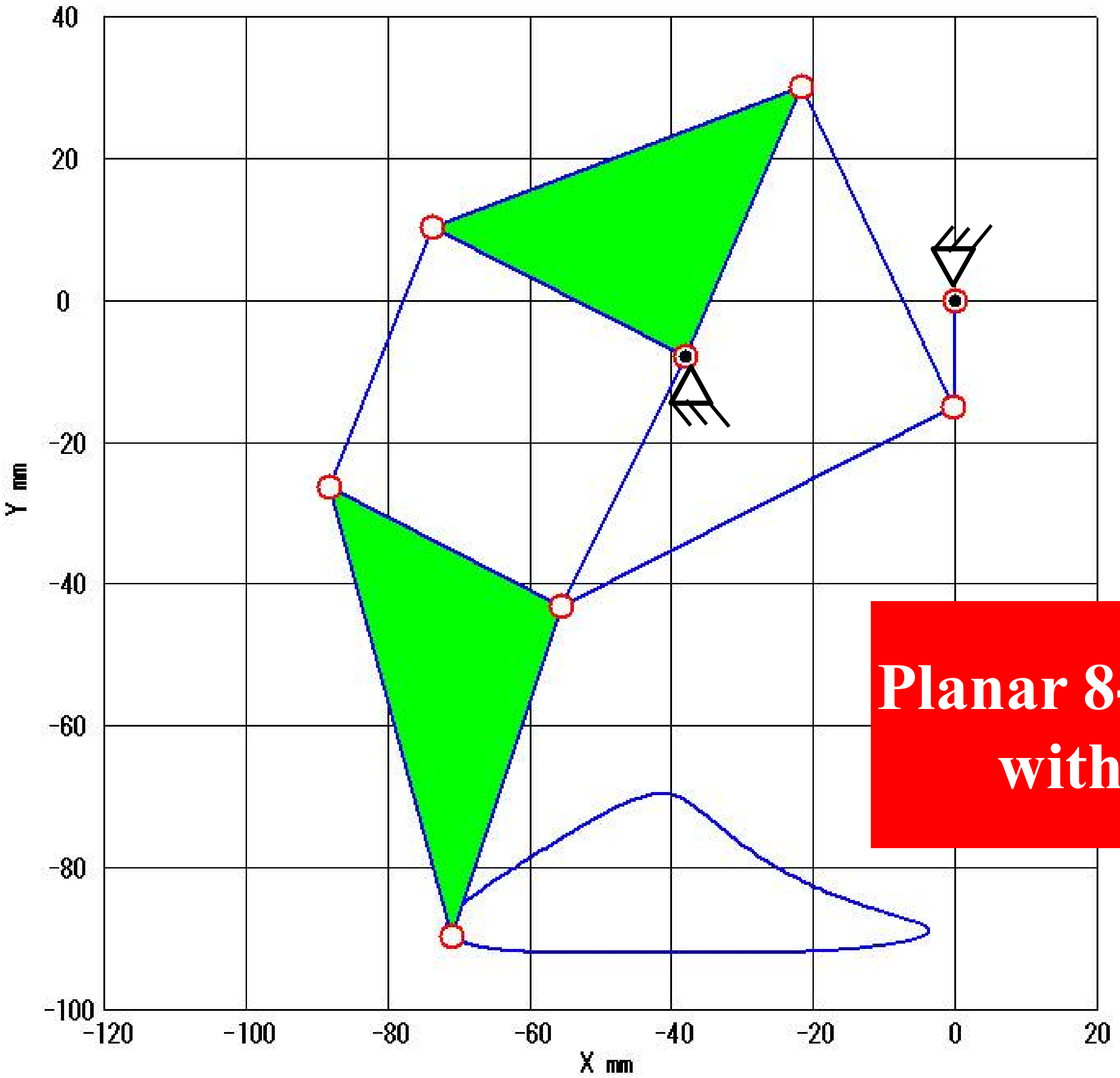
(7)Planar closed-loop leg-mechanism with 1 DOF



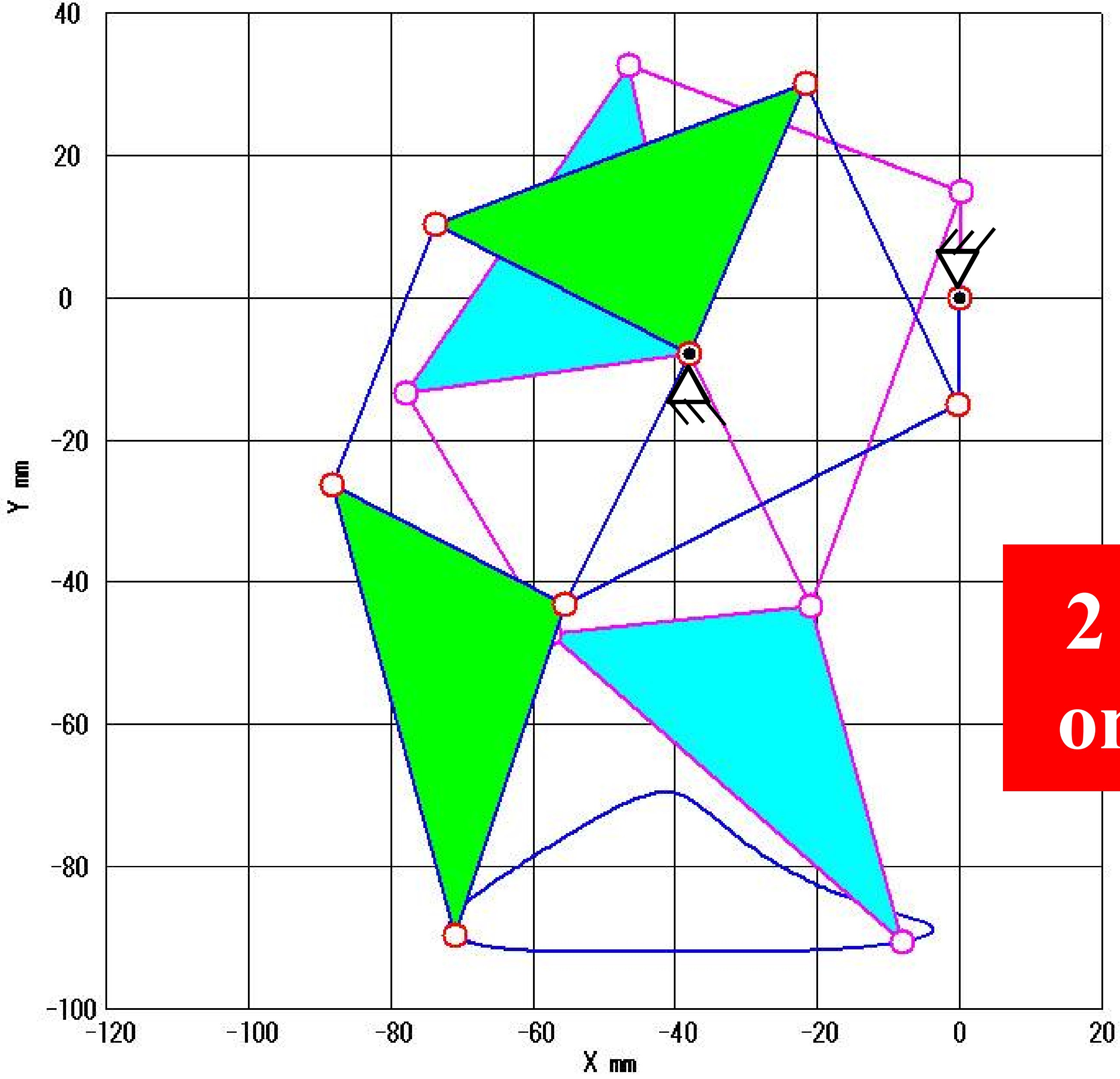


2 legs with only 1 DOF

(8)Planar closed-loop leg-mechanism with 1 DOF
"Theo Jansen's mechanism"

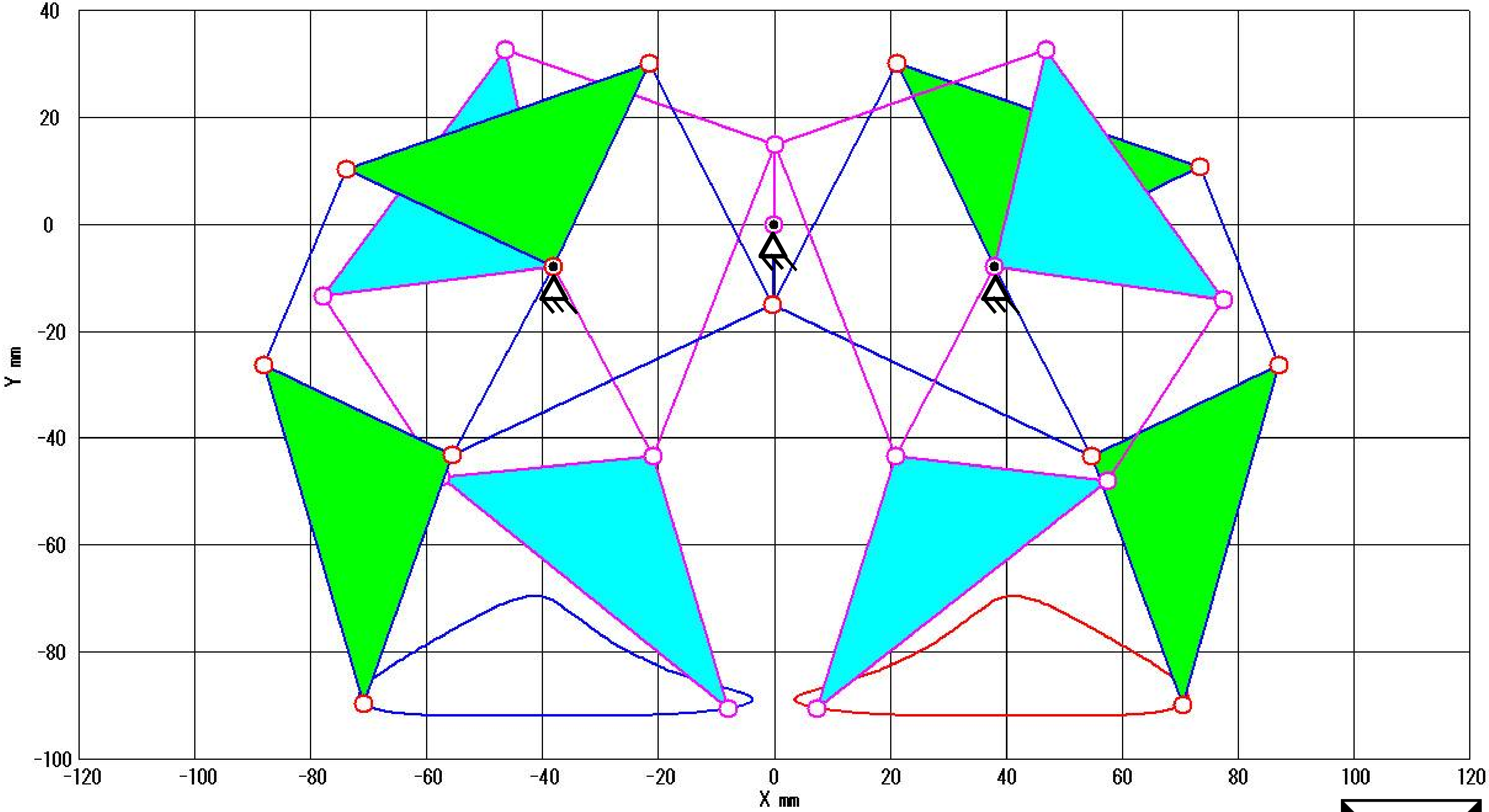


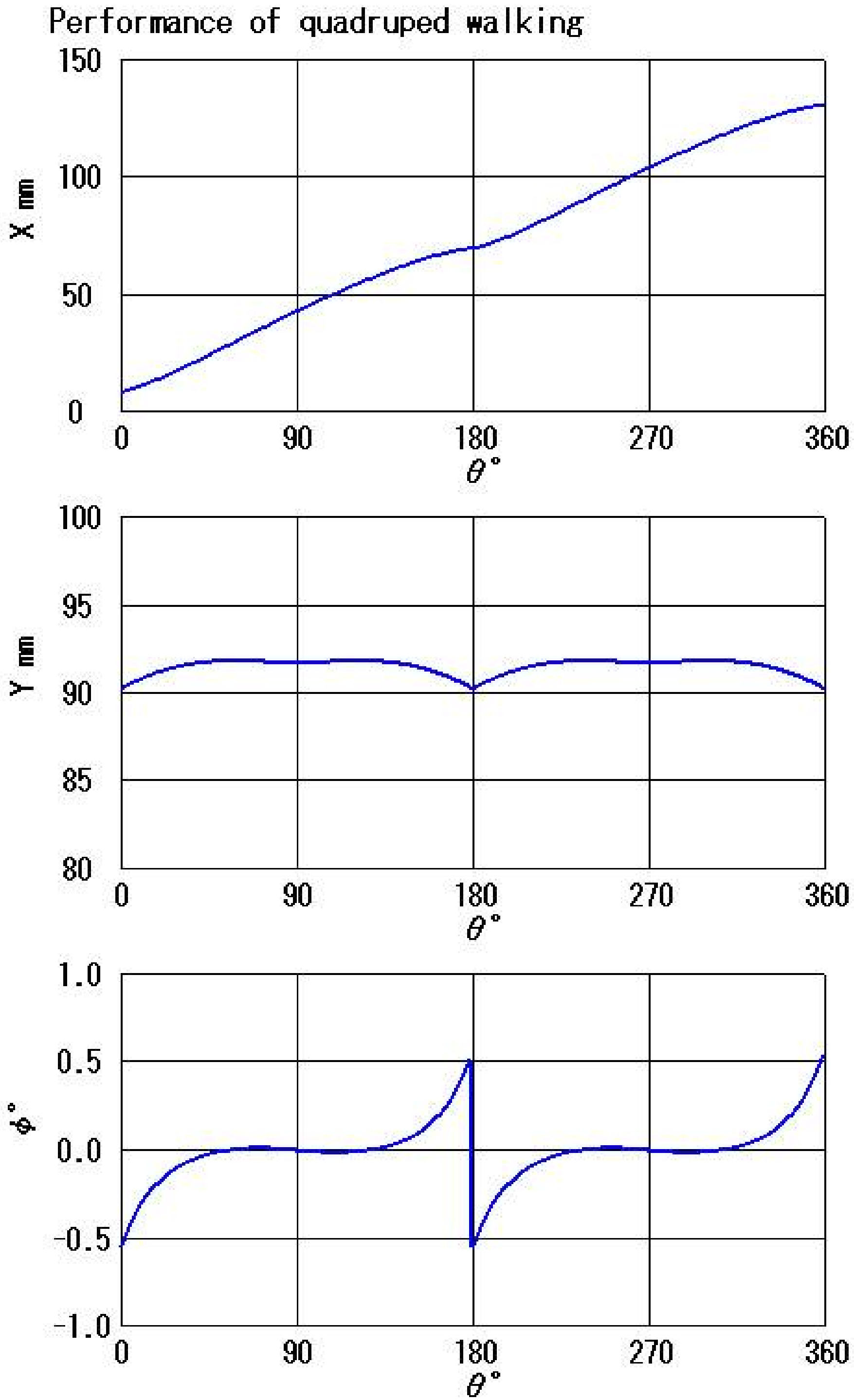
Planar 8-bar mechanism with only 1 DOF

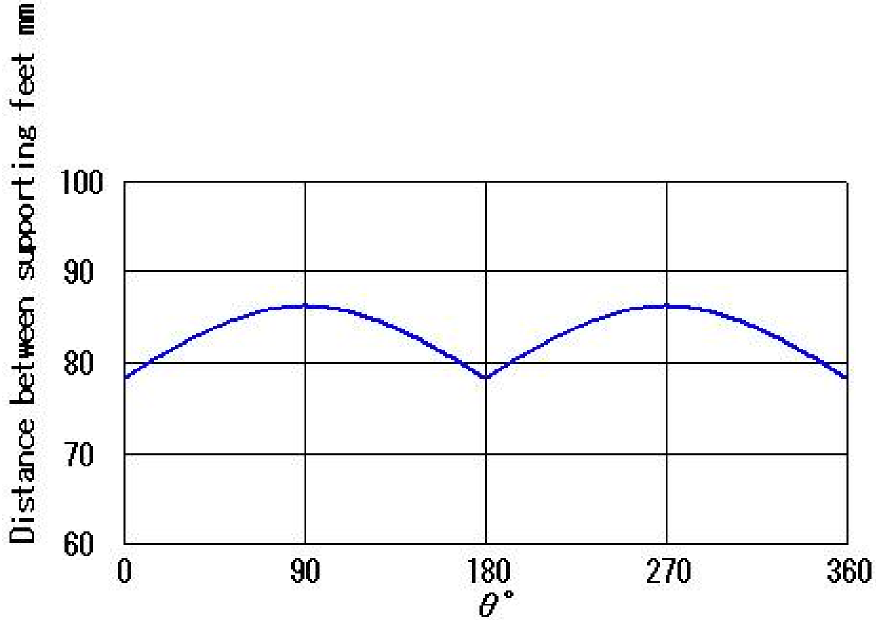


2 legs with only 1 DOF

A quadruped walking machines with only 1 DOF







Slippage between 2 legs

↓

It can walk on sand beach.

5. Concluding remarks

As an available tool to analyze or design link mechanisms is proposed, formulated and coded.

(1)Various planar mechanisms can be systematically analyzed.

(2)Not only displacement but also velocity and acceleration can be calculated and will be applied to dynamics calculation.

(3)Students are expected to experience to analyze some mechanisms with the offered programs.

Homework 1

Calculate time history of output displacement, velocity and acceleration of some planar link mechanism by using the offered program or self-coded program. It is OK to use the offered program as is.

The result will be summarized in A4 size PDF with less than 5 pages and sent to Prof. Iwatsuki via T2SCOLA by April 24, 2023.

Of course, students should describe comments for the calculated results!