Modeling, Design, and Control of Robotic Mechanisms

Han Sung Kim

Department of Mechanical Engineering,
Kyungnam University
Email: hkim@kyungnam.ac.kr
Homepage: http://imecha.kyungnam.ac.kr
(Phone: 82-55-249-2627, Fax: 82-505-999-2160)

Contents

Project Summary

- **Title: Design of Mechatronics System and Capstone Design**
- **Subtitle: Modeling, Design, and Control of Robotic Mechanisms**
- **Abstract:**

  This project is to make analytical and CAD models of robotic mechanisms (for example, industrial robots, walking robots, etc.), to design robotic mechanisms and position controller based on the models, to fabricate the robotic mechanisms and the embedded real-time controller by using xPC Target, and to do the control experiments on prototype robot system.
- **Project Goals:**

  To make kinematic & dynamic models and simulations of simple robotic mechanisms

  To design controller based on plant model & To implement real hardware
- **Lecture notes (for 15 weeks)**

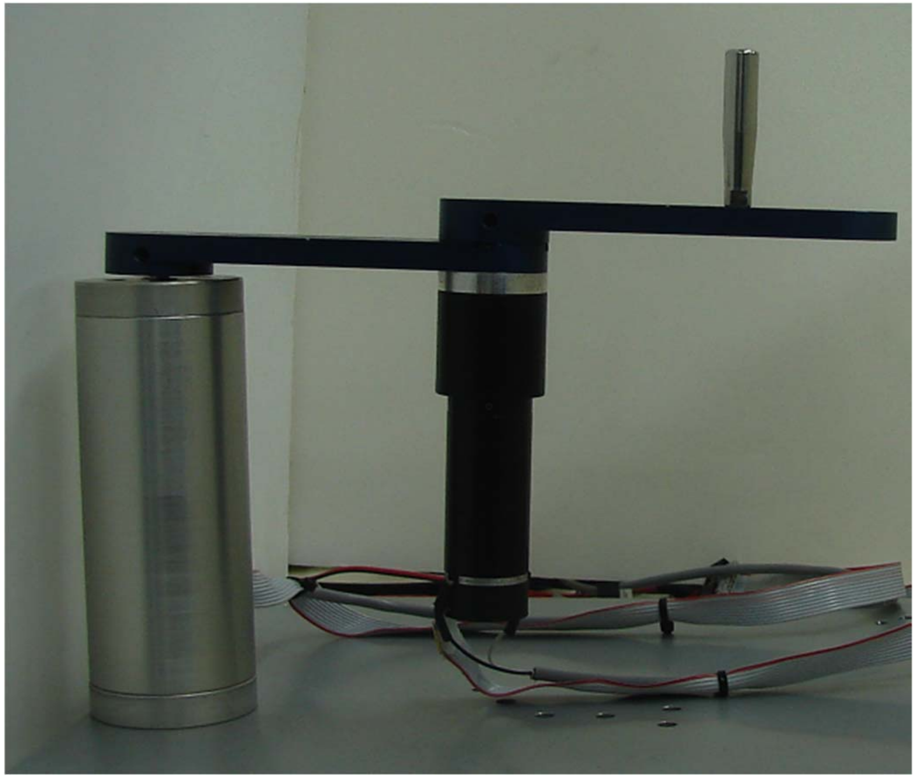
  Applicable subjects: Robotics, Mechatronics, and Capstone Design

(1) Development of Robotic Mechanisms

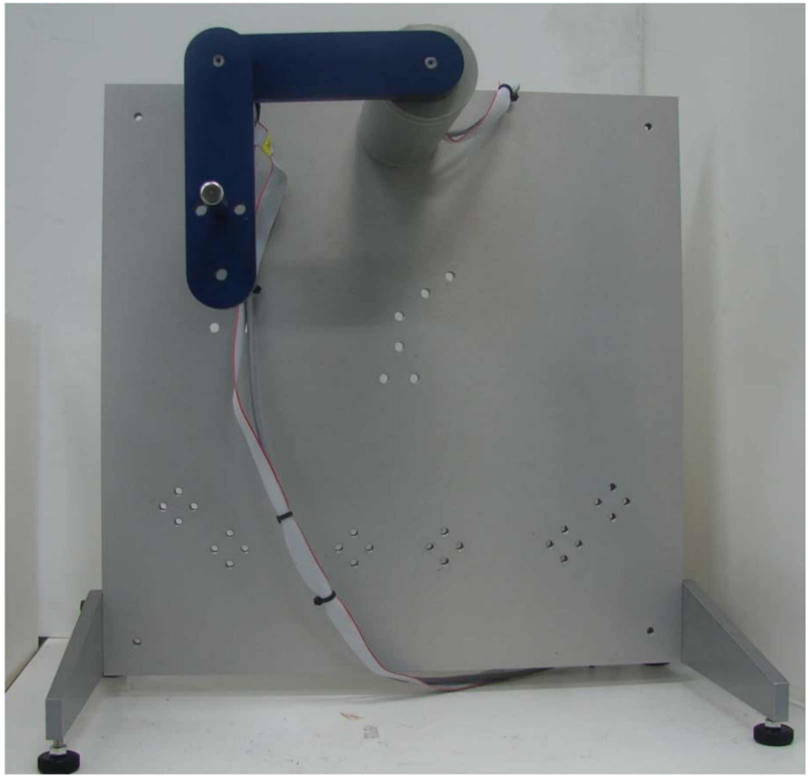
- **2-DOF/3-DOF Planar Serial Robots, 2-DOF 5-bar Closed-loop Mechanism (Type I & II), and 3-DOF 3-RRR Parallel Robot**
  - ✓ 4 DC servo motors (Maxon 20W) with one 66:1 and three 23:1 gears
  - ✓ Use the same base platform
  - ✓ Horizontal and vertical configurations (gravity compensation experiment)
  - ✓ Double ball bearings are employed for better rigidity and small friction
  - ✓ Could be used as force feedback haptic device (low gear ratio)
  - ✓ Machine vision and vision-based control experiments

- **Waling Robots with 6-bar and 8-bar Legs**
  - ✓ Actuated by two DC motors and Move forward/backward/right/left
  - ✓ Statically balanced walking robot

2-DOF Planar Serial Robot (R-R)



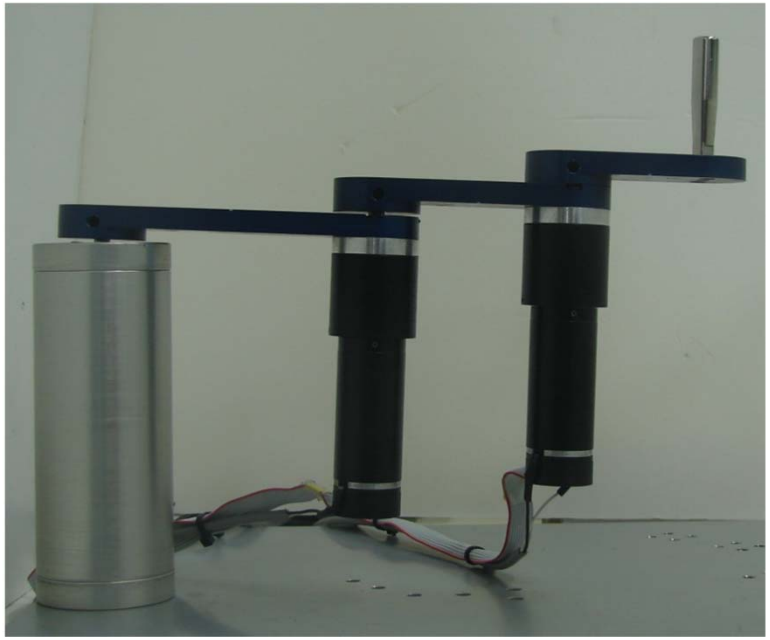
[Horizontal Configuration]



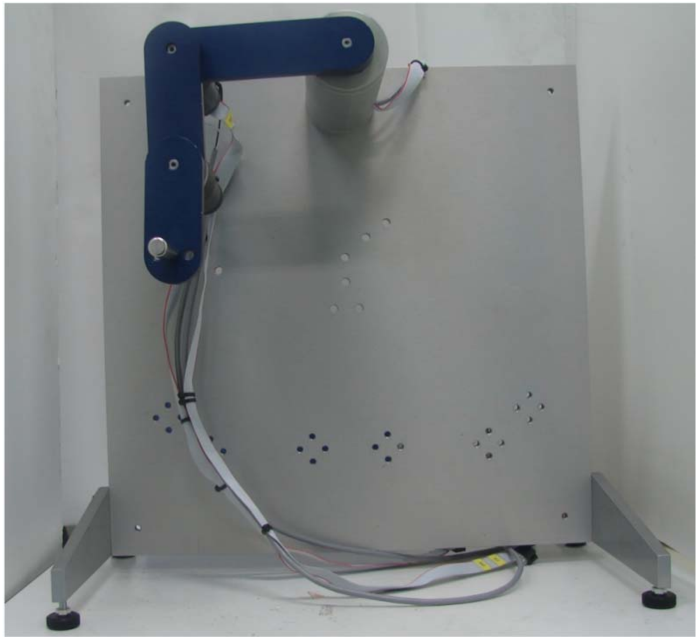
[Vertical Configuration]

MOVIE

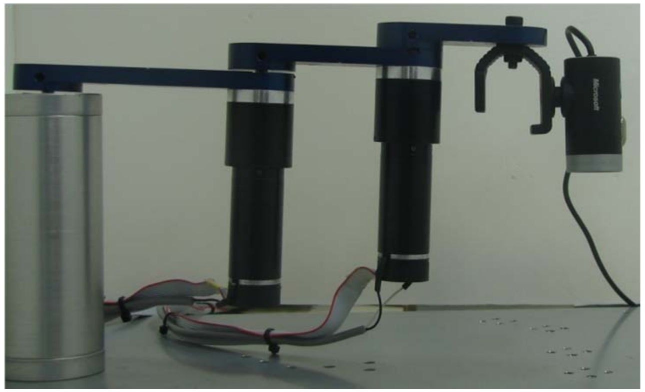
3-DOF Planar Serial Robot (R-R-R)



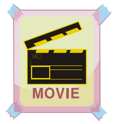
[Horizontal Configuration]



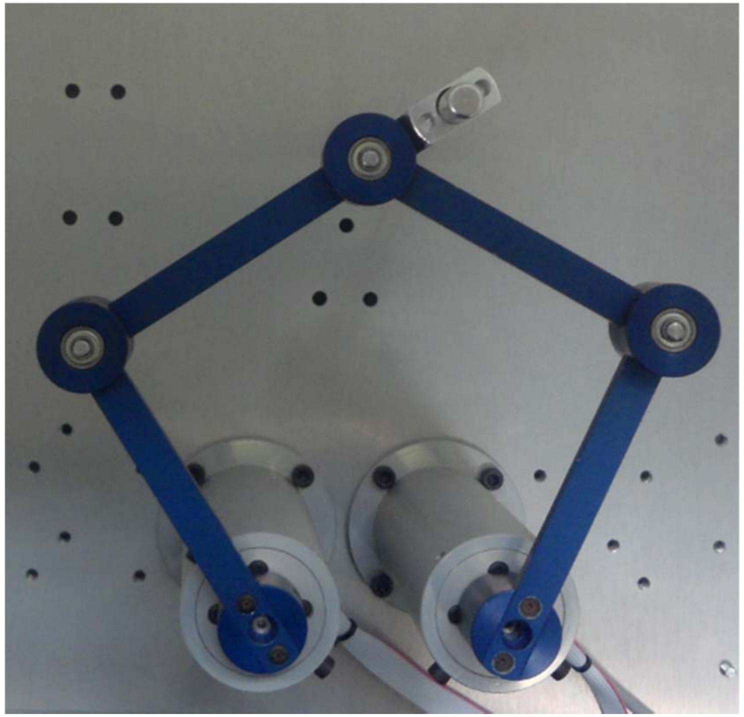
[Vertical Configuration]



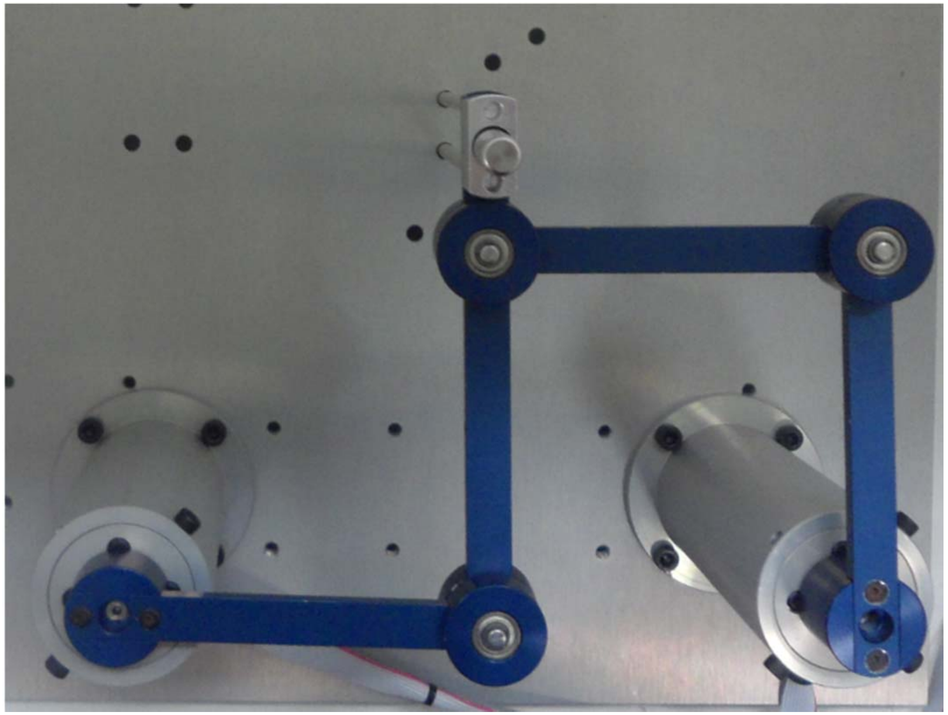
[PC CAM]



2-DOF 5-bar Closed-loop Mechanisms



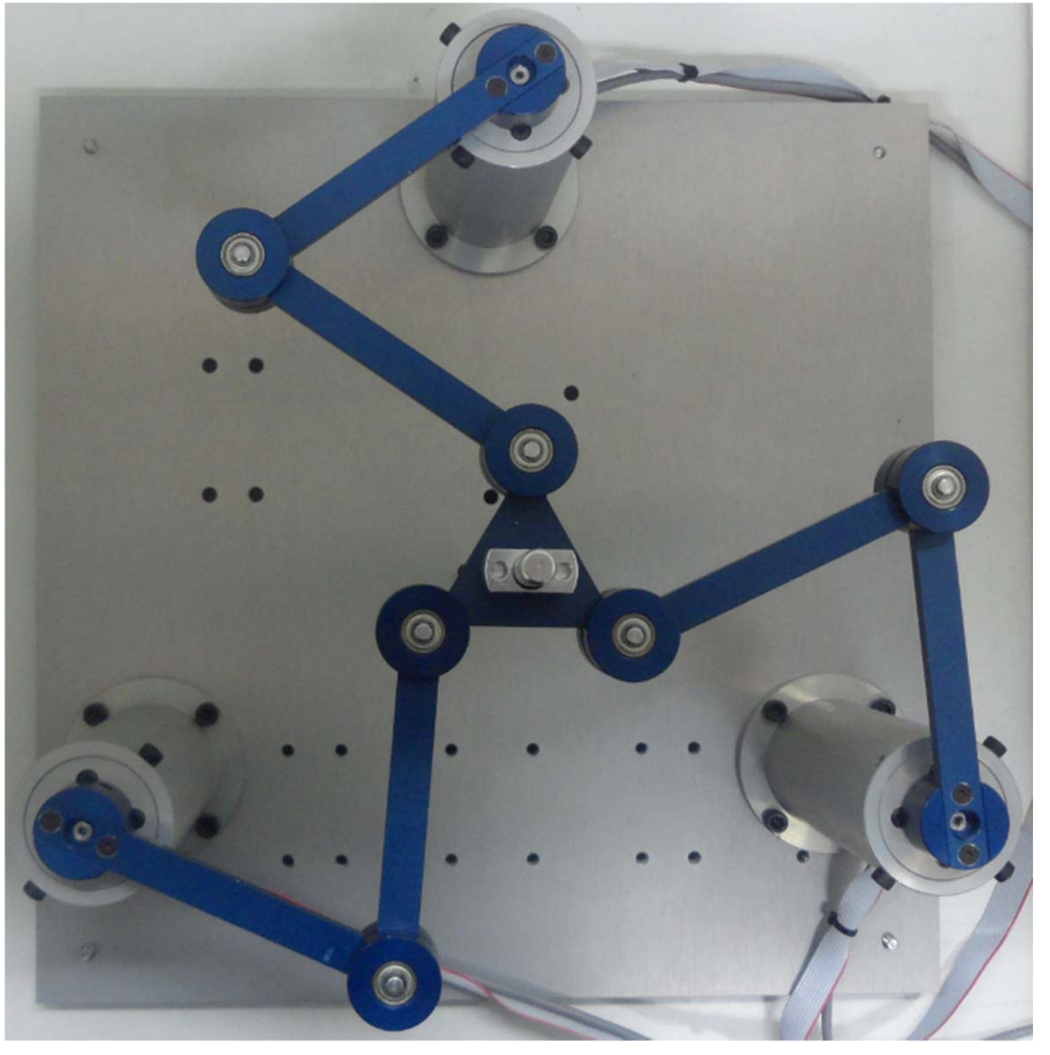
[Type I]



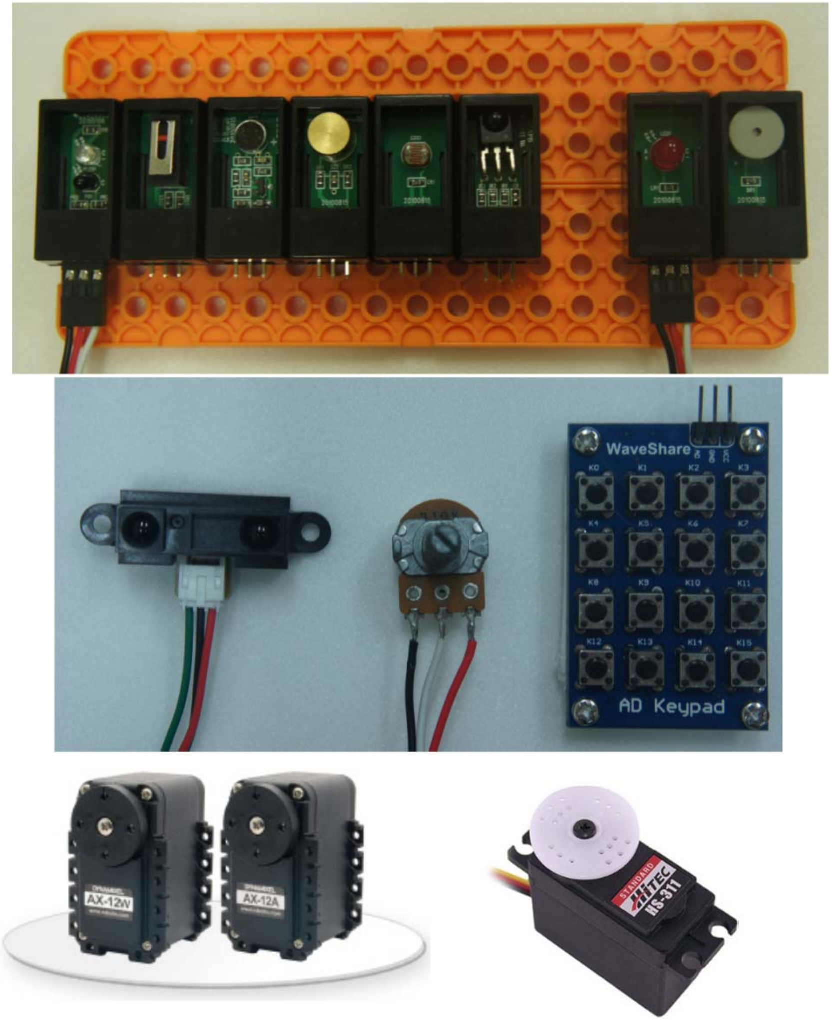
[Type II]

MOVIE

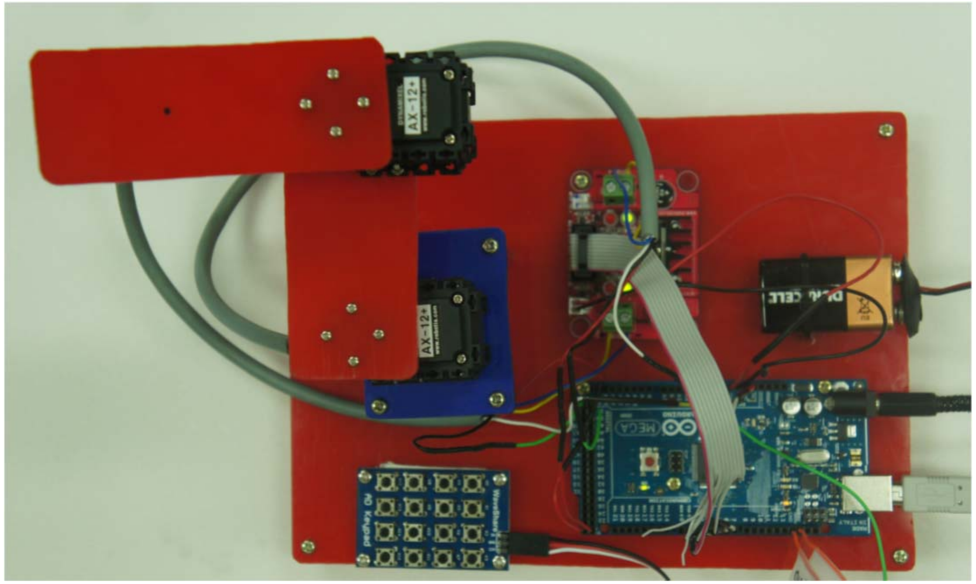
3-DOF Parallel Robot (3-RRR)



Arduino Projects



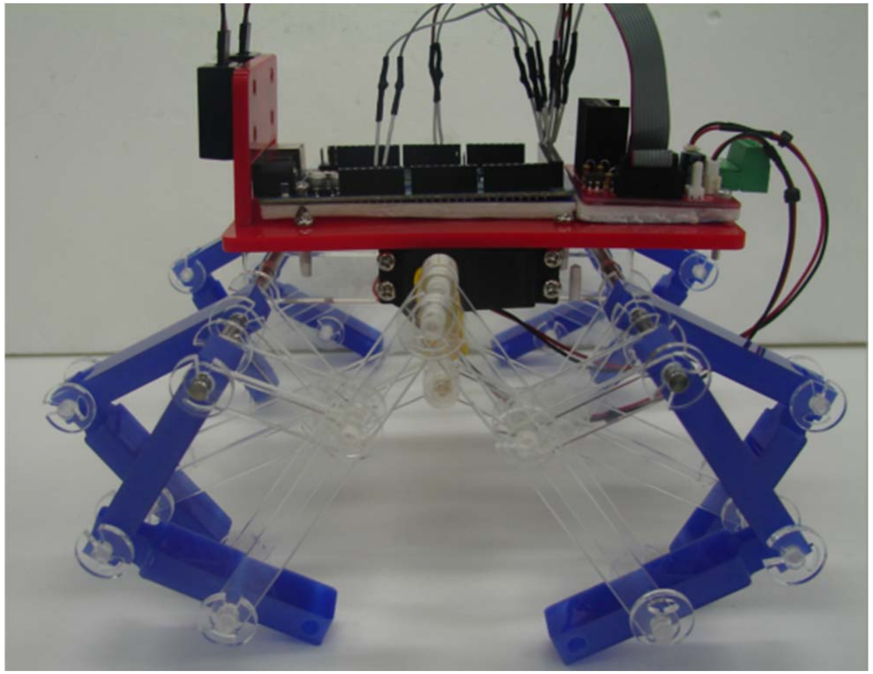
[Sensor & Actuator Interface]



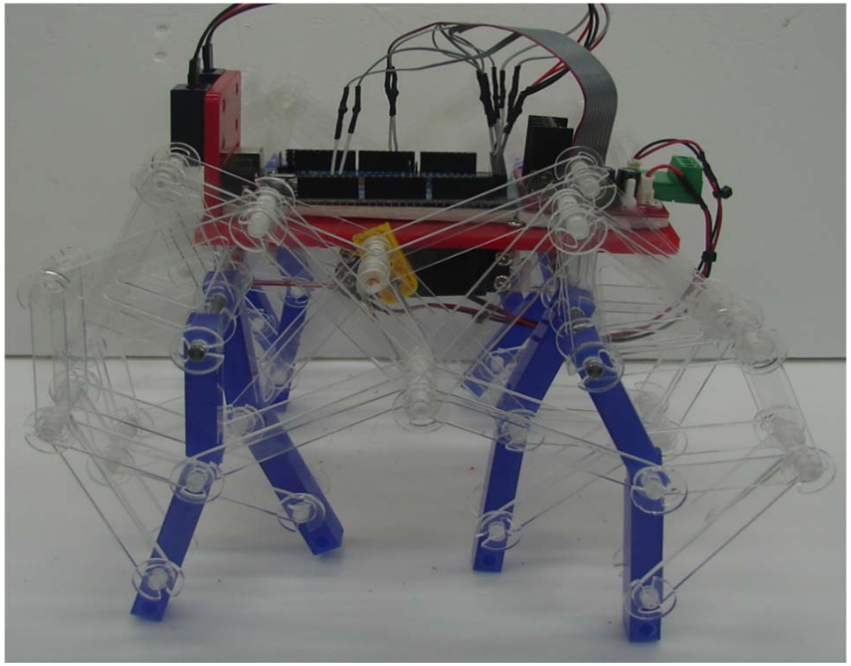
[2-DOF Robot Arm with Arduino Mega 2560]

MOVIE

Walking Robots



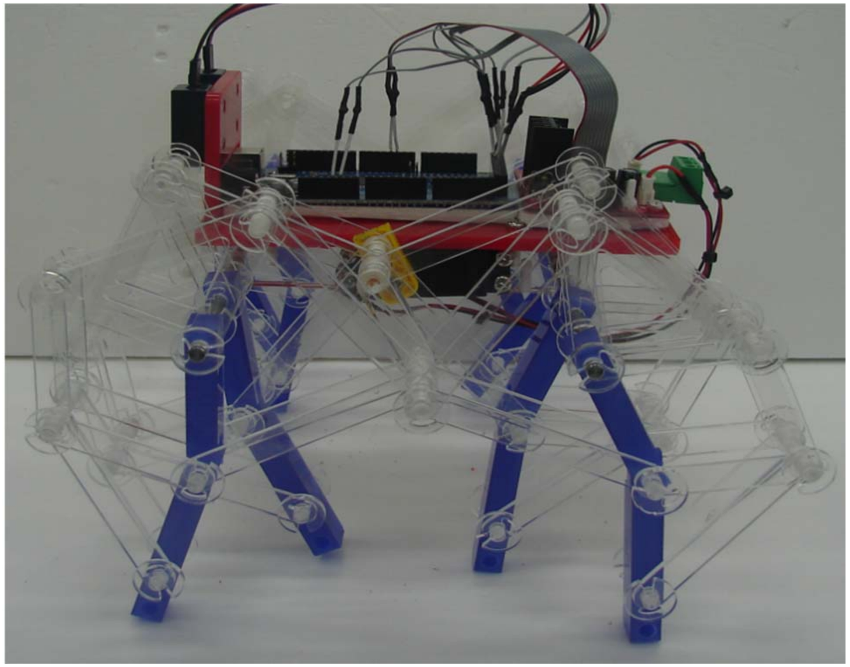
[with 6-bar Legs]



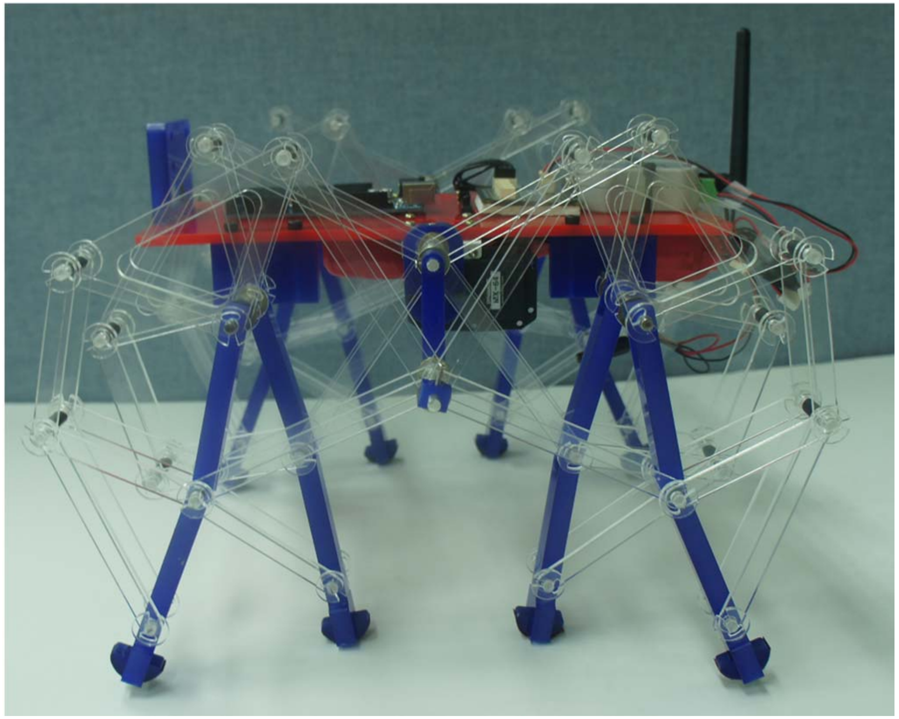
[with 8-bar Legs]

MOVIE

Walking Robots



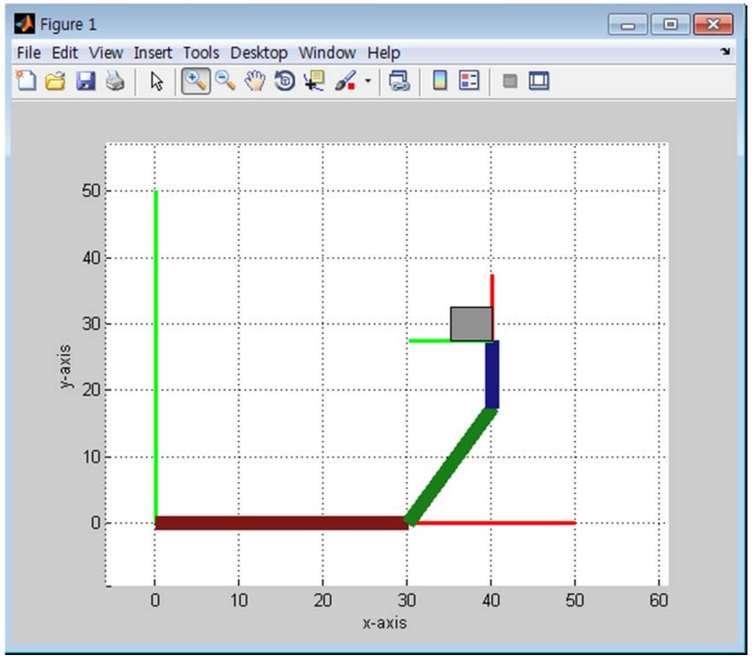
[DC Motor & External Power]



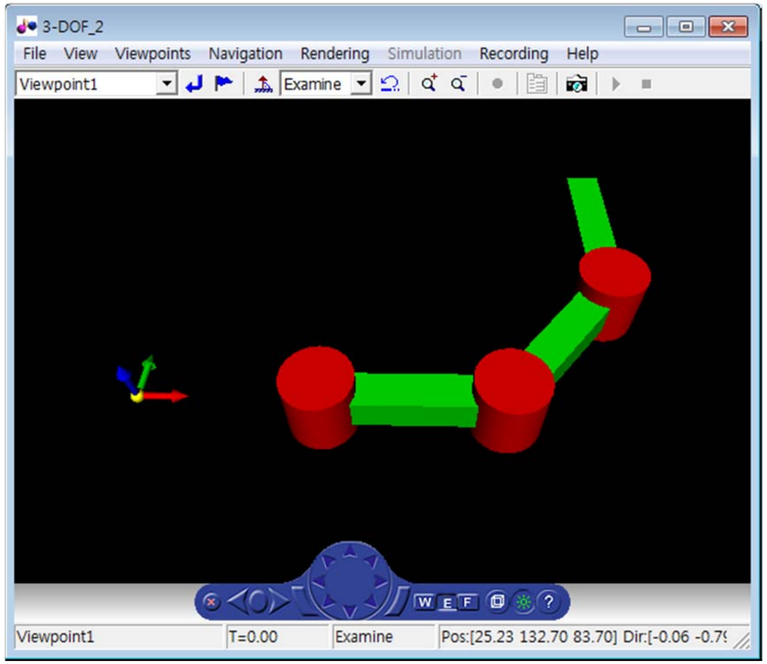
[Servo Motor, Wireless & Battery]

(2) Analysis and MATLAB Practice

- Forward & Inverse Kinematic Analysis
- Velocity & Statics Analysis
- Forward & Inverse Dynamics Analysis
- MATLAB Practices (MATLAB Demo)

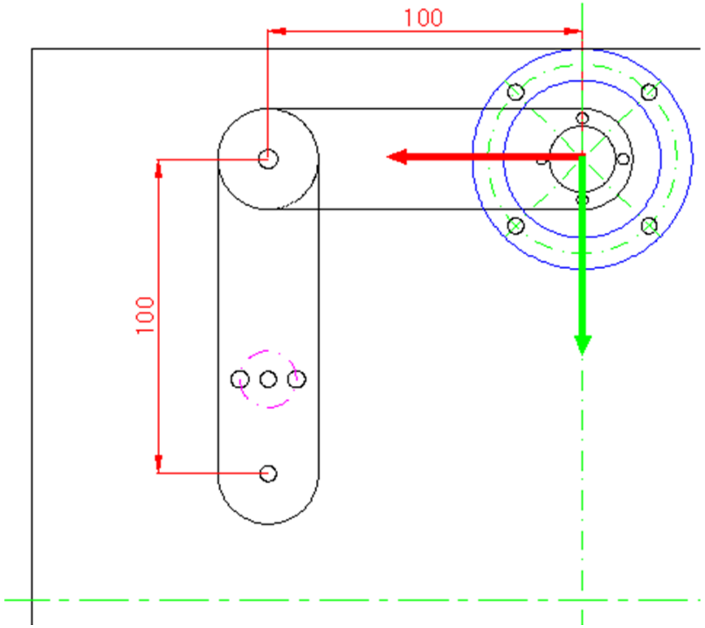


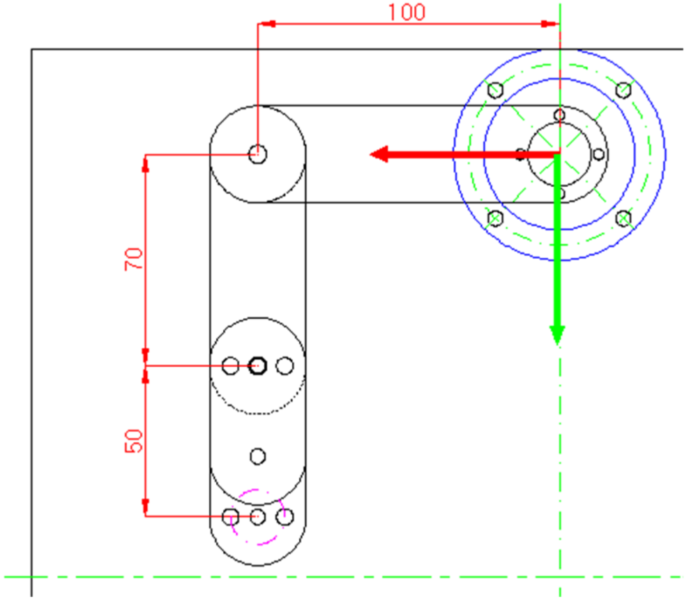
[MATLAB Figure]



[VRML Output]

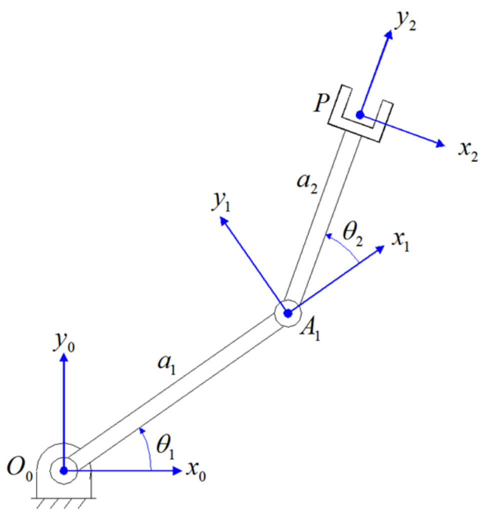
Dimensions of Serial Robots

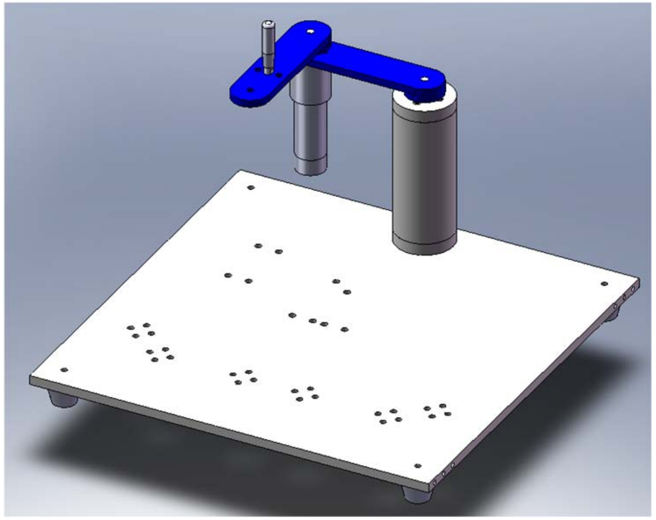




[2-DOF Serial Robot]	[3-DOF Serial Robot]
- Motor1: 66:1, Motor2: 23:1	- Motor1: 66:1, Motor2,3: 23:1
- $a_1 = 100$, $a_2 = 100$ mm	- $a_1 = 100$, $a_2 = 70$, $a_3 = 50$ mm
- $\theta_1 = 0^\circ, \theta_2 = 90^\circ$	- $\theta_1 = 0^\circ, \theta_2 = 90^\circ, \theta_3 = 0^\circ$
- $p_x = 100, p_y = 100$ mm	- $p_x = 100, p_y = 120$ mm, ϕ=90°

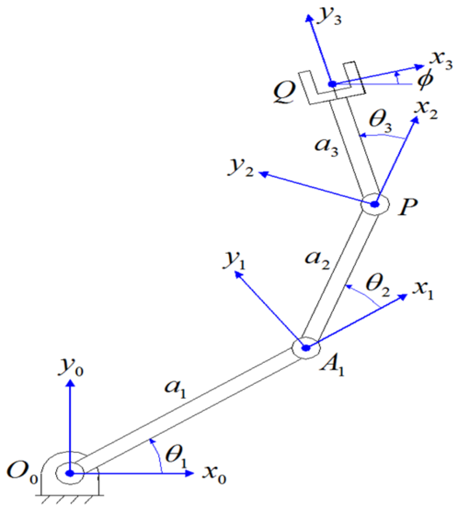
Analysis of 2-DOF Serial Robot

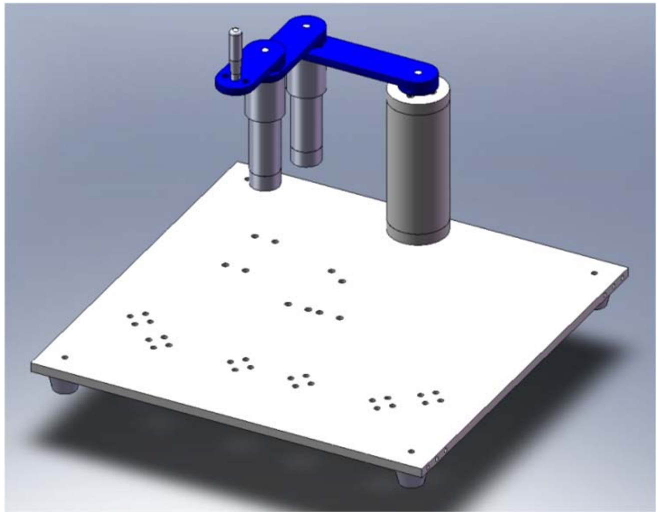




- Forward Kinematics: Find p_x, p_y for given θ_1, θ_2
- Inverse Kinematics: Find θ_1, θ_2 for given p_x, p_y
- Velocity & Statics: $\begin{bmatrix} v_x \\ v_y \end{bmatrix} = J \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix}, \quad \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} = J^T \begin{bmatrix} f_x \\ f_y \end{bmatrix}$
- Dynamics: $\begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} + \begin{bmatrix} C_1(\dot{\theta}_1, \dot{\theta}_2) \\ C_2(\dot{\theta}_1, \dot{\theta}_2) \end{bmatrix} + \begin{bmatrix} G_1 \\ G_2 \end{bmatrix}$

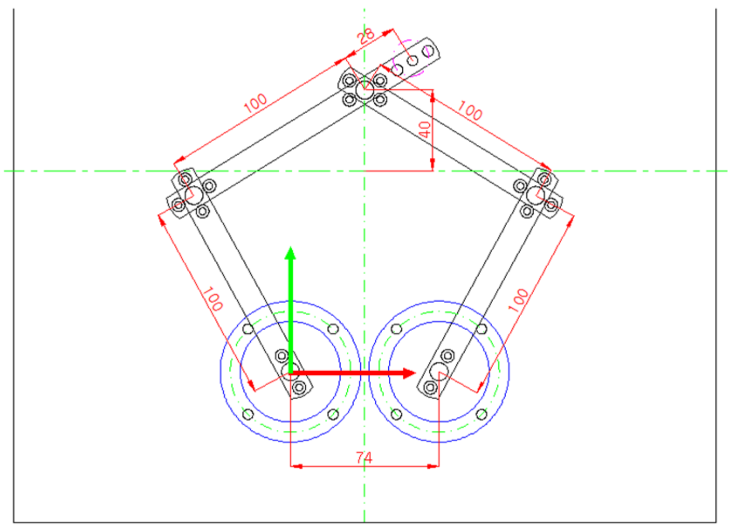
Analysis of 3-DOF Serial Robot

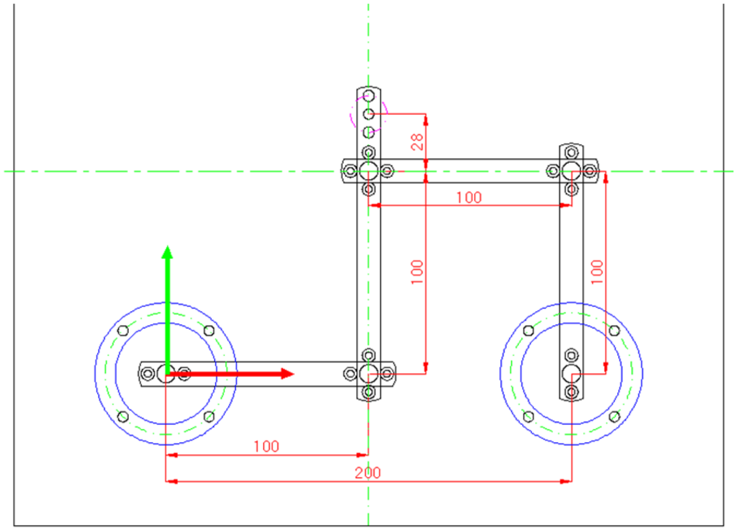




- Forward Kinematics: Find p_x, p_y, ϕ for given $\theta_1, \theta_2, \theta_3$
- Inverse Kinematics: Find $\theta_1, \theta_2, \theta_3$ for given p_x, p_y, ϕ
- Velocity & Statics: $\begin{bmatrix} v_x \\ v_x \\ \omega_z \end{bmatrix} = J \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix}, \quad \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} = J^T \begin{bmatrix} f_x \\ f_y \\ n_z \end{bmatrix}$
- Dynamics: $\begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} = \begin{bmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \\ \ddot{\theta}_3 \end{bmatrix} + \begin{bmatrix} C_1(\dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3) \\ C_2(\dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3) \\ C_3(\dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3) \end{bmatrix} + \begin{bmatrix} G_1 \\ G_2 \\ G_3 \end{bmatrix}$

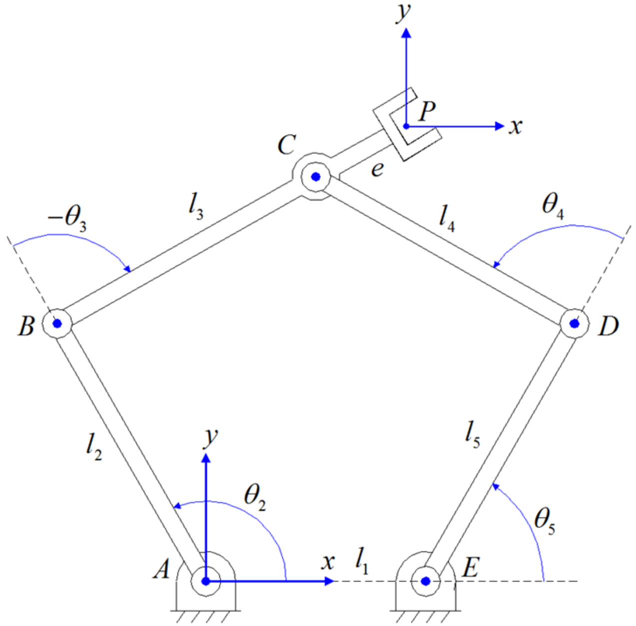
Dimensions of 5-bar Mechanisms

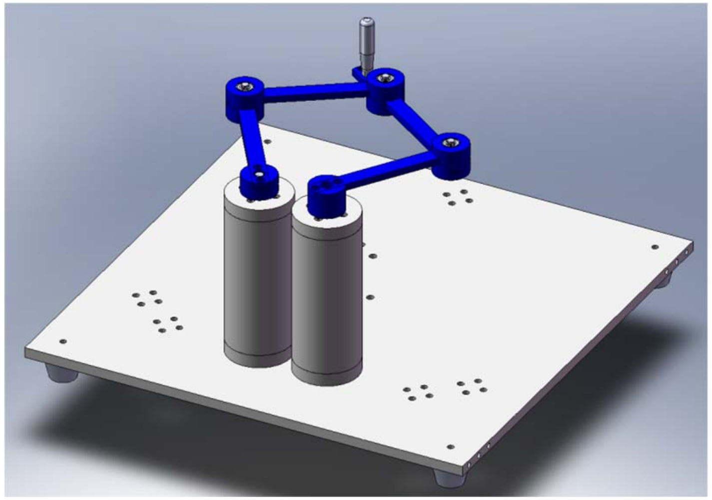




[5-bar Type I]	[5-bar Type II]
- Motor1,2: 23:1	- Motor1,2: 23:1
- $l_1 = 74,\ l_3 = 128,\ l_2 = l_4 = l_5 = 100$	- $l_1 = 200,\ l_3 = 128,\ l_2 = l_4 = l_5 = 100$
- $\theta_2 = 118.81^\circ, \theta_5 = 61.19^\circ$	- $\theta_2 = 0^\circ, \theta_5 = 90^\circ$
- $p_x = 60.8523, p_y = 154.6652$ mm	- $p_x = 100, p_y = 128$ mm

Analysis of 5-bar Mechanism (Type I)

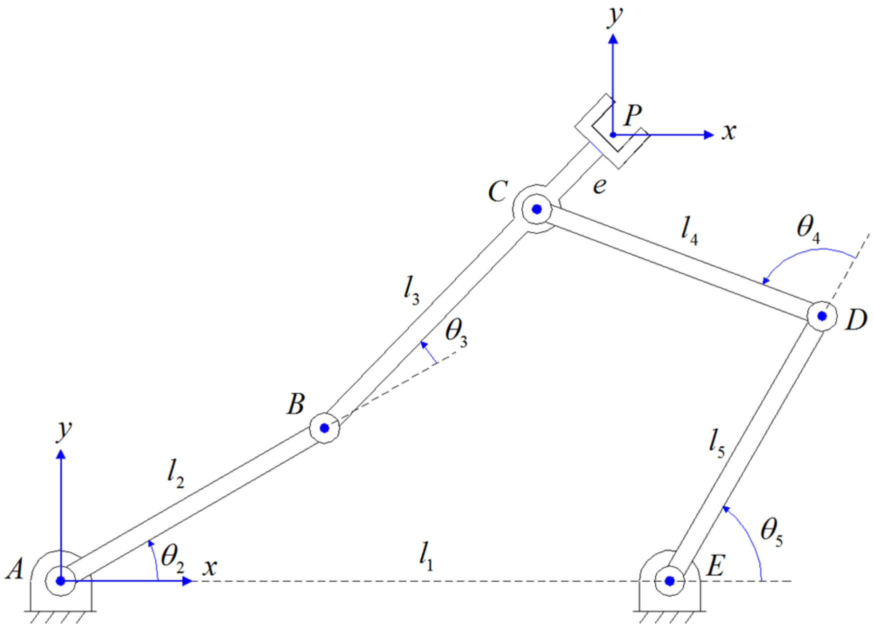


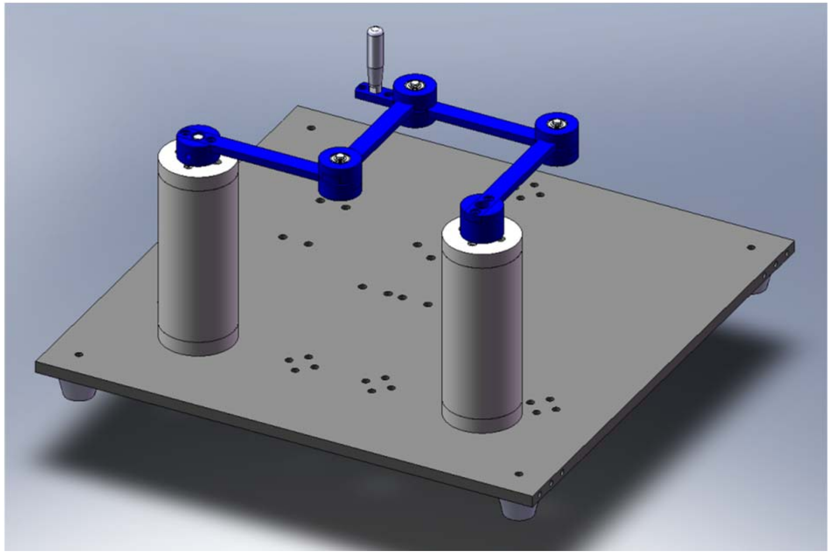


- Forward Kinematics: Find p_x, p_y for given θ_2, θ_5
- Inverse Kinematics: Find θ_2, θ_5 for given p_x, p_y

- Velocity & Statics: $\begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} = J^T \begin{bmatrix} v_x \\ v_y \end{bmatrix}, \begin{bmatrix} f_x \\ f_y \end{bmatrix} = J \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix}$

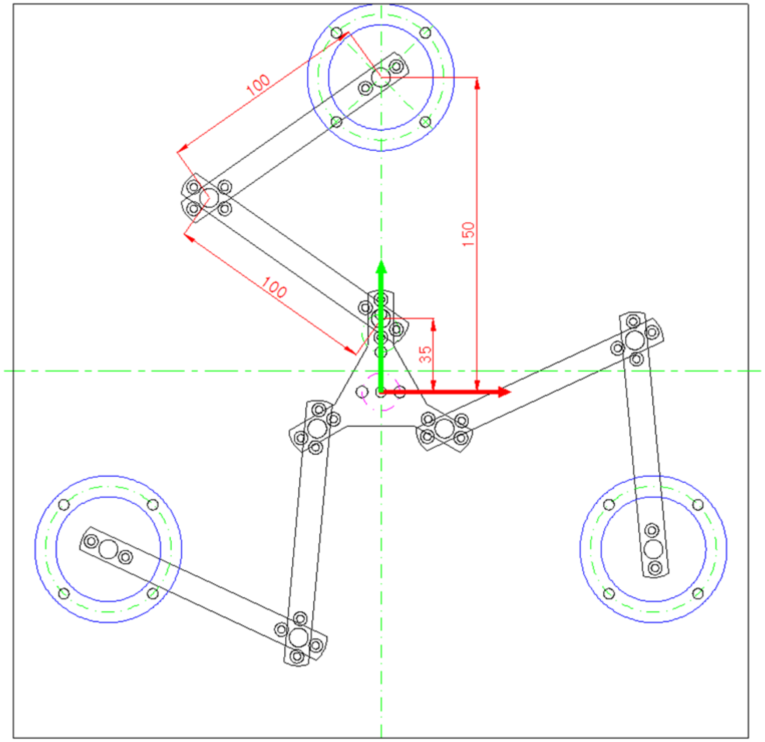
Analysis of 5-bar Mechanism (Type II)





- Forward Kinematics: Find p_x, p_y for given θ_2, θ_5
- Inverse Kinematics: Find θ_2, θ_5 for given p_x, p_y
- Velocity & Statics: $\begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} = J^T \begin{bmatrix} v_x \\ v_y \end{bmatrix}, \begin{bmatrix} f_x \\ f_y \end{bmatrix} = J \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix}$

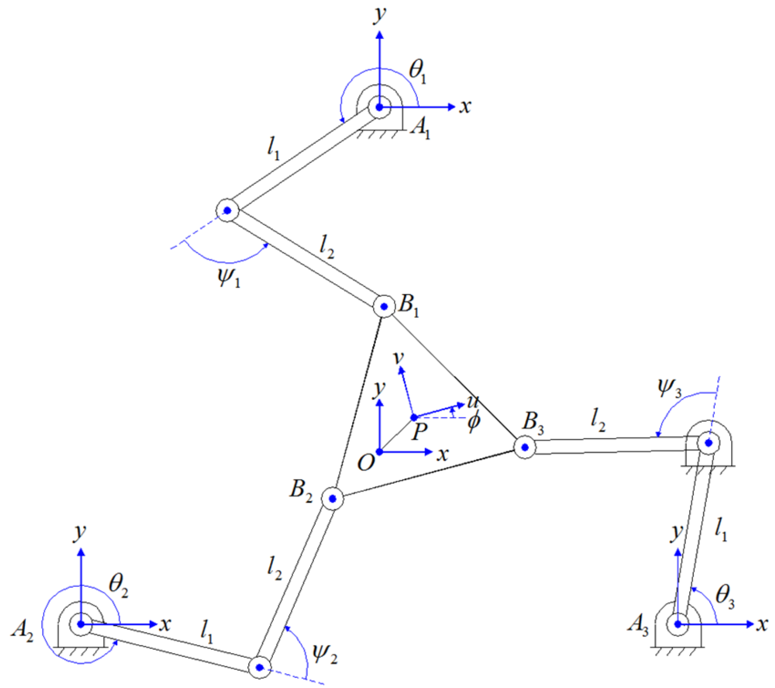
Dimensions of 3-DOF Parallel Robot

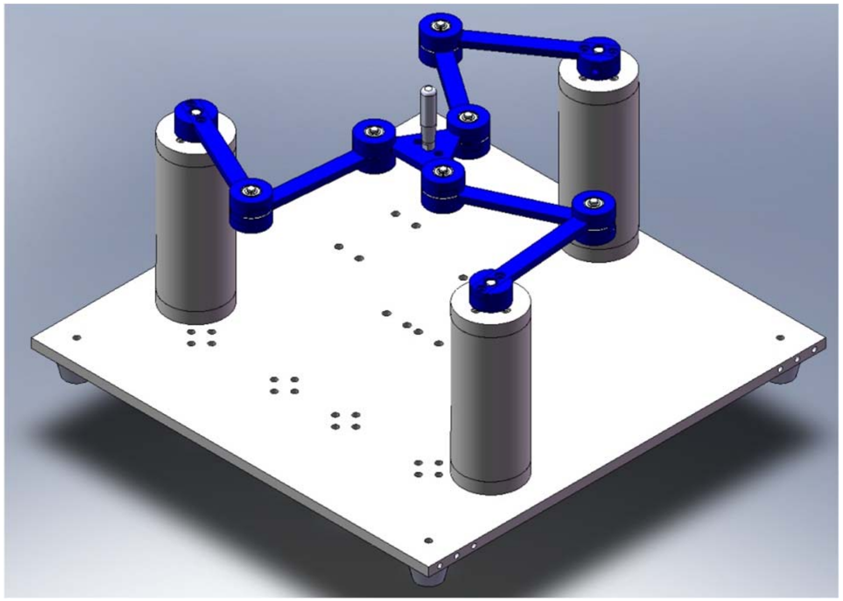


[3-DOF 병렬로봇]

- 1축, 2축, 3축: 23:1
- $a = 150$, $b = 35$, $l_1 = l_2 = 100$ [mm]
- $\theta_1 = -144.90^\circ, \theta_2 = -24.90^\circ, \theta_3 = 95.10^\circ$
- $p_x = 0, p_y = 0$ mm, $\phi = 0^\circ$

Analysis of 3-DOF Parallel Robot





- Forward Kinematics: Find p_x, p_y, ϕ for given $\theta_1, \theta_2, \theta_3$
- Inverse Kinematics: Find $\theta_1, \theta_2, \theta_3$ for given p_x, p_y, ϕ
- Velocity & Statics: $\begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix} = J^T \begin{bmatrix} v_x \\ v_x \\ \omega_z \end{bmatrix}, \begin{bmatrix} f_x \\ f_y \\ n_z \end{bmatrix} = J \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix}$

[3] SimMechanics Simulation Practice

- **Forward Dynamics Simulation**
  - ✓ Apply forces → Find motions
  - ✓ Numerical integration (ode45)

- **Inverse Dynamics Simulation**
  - ✓ Specify motions (Trajectory) → Find joint torques
  - ✓ Design of actuator size
  - ✓ Computed torque control (simple gravity compensation)

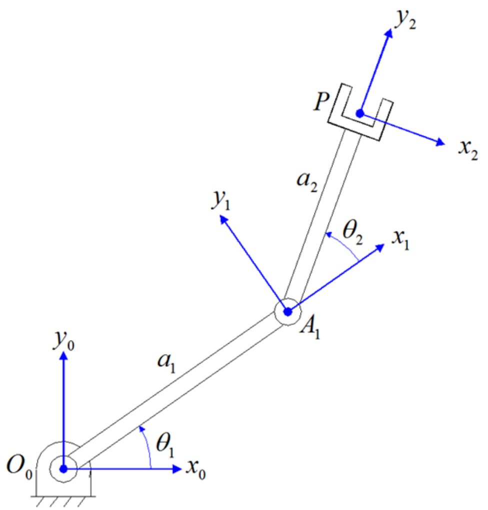
$$\begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} + \begin{bmatrix} C_1(\dot{\theta}_1, \dot{\theta}_2) \\ C_2(\dot{\theta}_1, \dot{\theta}_2) \end{bmatrix} + \begin{bmatrix} G_1 \\ G_2 \end{bmatrix}$$

- **Comparing SimMechanics results with those of the Dynamics Equations and ADAMS simulations**

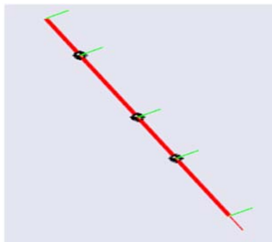
MOVIE

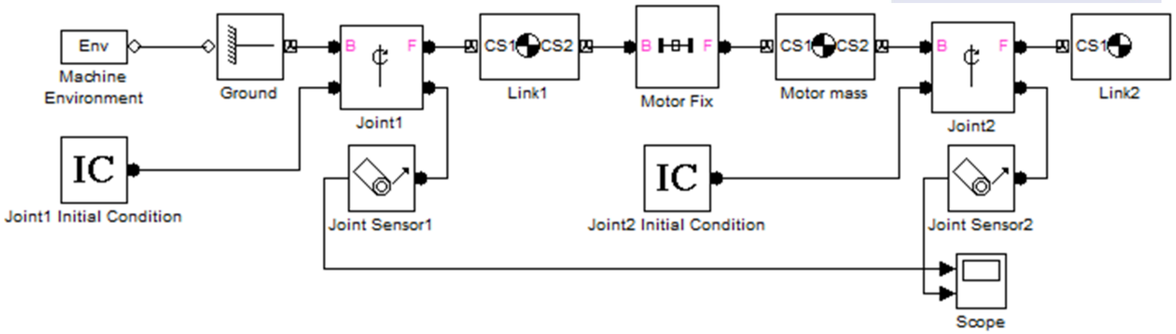
Forward Dynamics Simulation (2-DOF Serial Robot)

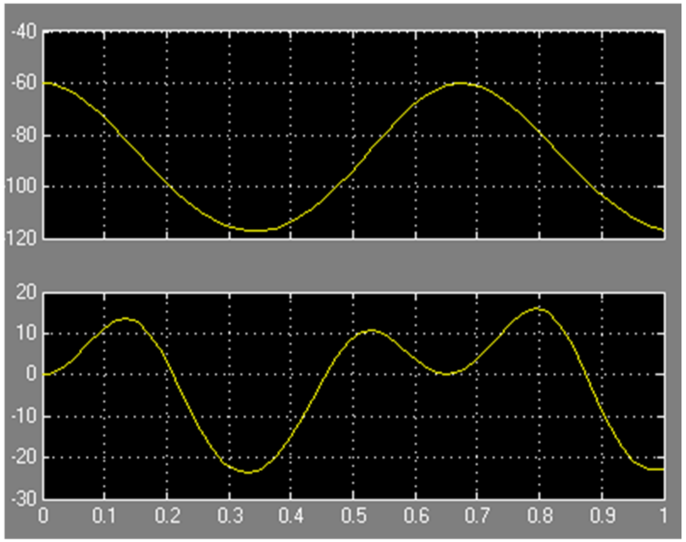
- **Initial Condition: $[\theta_1, \theta_2]$=[-60,0]**



	Mass(kg)	Inertia (kg mm²)	CG Position(m) /from CS1	CS(m) /from World
Ground	-	-	-	[0 0 0]
Link1	0.071	89.395	[0.0373 0 0]	CS1 = [0 0 0] CS2 = [0.1 0 0]
Motor	0.292	0	[0 0 0]	CS1 = [0.1 0 0] CS2 = [0.1 0 0]
Link2	0.075	109.864	[0.04168 0 0]	CS1 = [0.1 0 0] CS2 = [0.2 0 0]

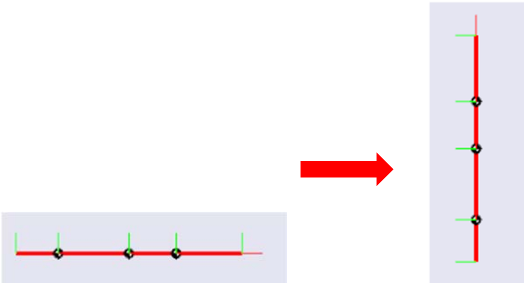


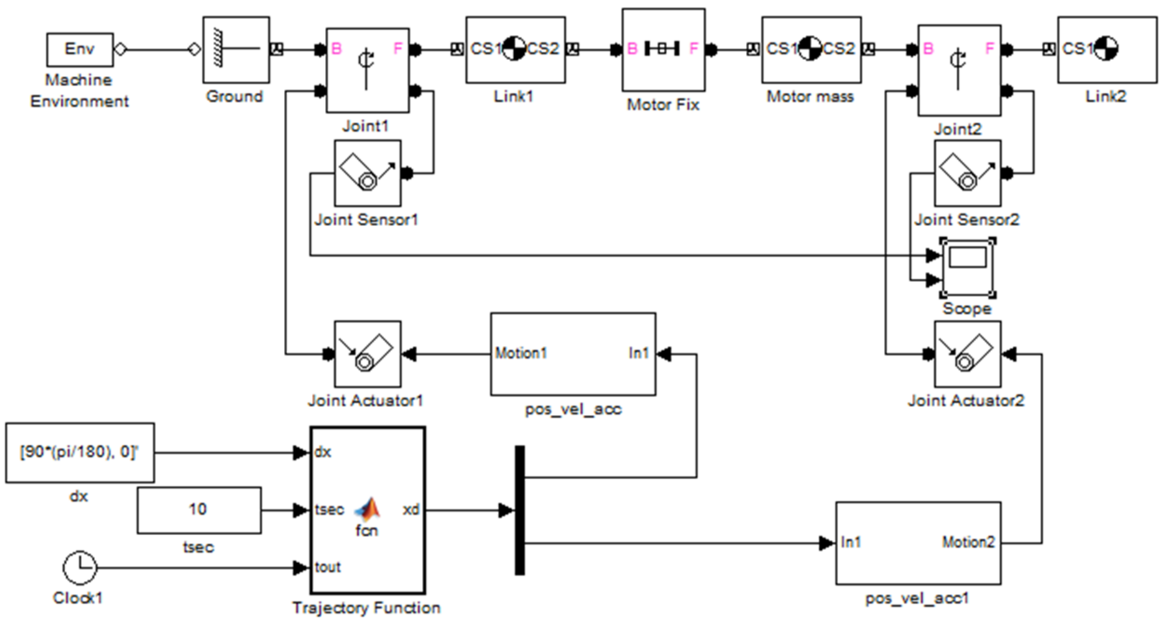


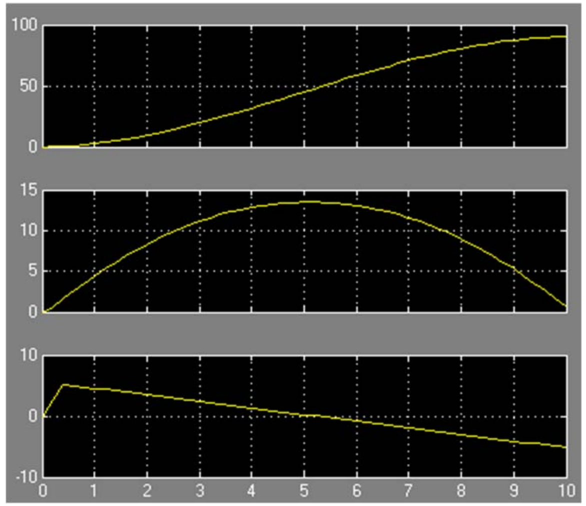


Inverse Dynamics Simulation (2-DOF Serial Robot)

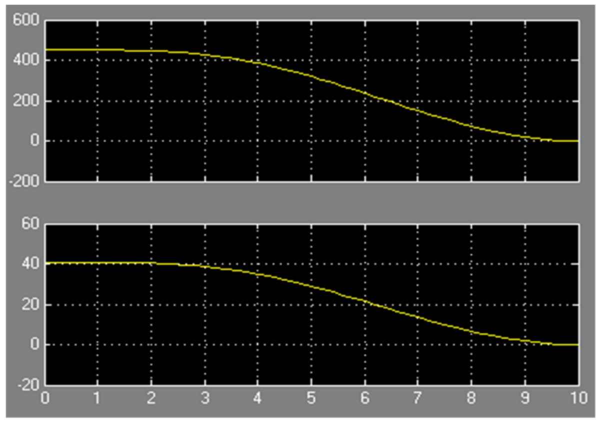
- **Joint-space Simulation: [θ_1, θ_2]=[0,0] → [90,0]**







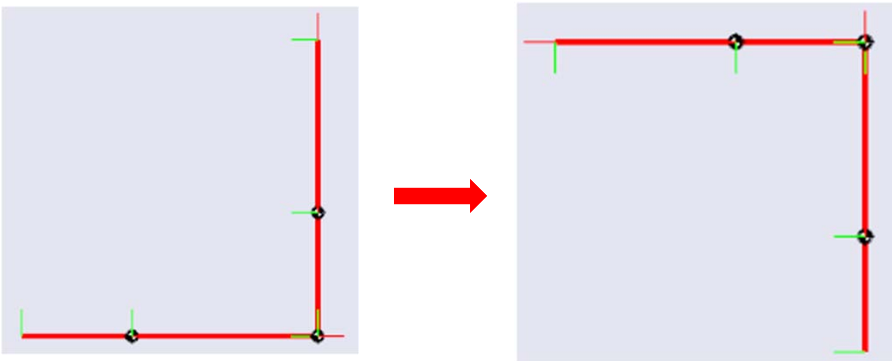
[Cubic Trajectory]

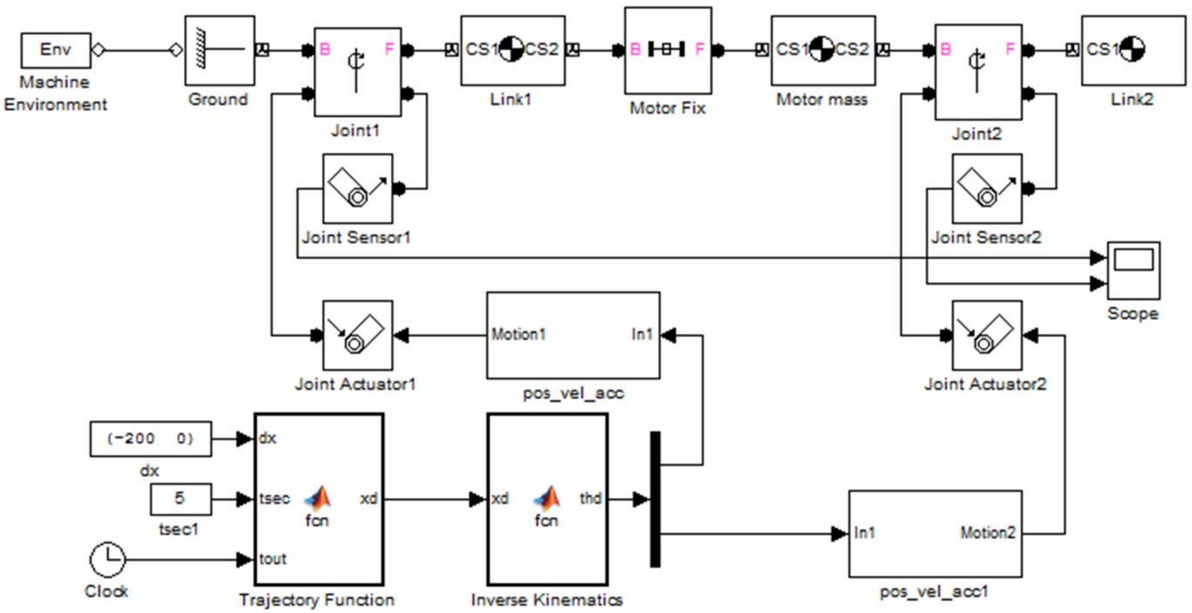


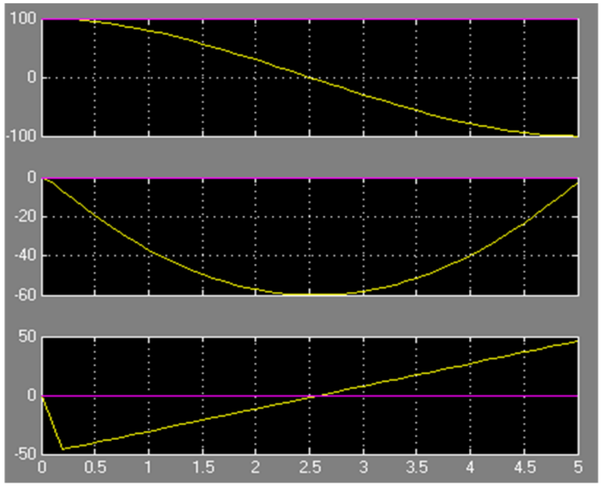
[Joint Torque]

Inverse Dynamics Simulation (2-DOF Serial Robot)

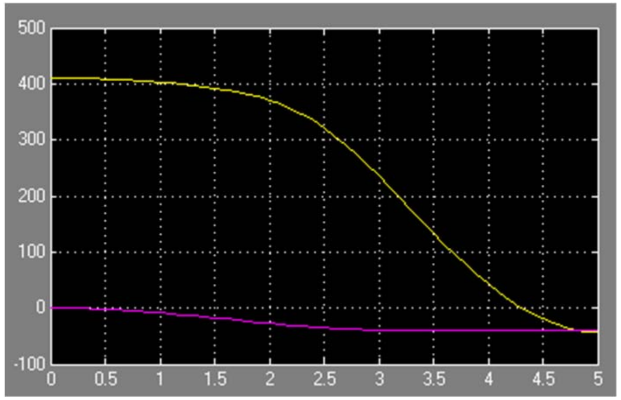
- **Cartesian-space Simulation: [p_x, p_y]=[100,100] → [-100,100]**







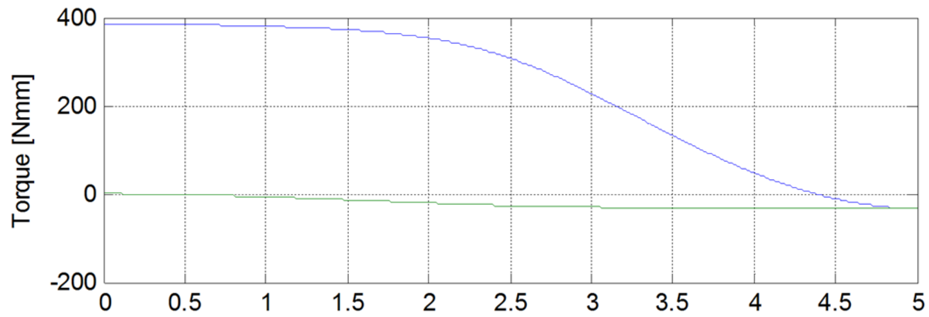
[Cubic Trajectory]



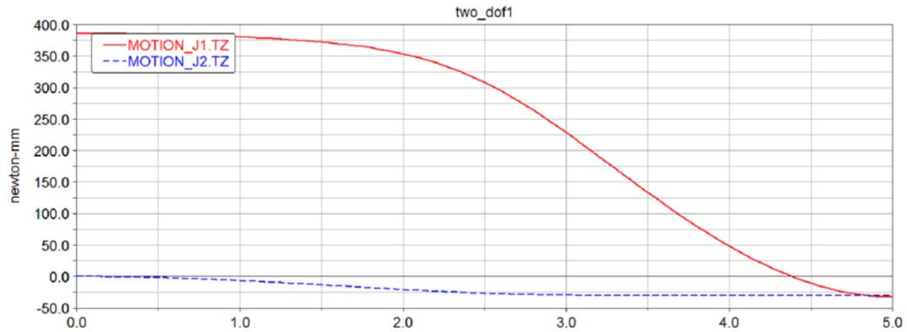
[Joint Torque]

Comparison of Simulation Results (2-DOF Serial Robot)

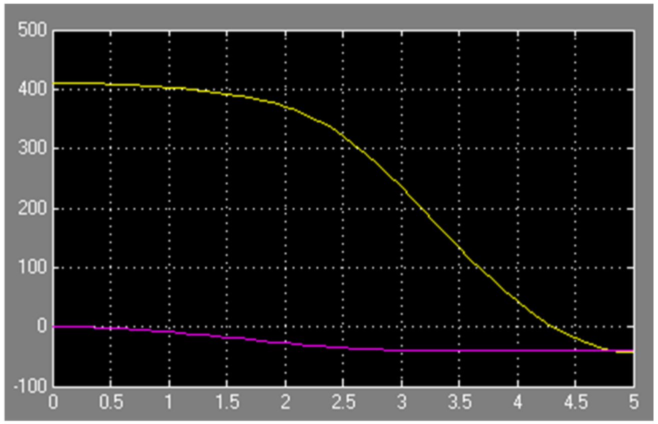
- **Dynamics Equation:**



- **ADAMS Result:**

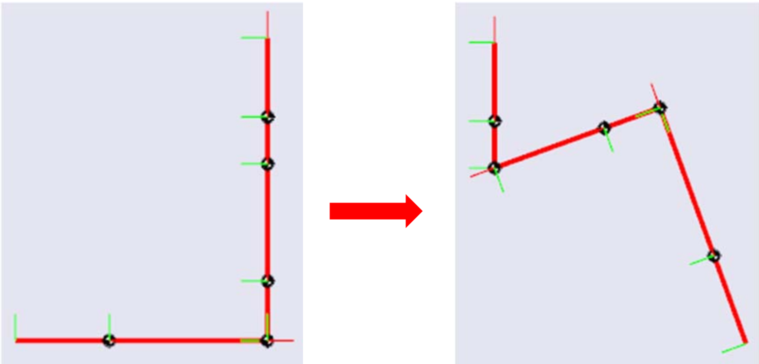


- **SimMechanics Result:**

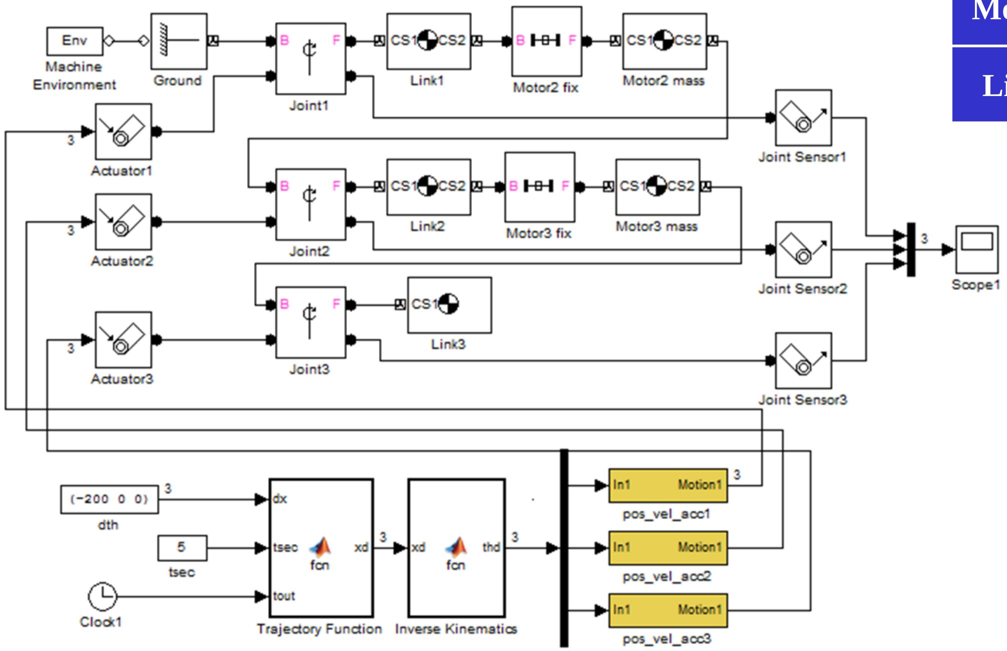


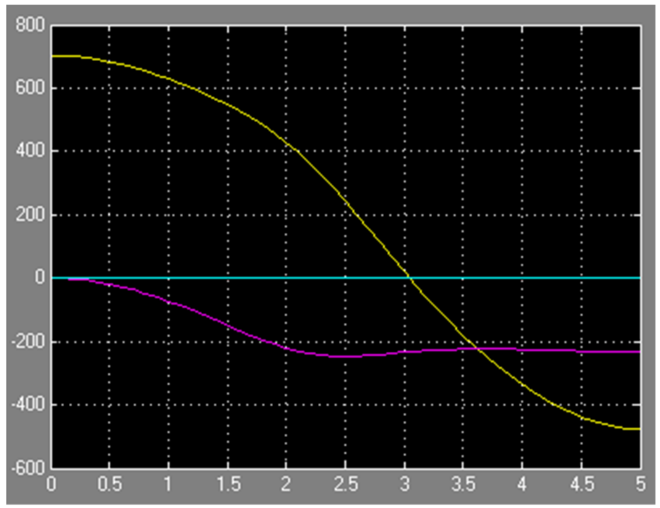
Inverse Dynamics Simulation (3-DOF Serial Robot)

- Cartesian-space Simulation: $[p_x, p_y]=[100,100]$, $\phi=90$ → $[-100,100]$, $\phi=90$



	Mass (kg)	Inertia (kg mm²)	CG Position(m) /from CS1	CS(m) /from World
Ground	-	-	-	[0 0 0]
Link1	0.071	89.395	[0.0373 0 0]	CS1 = [0 0 0] CS2 = [0.1 0 0]
Motor2	0.292	0	[0 0 0]	CS1 = [0.1 0 0] CS2 = [0.1 0 0]
Link2	0.055	38.709	[0.0236 0 0]	CS1 = [0.1 0 0] CS2 = [0.17 0 0]
Motor3	0.292	0	[0 0 0]	CS1 = [0.17 0 0] CS2 = [0.17 0 0]
Link3	0.05	24.587	[0.0186 0 0]	CS1 = [0.17 0 0] CS2 = [0.22 0 0]

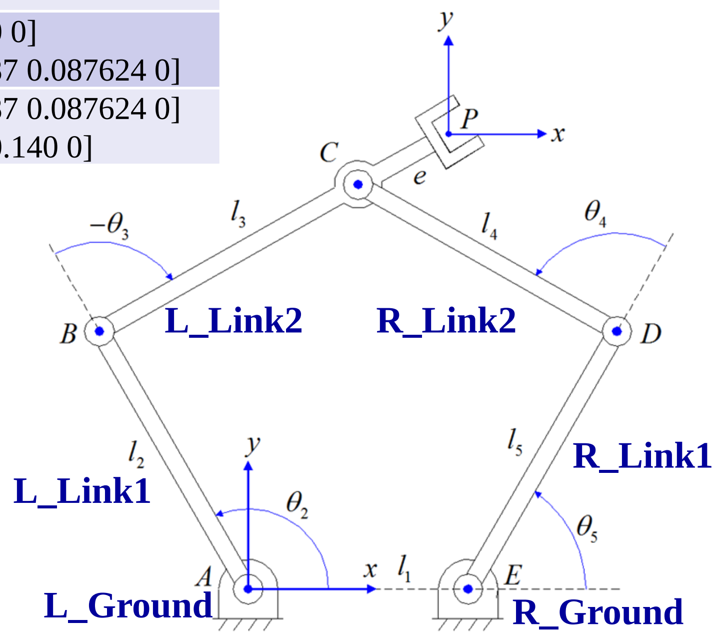




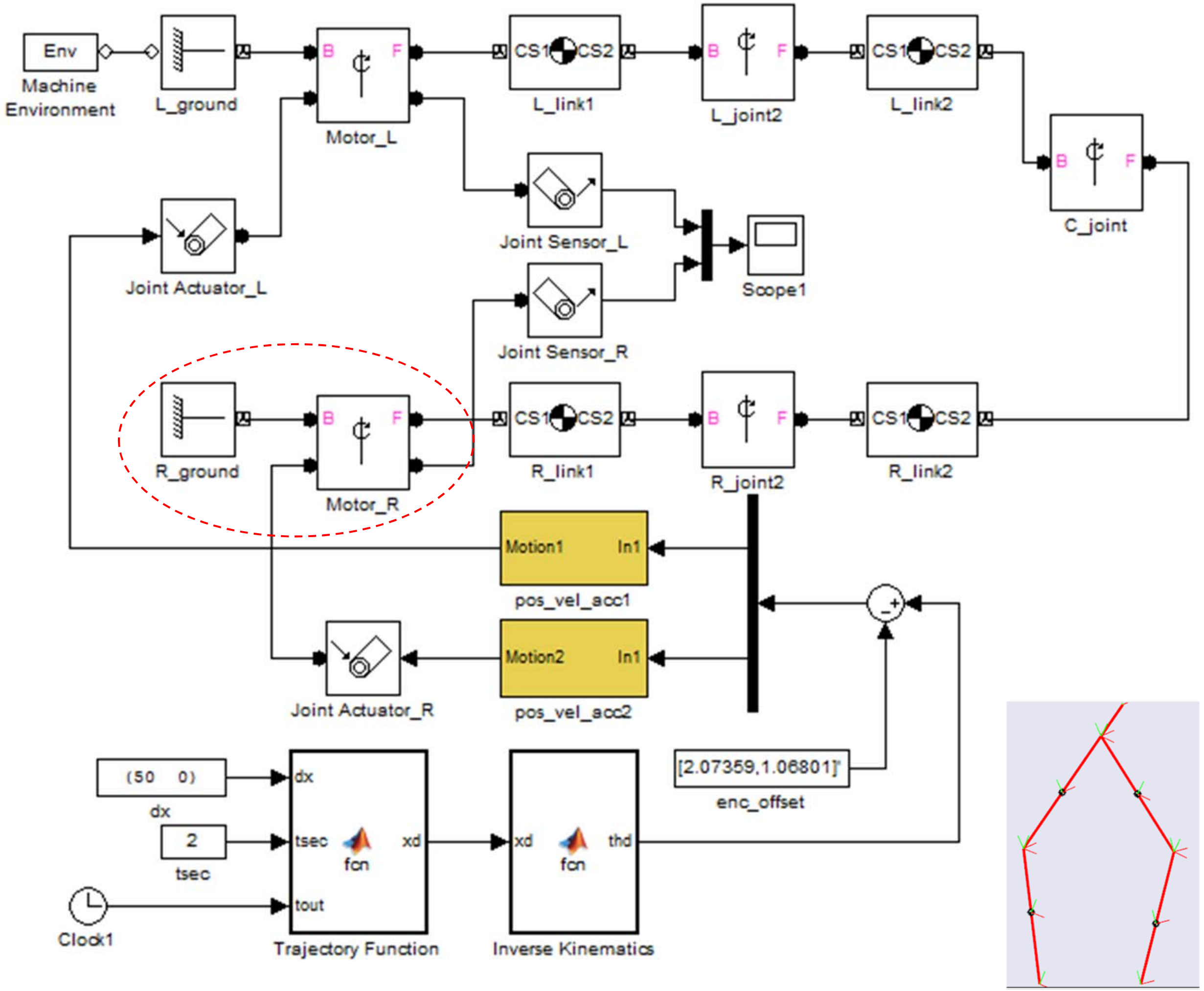
Inverse Dynamics Simulation (5-bar mechanism)

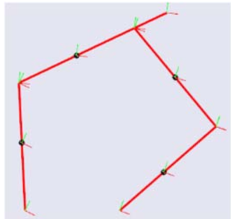
- **Cartesian-space Simulation: Δp_x=50, Δp_y=50,**

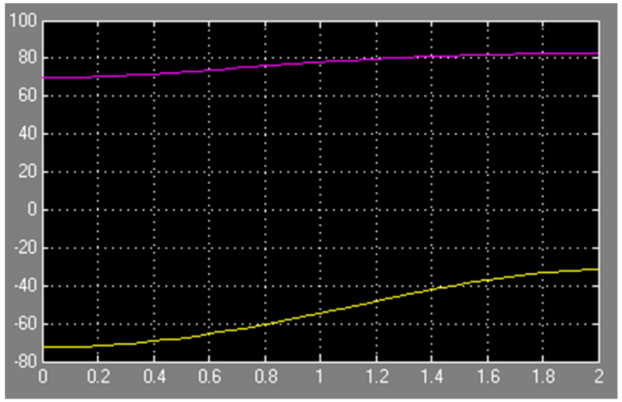
	Mass(kg)	Inertia (kg mm²)	CG Position(m) /from CS1	CS(m) /from World
L_Ground	-	-	-	[0 0 0]
L_Link1	0.053	110.61	[-0.025592 0.046537 0]	CS1 = [0 0 0] CS2 = [-0.048187 0.087624 0]
L_Link2	0.054	120.188	[0.042537 0.026153 0]	CS1 = [-0.048187 0.087624 0] CS2 = [0.037 0.140 0] CS3 = [0.060852 0.154665 0]
R_Ground	-	-	-	[0.074 0 0]
R_Link1	0.061	134.303	[0.022017 0.040035 0]	CS1 = [0.074 0 0] CS2= [0.122187 0.087624 0]
R_Link2	0.045	90.992	[-0.042594 0.026188 0]	CS1= [0.122187 0.087624 0] CS2 = [0.037 0.140 0]



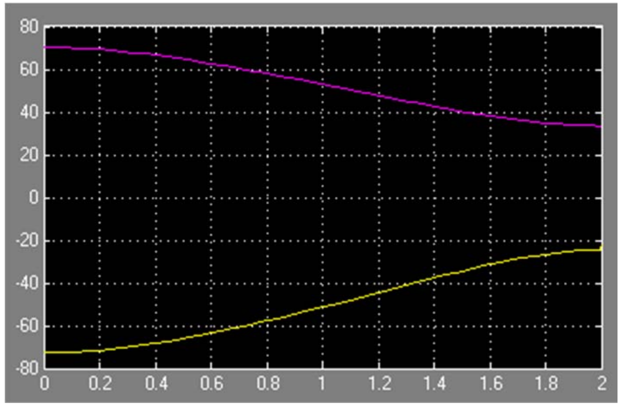
Inverse Dynamics Simulation (5-bar mechanism, Type I)





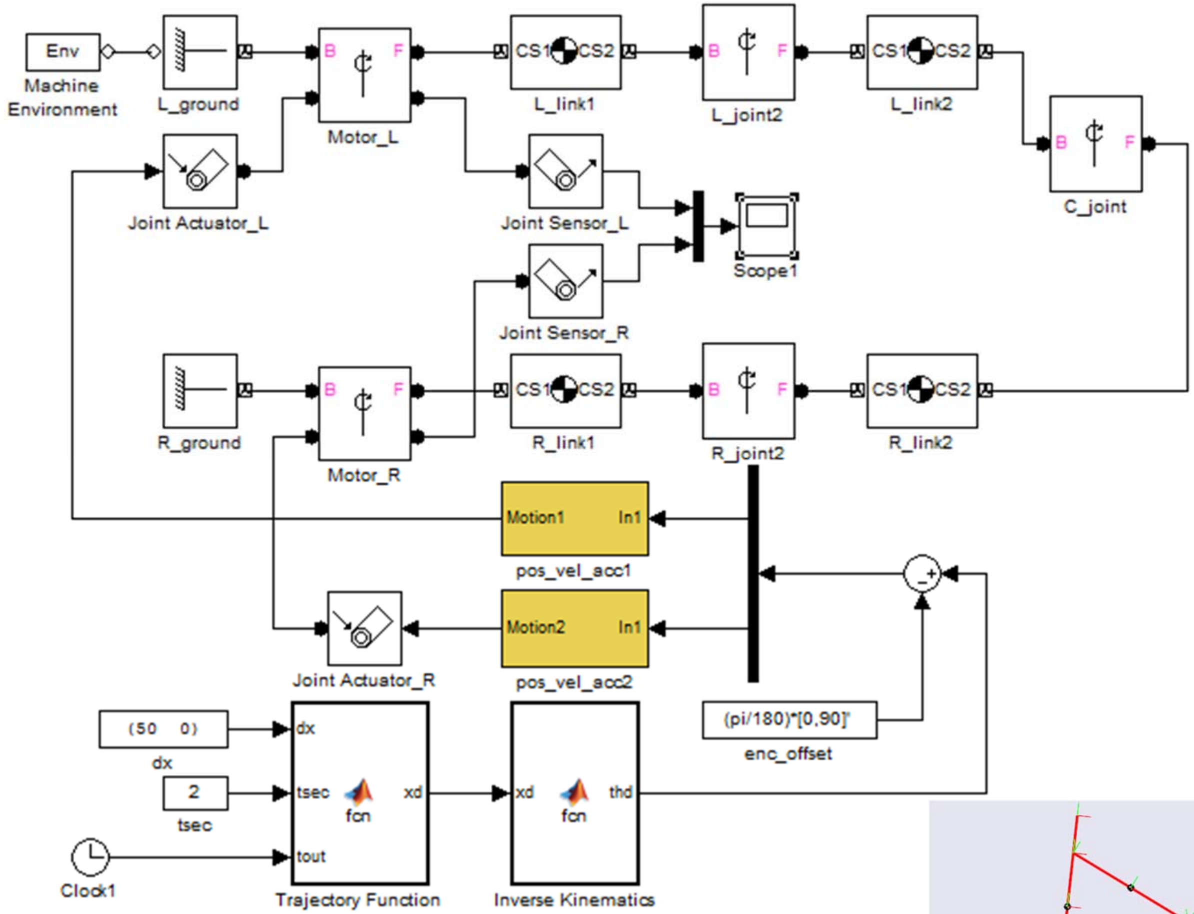


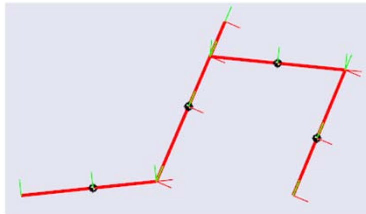
[Δp_x move]

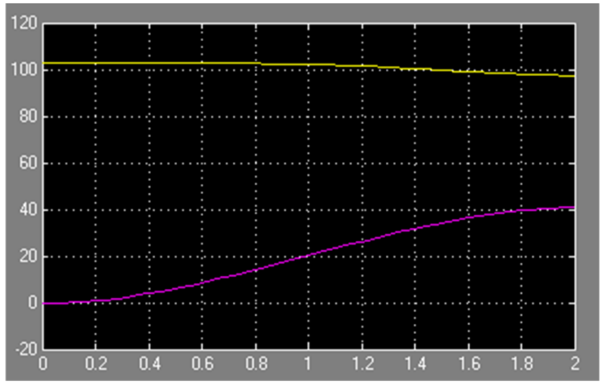


[Δp_y move]

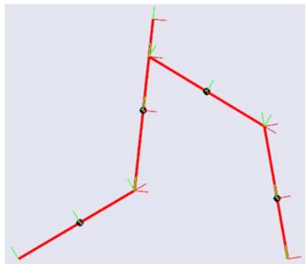
Inverse Dynamics Simulation (5-bar mechanism, Type II)

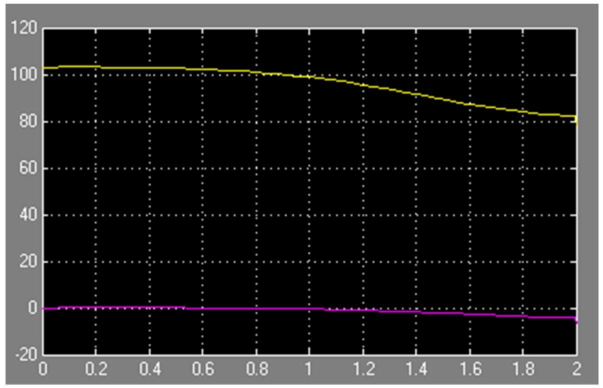






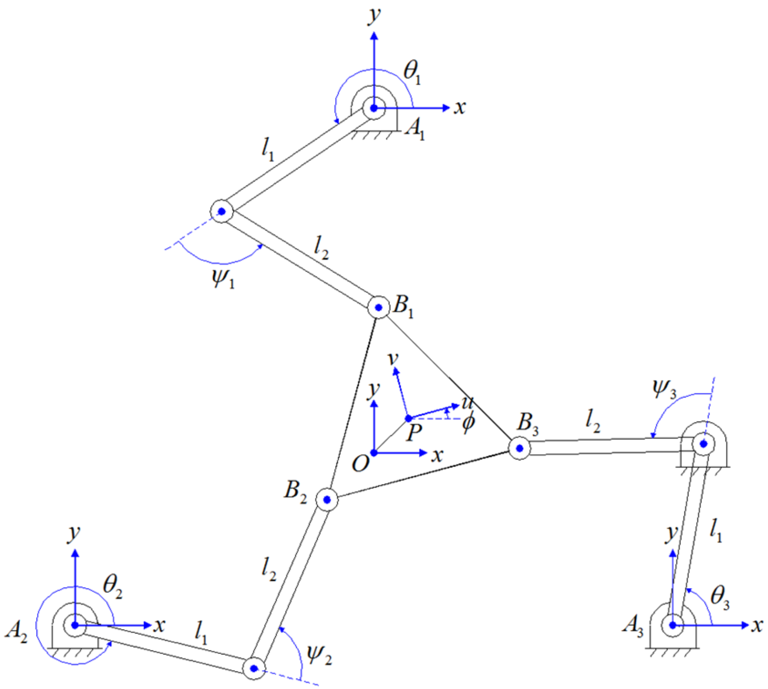
[Δp_x move]





[Δp_y move]

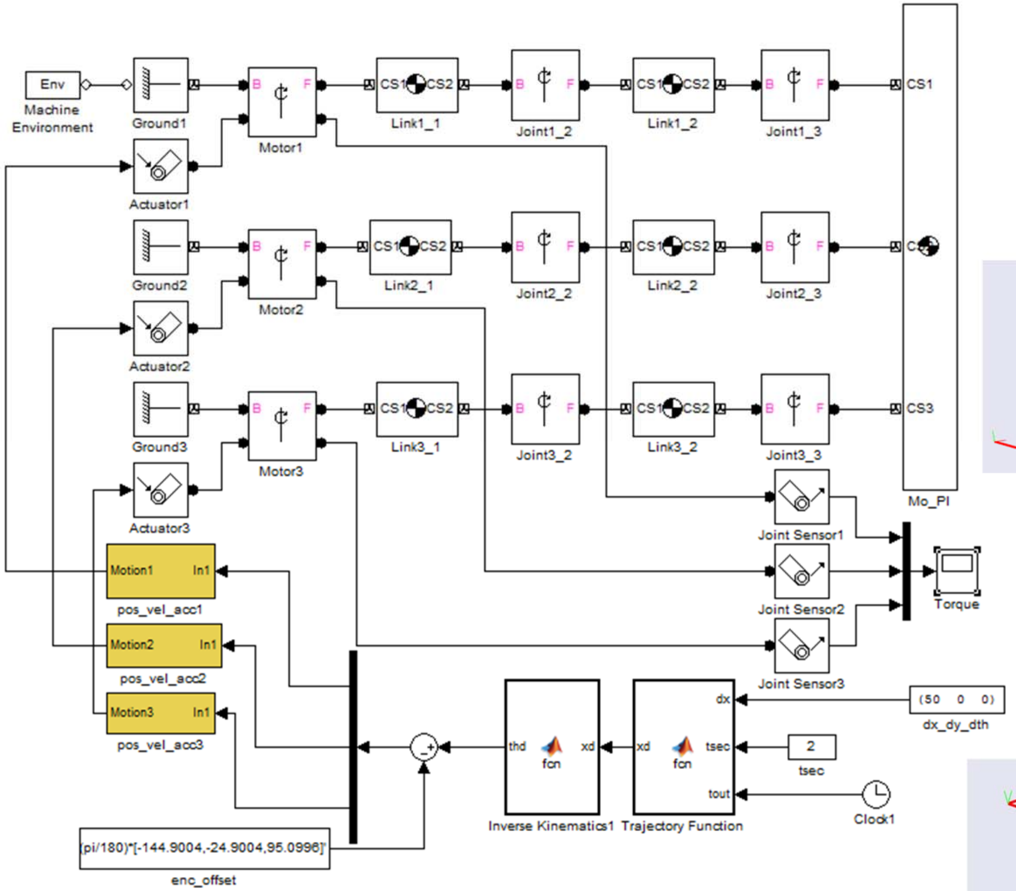
Inverse Dynamics Simulation (3-DOF Parallel Robot)

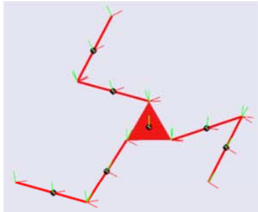


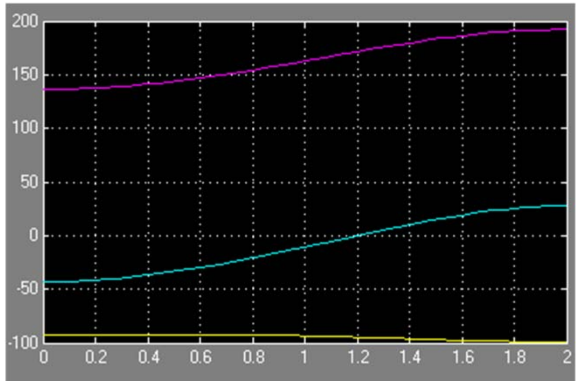
	Mass (kg)	Inertia (kg mm2)	CG Position(m) /from CS1	CS(m) /from World
Ground1	-	-	-	[0 0.15 0]
Link1_1	0.053	110.621	[-0.043452 -0.030538 0]	CS1=[0 0.15 0] CS2=[-0.081818 0.092497 0]
Link1_2	0.045	90992	[0.040909 -0.028749 0]	CS1=[-0.081818 0.092497 0] CS2=[0 0.035 0]
Ground2	-	-	-	[-0.1299 -0.075 0]
Link2_1	0.053	110.621	[0.048173 -0.022361 0]	CS1=[-0.1299 -0.075 0] CS2=[-0.039196 -0.117104 0]
Link2_2	0.045	90.992	[0.0044425 0.049802 0]	CS1=[-0.039196 -0.117104 0] CS2=[-0.03011 -0.0175 0]
Ground3	-	-	-	[0.129902 -0.074997 0]
Link3_1	0.053	110.621	[-0.0047204 0.0528997 0]	CS1=[0.129902 -0.074997 0] CS2=[0.121013 0.024607 0]
Link3_2	0.045	90.992	[-0.0453515 -0.0210535 0	CS1=[0.121013 0.024607 0] CS2=[0.030311 -0.0175 0]
Mo-Pl	0.081	46.643	[0 0 0] /from World	CS1=[0 0.035 0] CS2=[-0.03011 -0.0175 0] CS3=[0.030311 -0.0175 0]

Inverse Dynamics Simulation (3-DOF Parallel Robot)

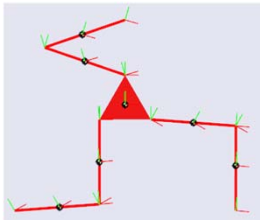
◆ **Cartesian-space Simulation: Δp_x=50, Δp_y=50,**

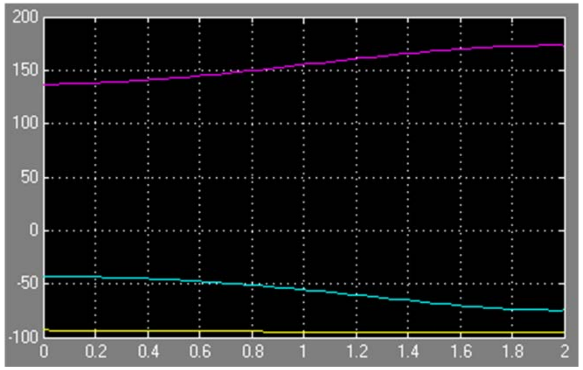






[Δp_x move]





[Δp_y move]

[4] Robot Control Practice

- **Introduction to xPC Target & speedgoat controller**
- **Design of speedgoat Controller**
- **Robot Control Practice**
  - ✓ Hardware Test
  - ✓ Control of DC servo motors (In Joint-space)
  - ✓ Simple PTP Control (In Cartesian-space)
  - ✓ Trajectory Control (In Cartesian-space)
  - ✓ Gravity Compensation (In Cartesian-space)

Introduction to xPC Target

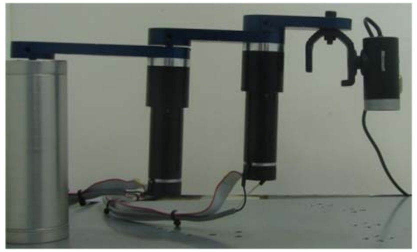
[Host PC]
Windows
Non Real-time

[Target PC]
xPC Target
Real-time Kernel

[Hardware]
Robots, Mechatronics

TCP/IP

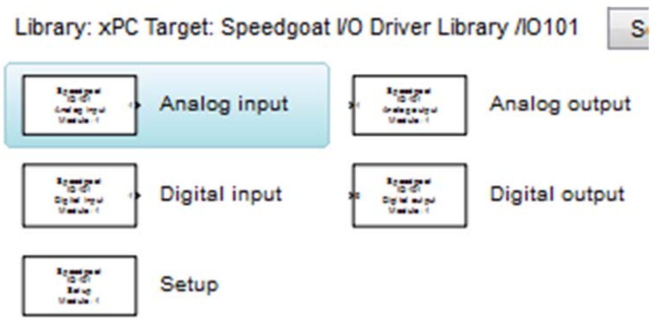
I/O

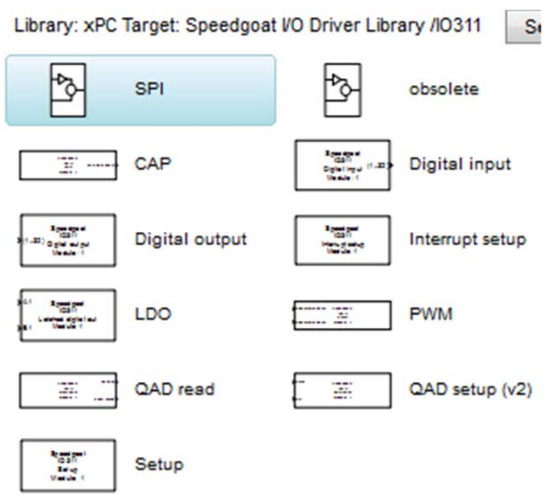


Simulink Programming → **Real-time Workshop (Simulink Coder)** → C-code Generation → **C-Compiler** → .dlm executable on xPC Target

Introduction to speedgoat Controller

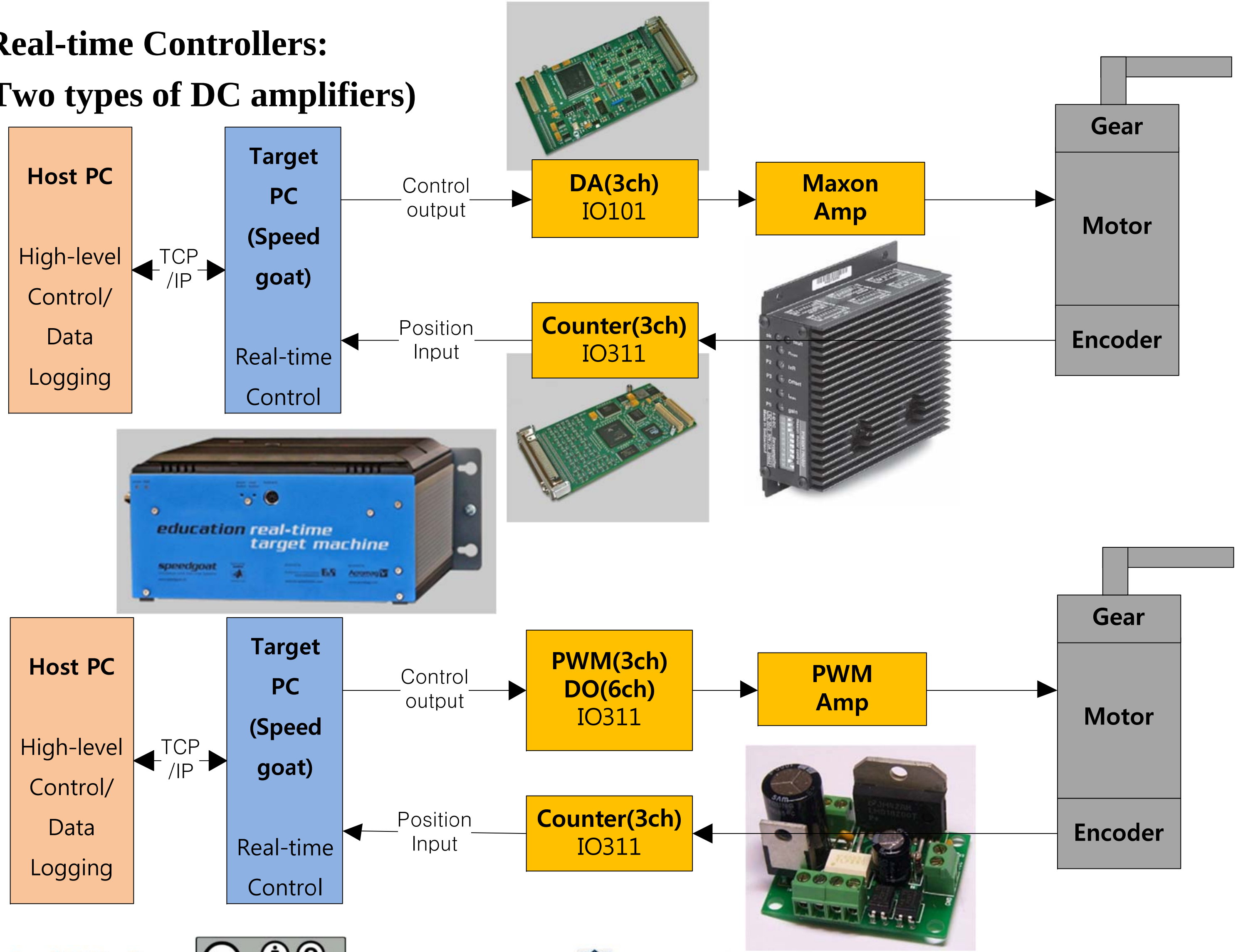
- **speedgoat Education real-time target machine**
  - ✓ Intel Celeron M 1GHz, 1GB IDE flash drive, 24V DC
  - ✓ I/O Slots: 2 (IO101 & IO311)
  - ✓ IO101 (fixed DAQ package): 16-ch AI (16 bit, differential), 8-ch AO, 8-ch DI, 8-ch DO
  - ✓ IO311 (Motion Control FPGA package): 64 TTL I/O, 24k logic cells, fixed set of 3-ch PWM, 3-ch Encoder, DIO
  - ✓ Supported xPC target drivers



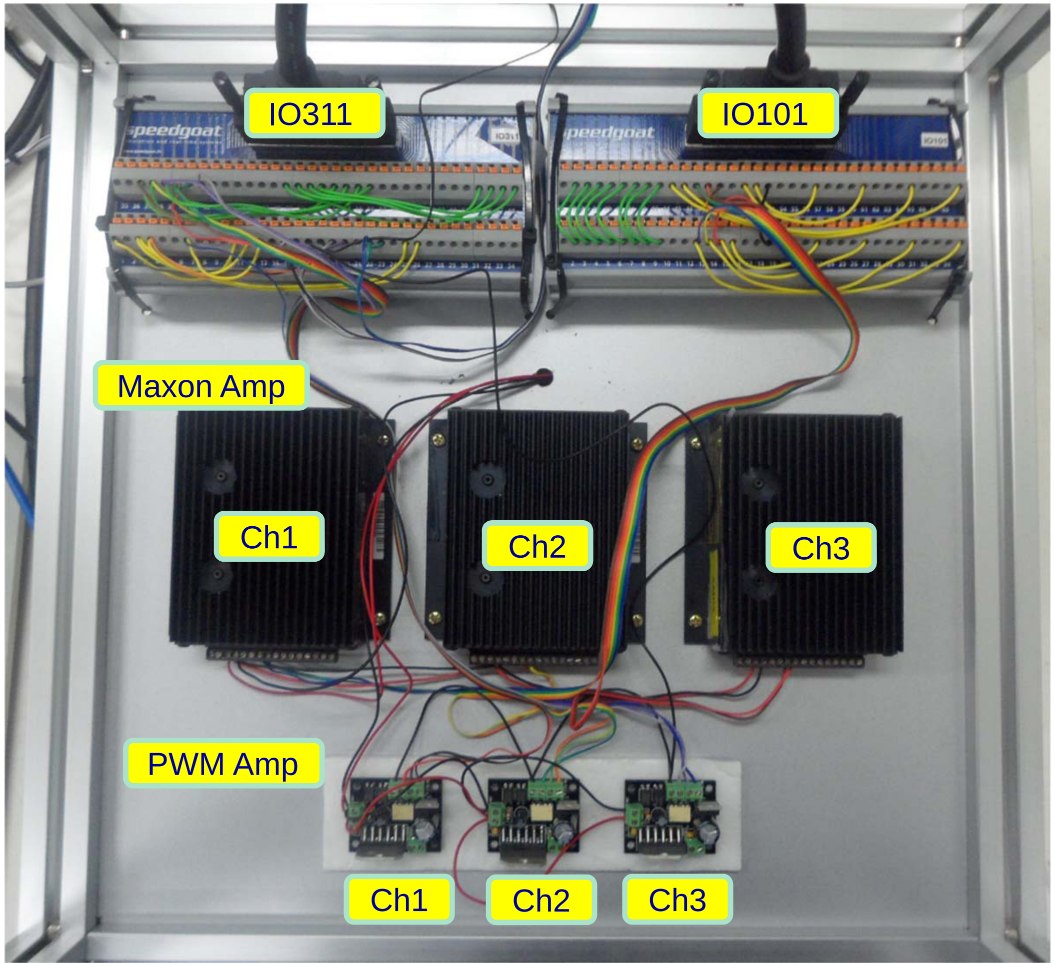


Design of speedgoat Controller

- **Real-time Controllers:**
  (Two types of DC amplifiers)

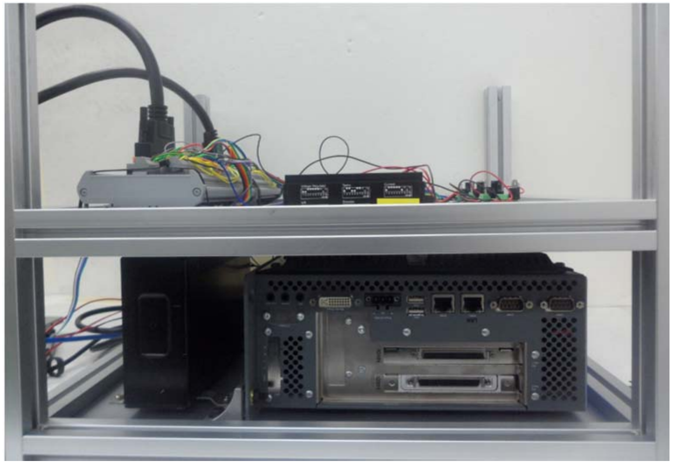


Design of speedgoat Controller



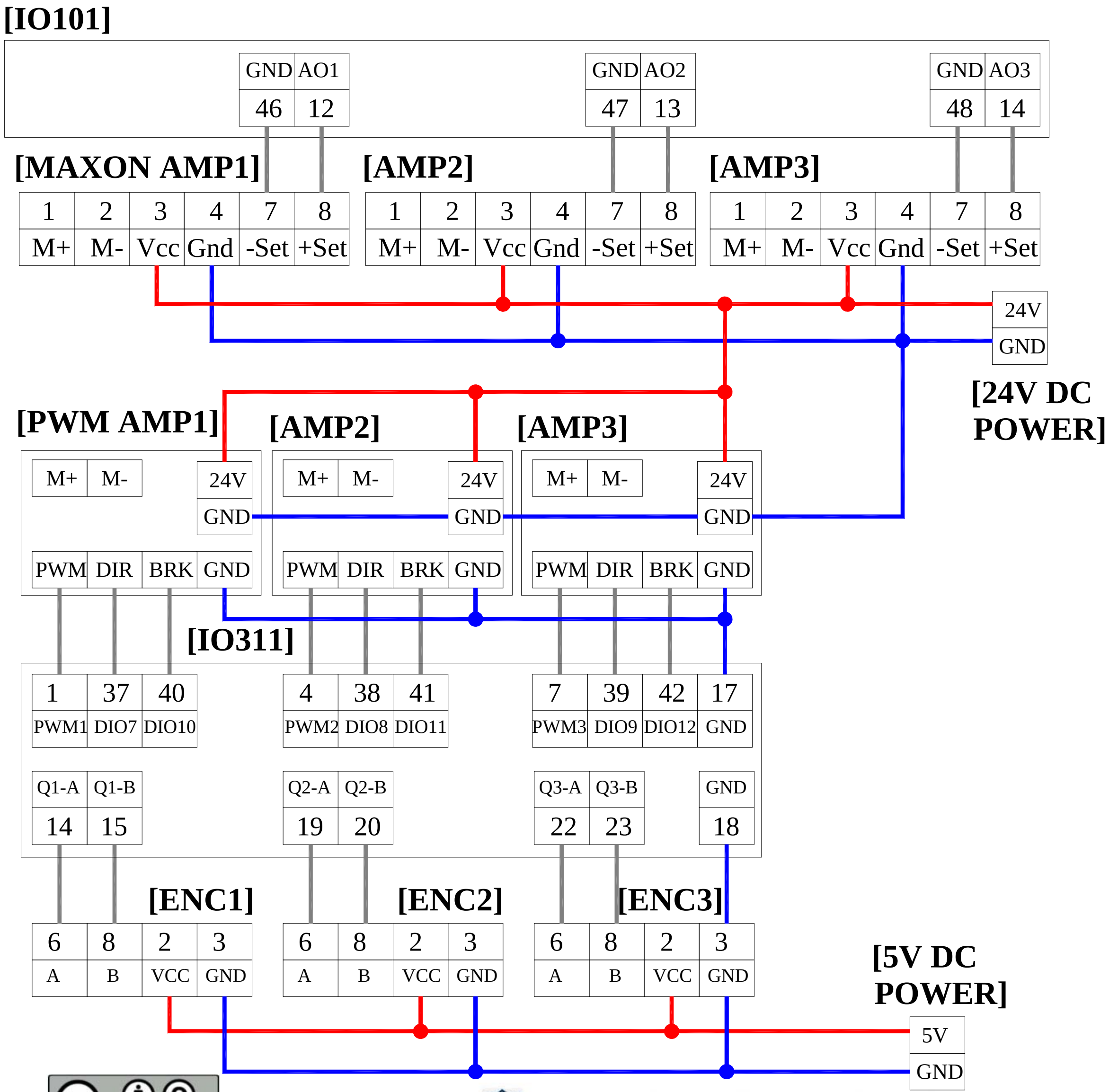
[Top view]

[Front view]



[Side view]

Design of speedgoat Controller



Design of speedgoat Controller

IO101 (Analog I/O Module, 8ch-DA, 16ch-AD, 8ch-DIO):

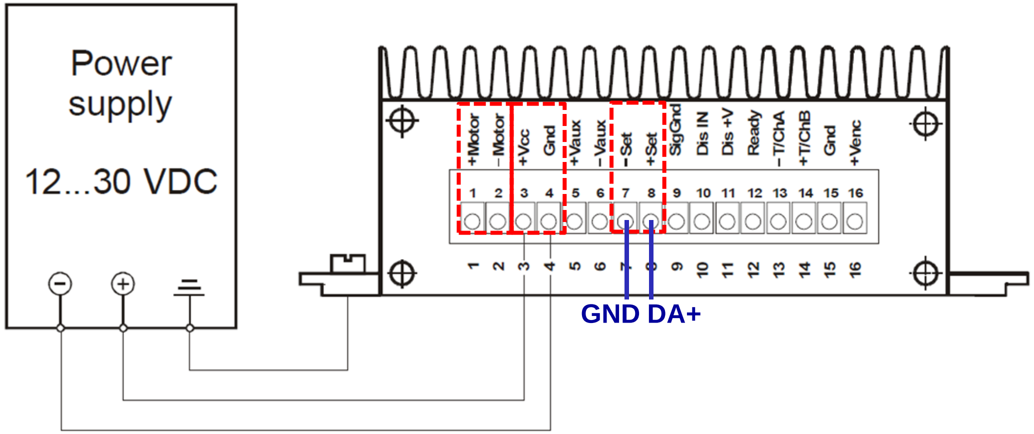
PIN	Signal	Amplifier	PIN	Signal	Amplifier
12	AO01	+set (ch1)	46	GND	-set (ch1)
13	AO02	+set (ch2)	47	GND	-set (ch2)
14	AO03	+set (ch3)	48	GND	-set (ch3)

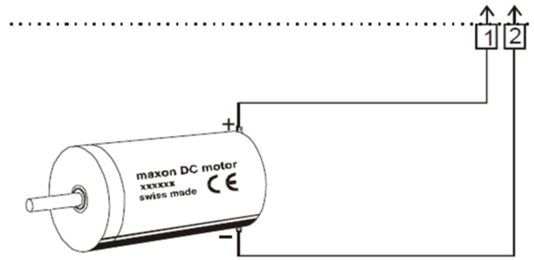
IO311 (Configurable FPGA I/O modules, Xilinx Virtex-II 24k):

PIN	Signal	Amp	PIN	Signal	Encoder	PIN	Signal	Amp
1	PWM1-A	PWM (ch1)	14	QAD1-A	A+ (ch1)	37	DIO07	DIR (ch1)
4	PWM2-A	PWM (ch2)	15	QAD1-B	B+ (ch1)	38	DIO08	DIR (ch2)
7	PWM3-A	PWM (ch3)	19	QAD2-A	A+ (ch2)	39	DIO09	DIR (ch3)
17	GND	GND (AMP)	20	QAD2-B	B+ (ch2)	40	DIO10	BRK (ch1)
18	GND	GND (ENC)	22	QAD3-A	A+ (ch3)	41	DIO11	BRK (ch2)
			23	QAD3-B	B+ (ch3)	42	DIO12	BRK (ch3)

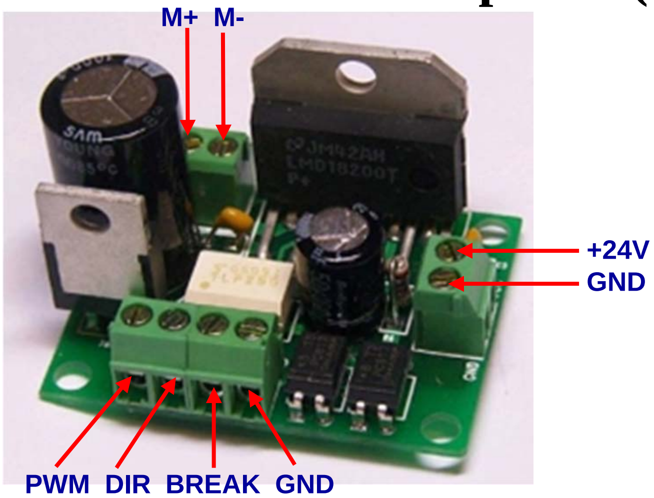
Design of speedgoat Controller

- **Maxon LSC 30/2 amplifier (24V, 2A):**





- **LMD18200 PWM amplifier (24V, 3A):**



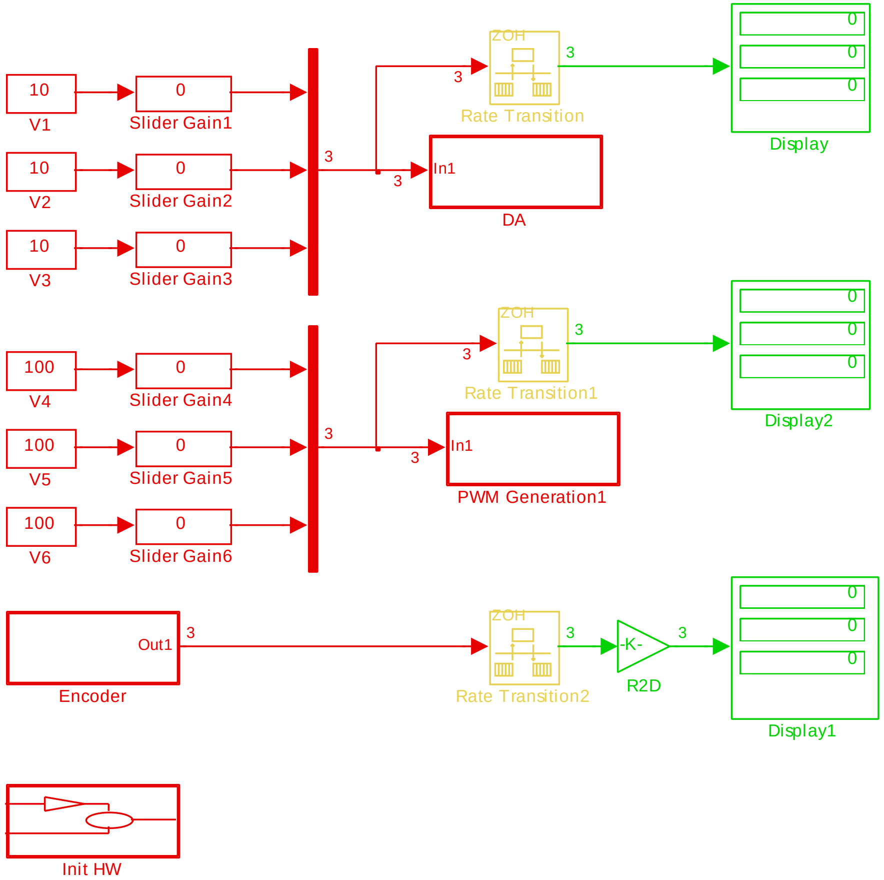
- **Maxon Motor (20W)**
- **66:1 Geared motor:**

  T_{max}=3.1 Nm
- **23:1 Geared motor**

  T_{max}=1.1 Nm
- **Encoder: 500 lines/rev**

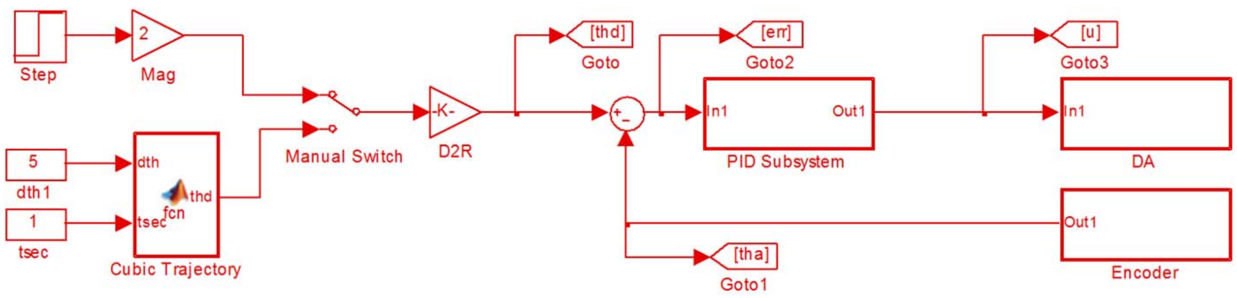
Robot Control Practice: Hardware Test

- **3-ch DA output, 3-ch PWM output, 6-ch Digital output, 3-ch Encoder input**
  - ✓ Sampling time: 1mse, external mode

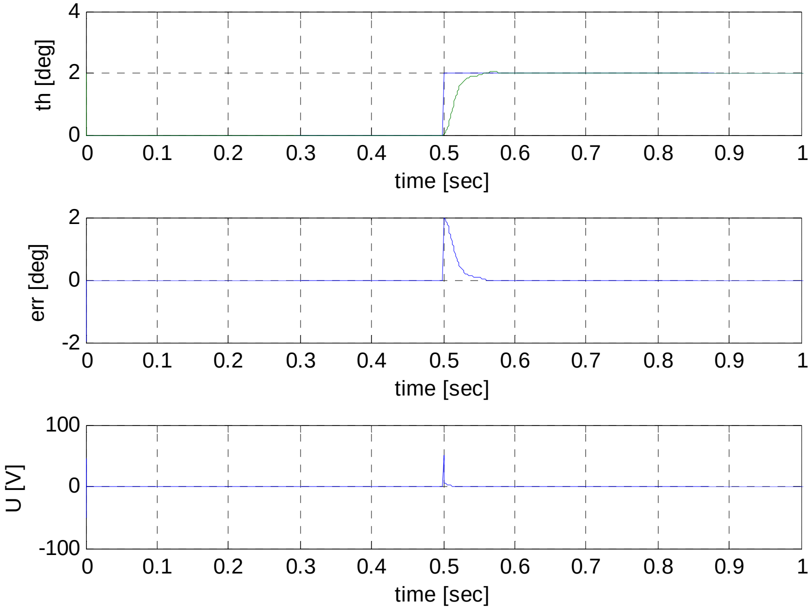


Robot Control Practice: DC motor control

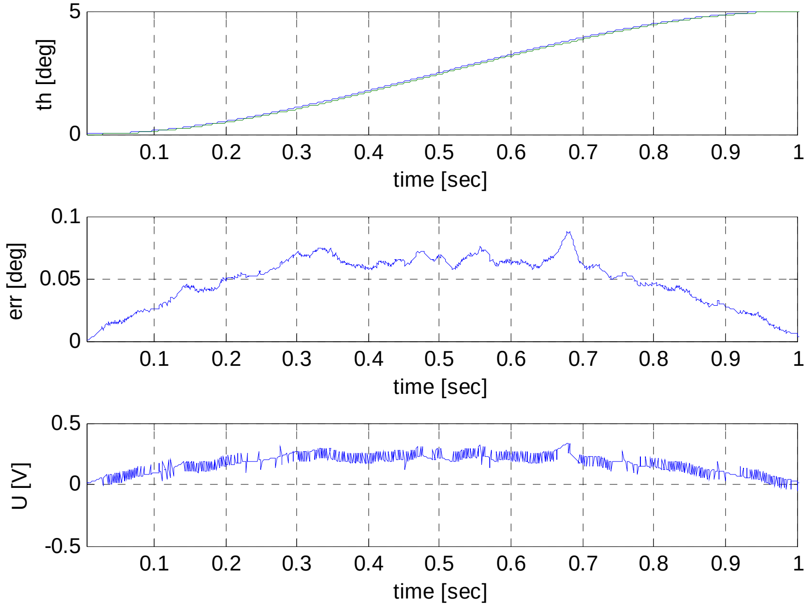
- **Control of DC motors for step and cubic trajectory inputs (Joint-space control)**



- **DC servo motor with 66:1 gear (k_p=150, k_d=1)**



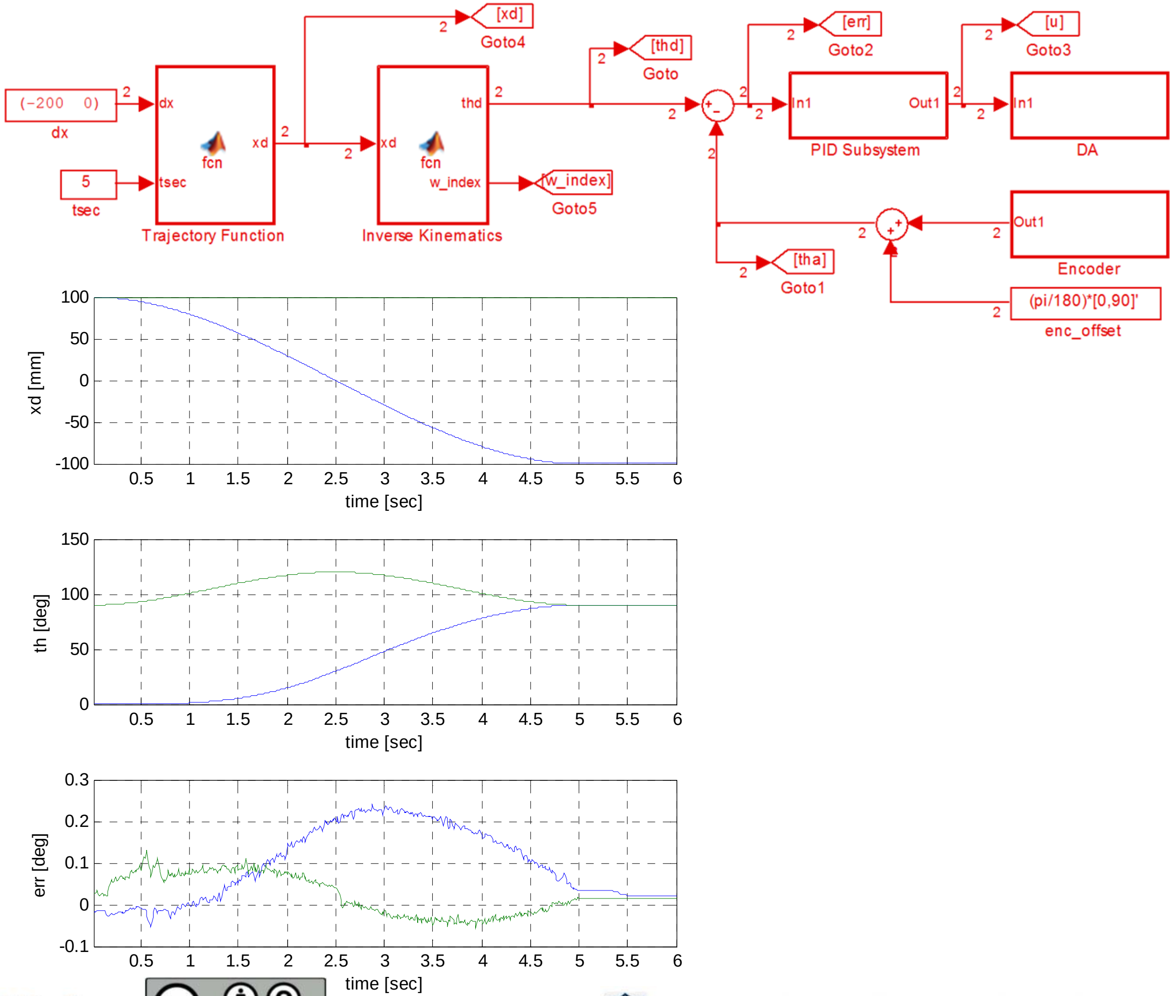
[step input]



[cubic trajectory input]

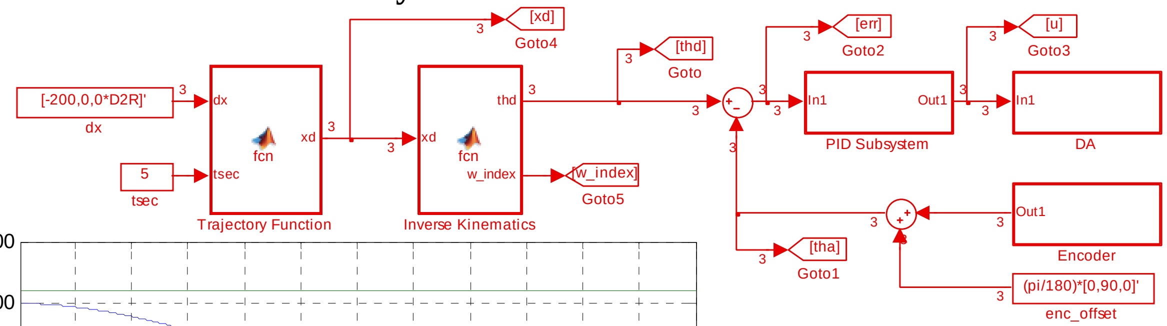
Robot Control Practice: PTP control

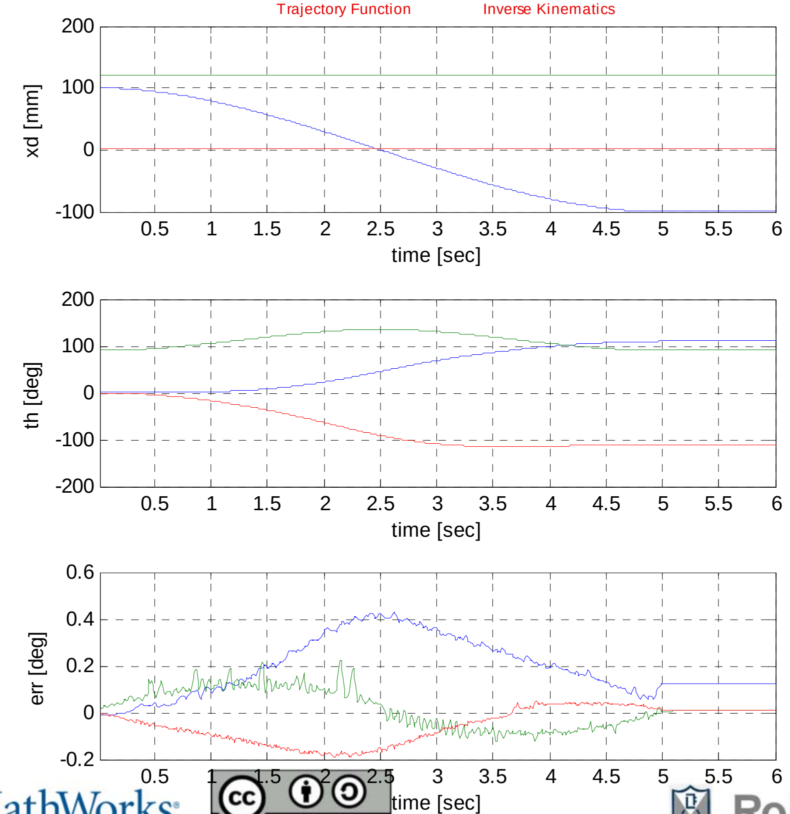
- **2-DOF Serial Robot: (M1) k_p=200, k_d=1.5, (M2) k_p=160, k_d=1.2**
- **Cubic Trajectory: $[p_x, p_y]$=[100,120] → [-100,120]**



Robot Control Practice: PTP control

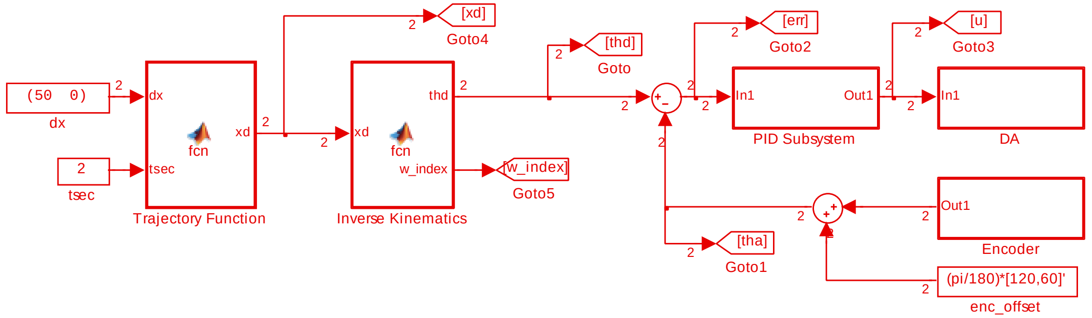
- 3-DOF Serial Robot: (M1) k_p=200, k_d=1.5, (M2,3) k_p=160, k_d=1.2
- Cubic Trajectory: [p_x, p_y]=[100,120], 90 deg → [-100,120], 90 deg

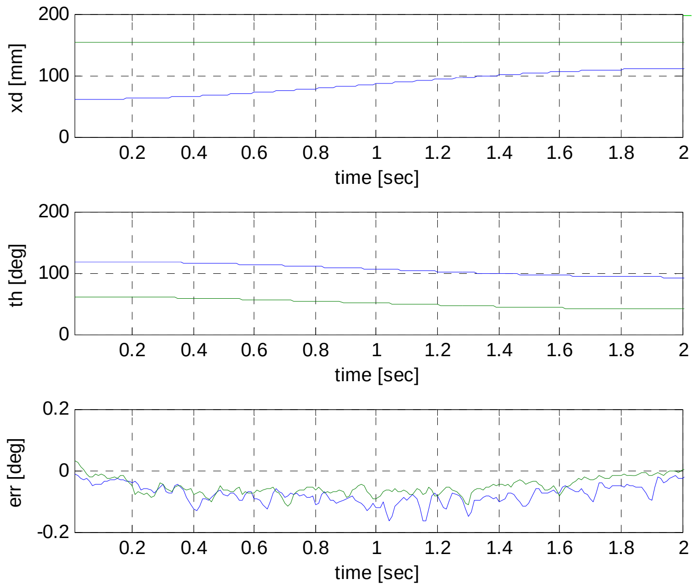




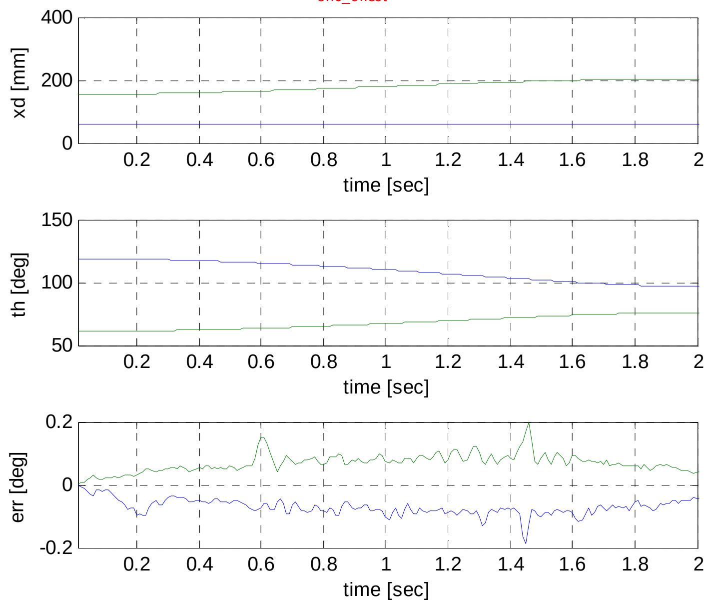
Robot Control Practice: PTP control

- **5-bar Mechanism (Type I) : (M1,2) k_p=160, k_d=1.2**
- **Cubic Trajectory: Δp_x=50, Δp_y=50**





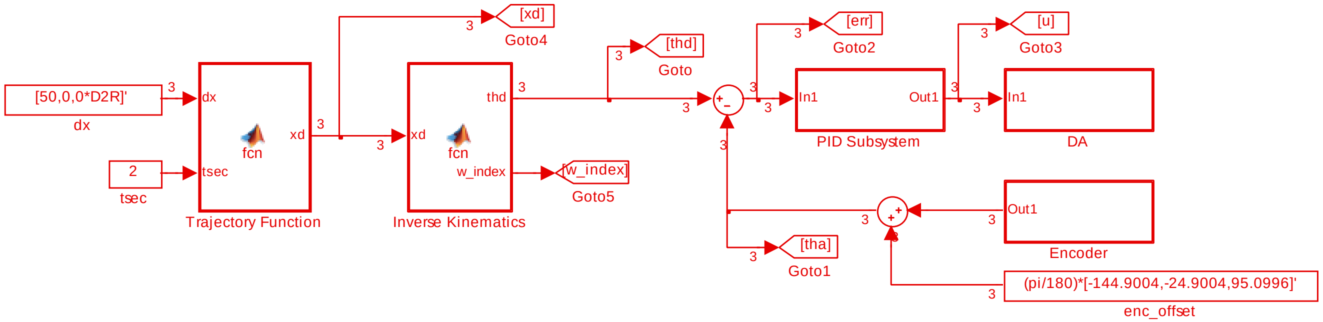
[X move]

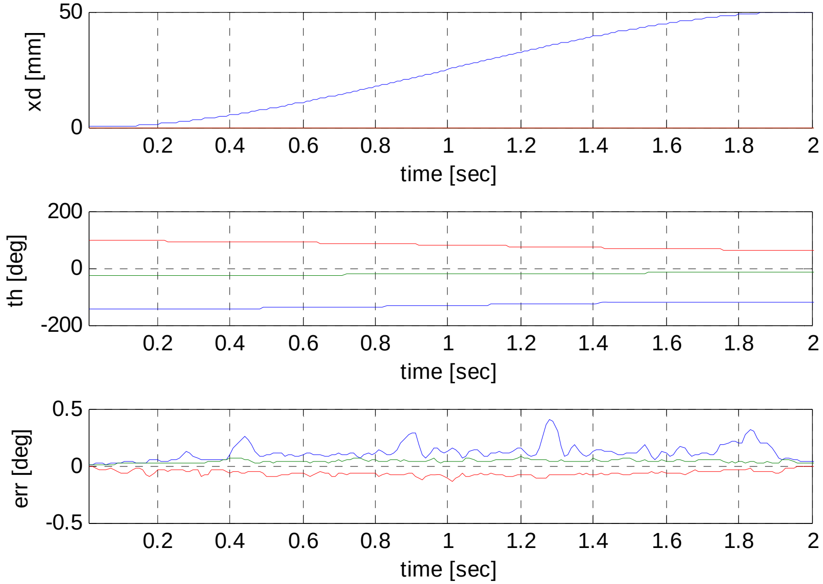


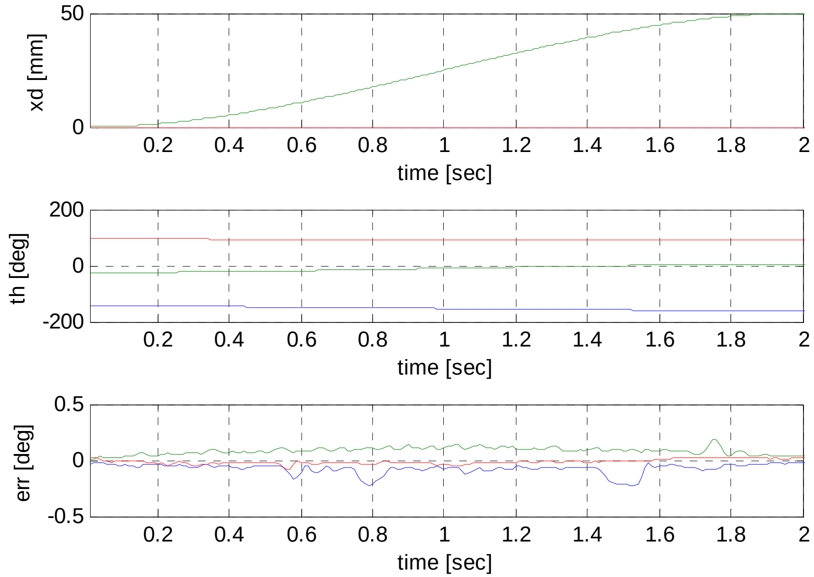
[Y move]

Robot Control Practice: PTP control

- **3-DOF Parallel Robot: (M1,2,3) k_p=160, k_d=1.2**
- **Cubic Trajectory: Δp_x=50, Δp_y=50**







[X move] **[Y move]**

Robot Control Practice: Trajectory

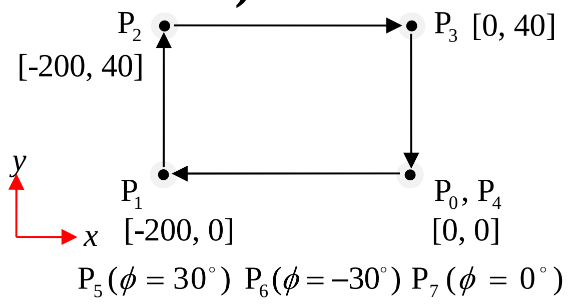
Trajectory Generation (similar to G-code)

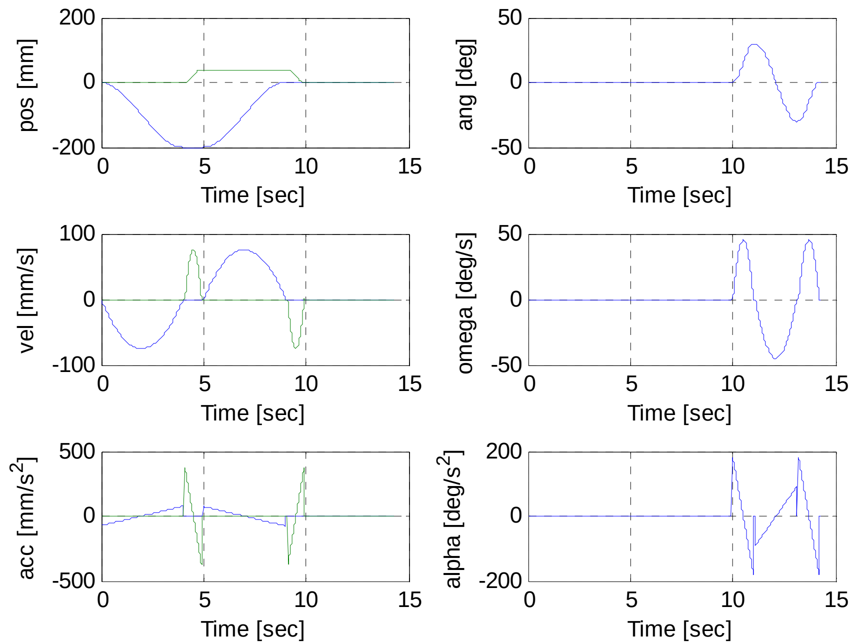
[Grammar]

% : notes
G01 : ptp move (by cubic polynomial)
Format: X Y W L A
[mm] [mm] [deg] [mm/s] [deg/s]
S00 T : set holding time in sec
M02 : script end

Format:

X	Y	W	L	A
[mm]	[mm]	[deg]	[mm/s]	[deg/s]

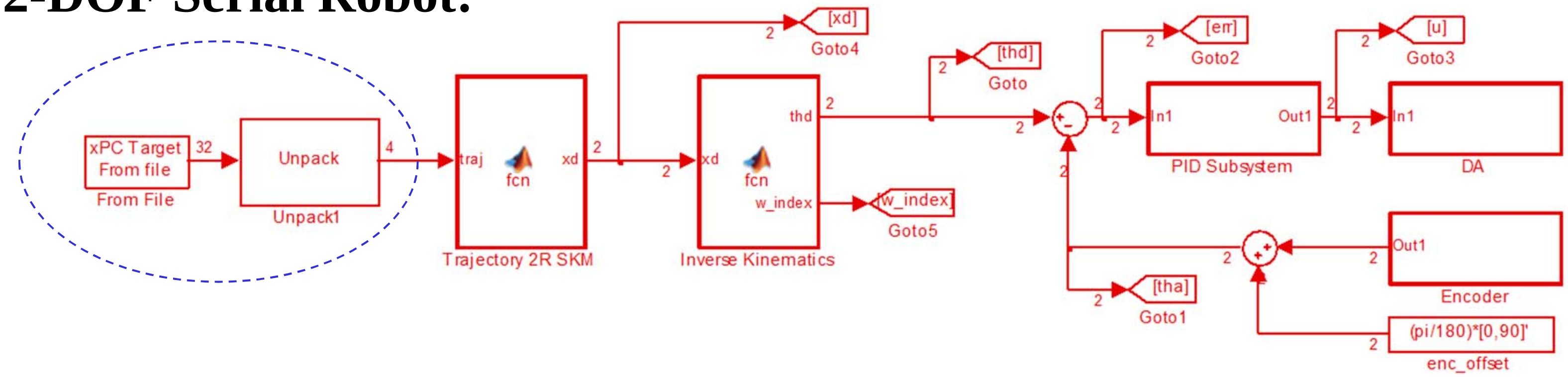


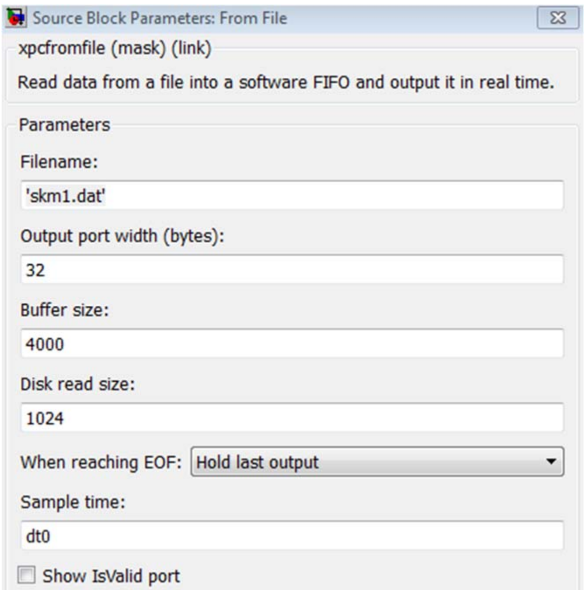



```
% 3-DOF SKM %
G01 X0 Y0 W0 L80 A30
% -X Move
G01 X-200 Y0 W0
S00 T0.1
% +Y Move
G01 X-200 Y40 W0
S00 T0.1
% +X Move
G01 X0 Y40 W0
S00 T0.1
% -Y Move
G01 X0 Y0 W0
S00 T0.1
% +ANG %
G01 X0 Y0 W30
S00 T0.1
% -ANG %
G01 X0 Y0 W-30
S00 T0.1
% HOME %
G01 X0 Y0 W0
S00 T0.1
M02
```

Robot Control Practice: Trajectory

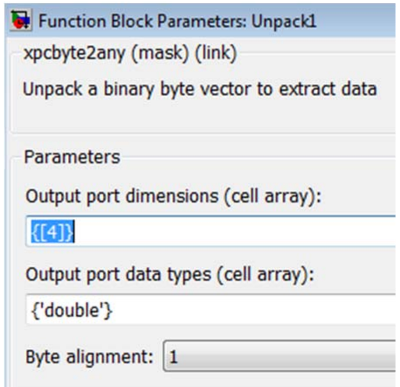
- **2-DOF Serial Robot:**

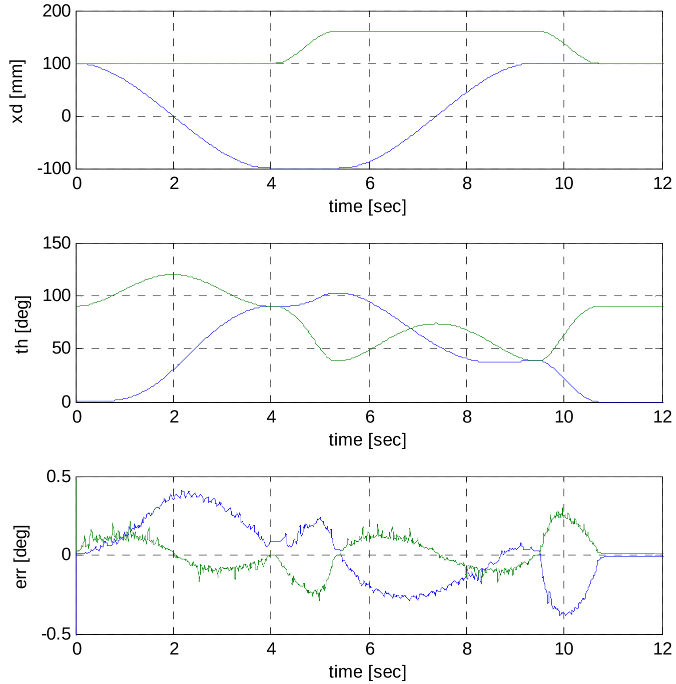




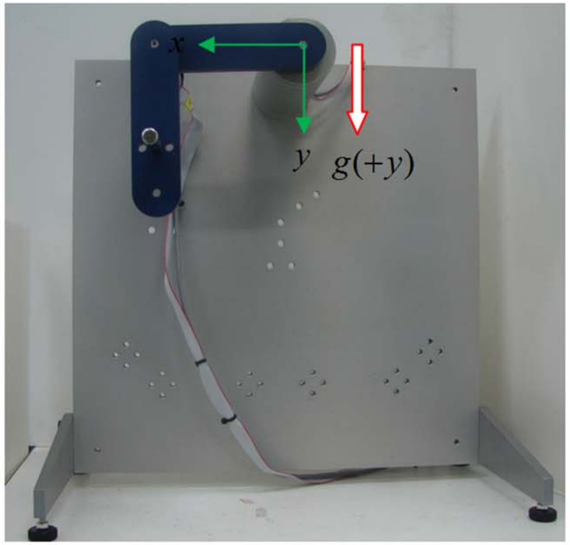
- Data output format:
$[p_x, p_y, \theta]$

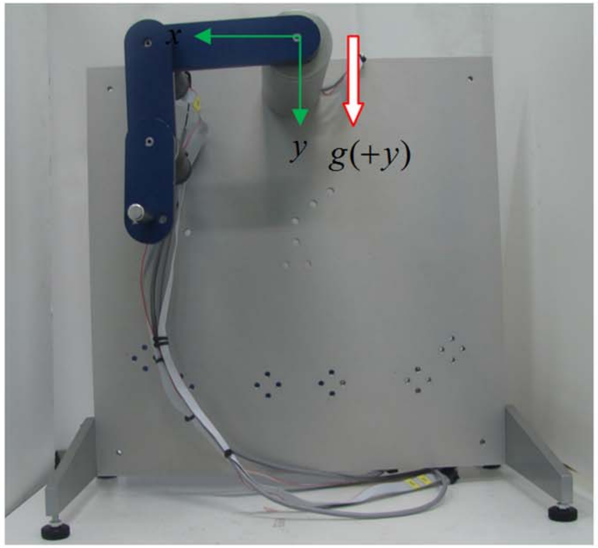
- Output port width:
$8 \times (\text{time}, p_x, p_y, \theta) = 32$

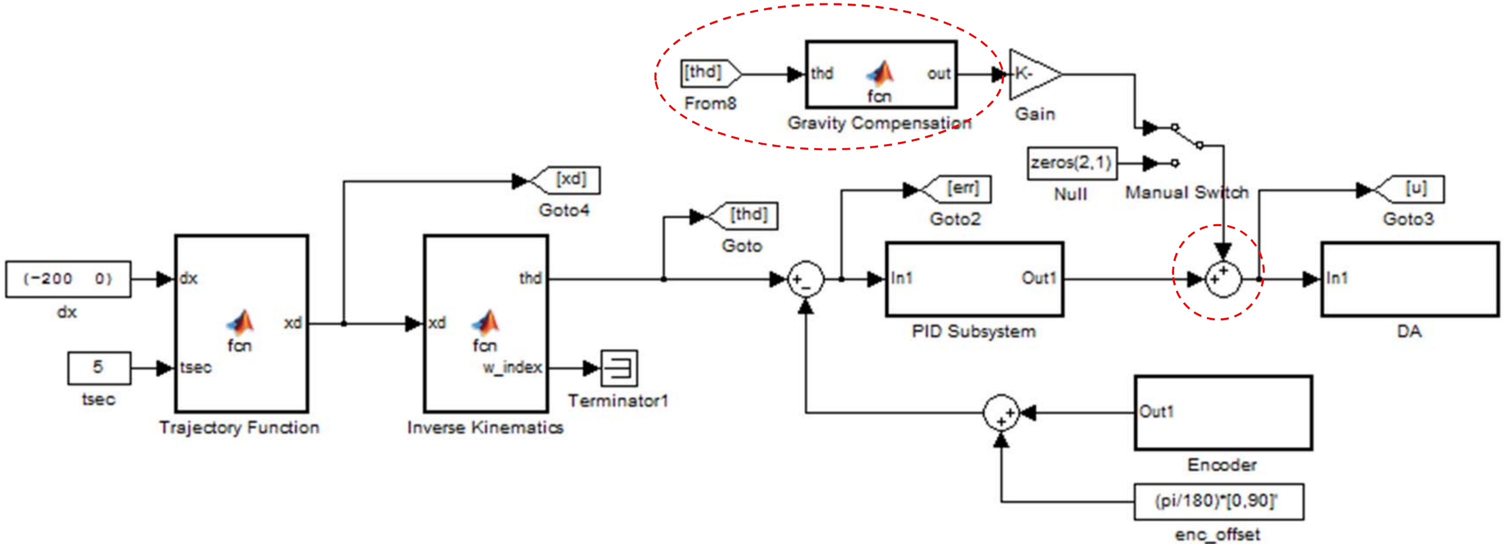




Robot Control Practice: Gravity Compensation



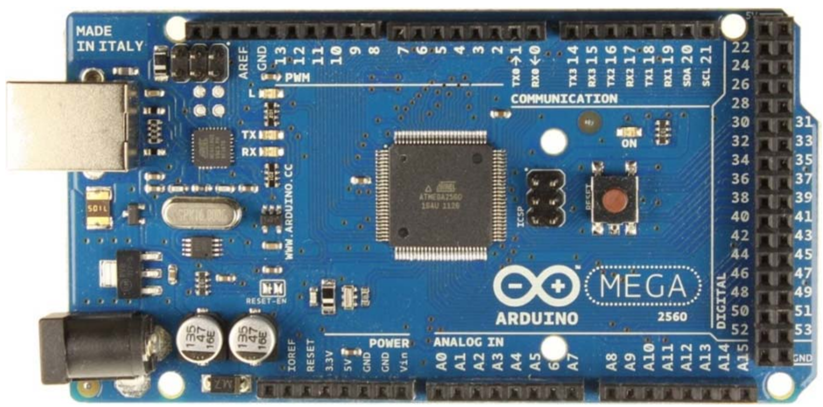


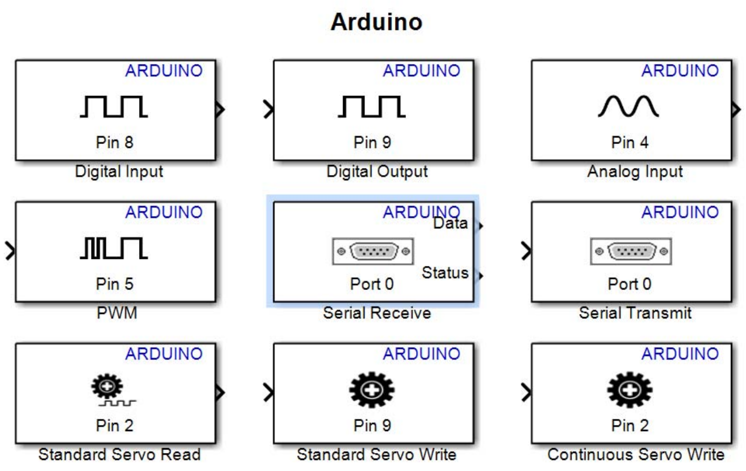


[5-1] Arduino Interface: Arduino Mega 2560

Arduino Mega 2560

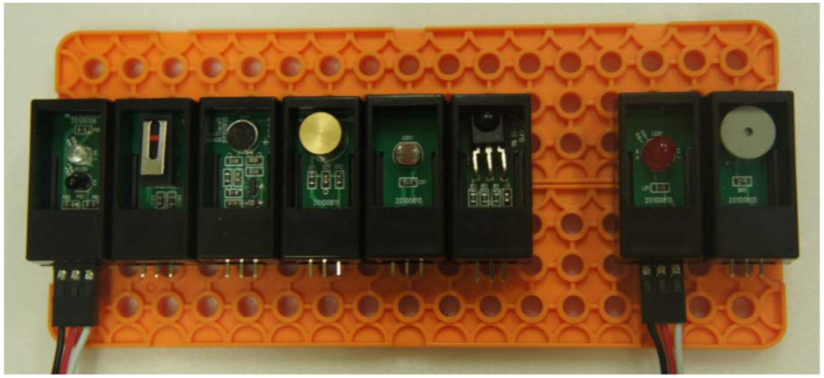
- ATmega 2560: 16MHz, 256kB Flash, 54-DIO, 14-PWM, 16-AI, 4-UART, USB connector
- Supported I/O drivers: AI, DI, DO, AI, PWM, Servo Read/Write, Serial Receive/Transmit
- MathWorks webpage: http://www.mathworks.co.kr/products/simulink/simulink-targets/

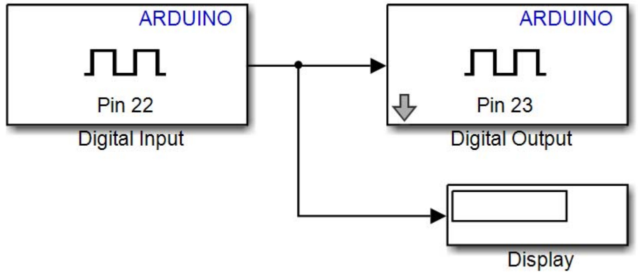




[5-2] Mechatronics Practice

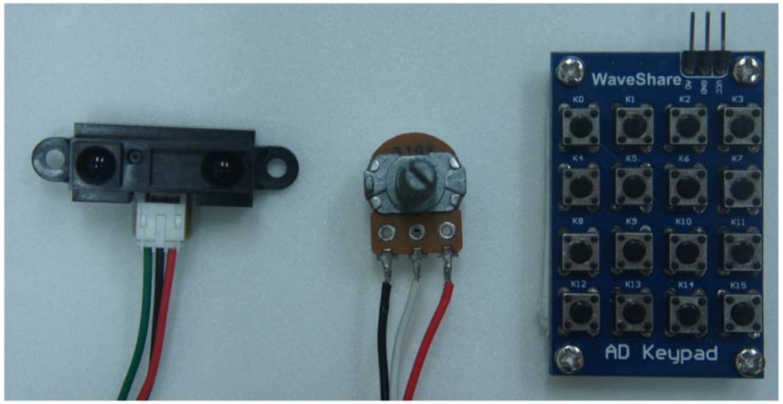
ON/OFF Type:



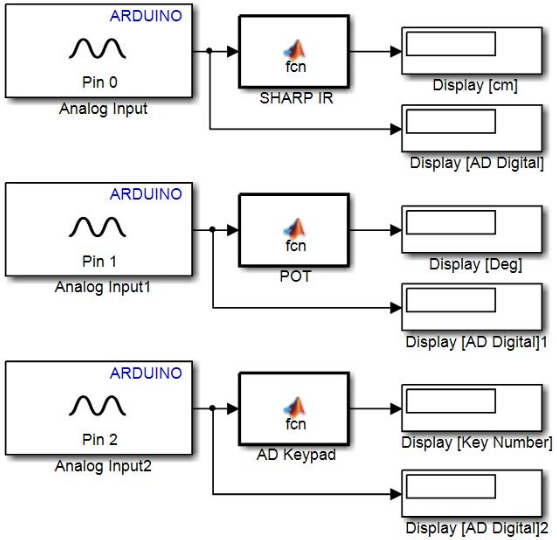


[Inputs: IR, Switch, Mic, Tilt CdS, Remote, Ouputs: LED, Buzzer]

Analog Sensor:



[IR, Pot, Keypad]

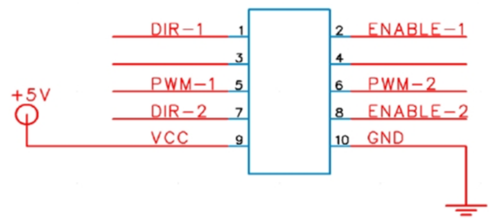


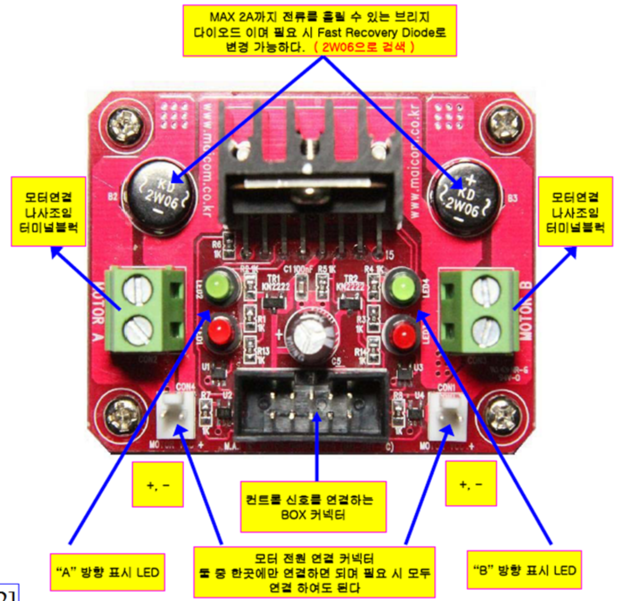
MOVIE

MOVIE

[5-2] Mechatronics Practice

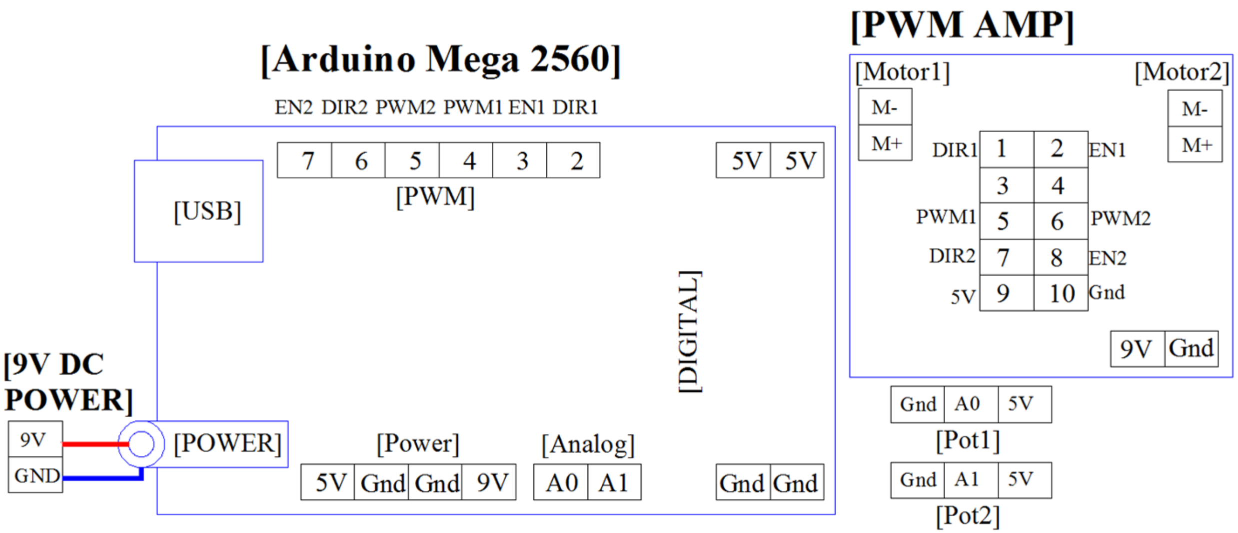
- **DC Motor:**





[Dynamixel (32:1, 254:1) → DC motor, Pot]

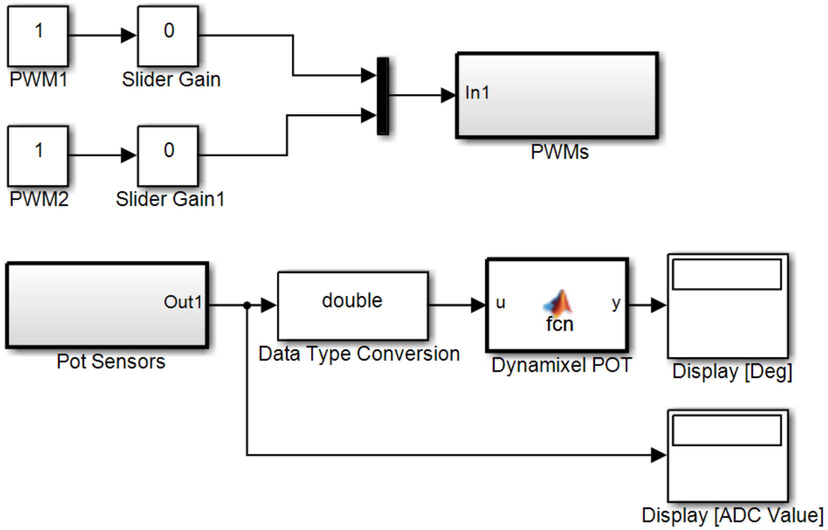
[PWM Amp]

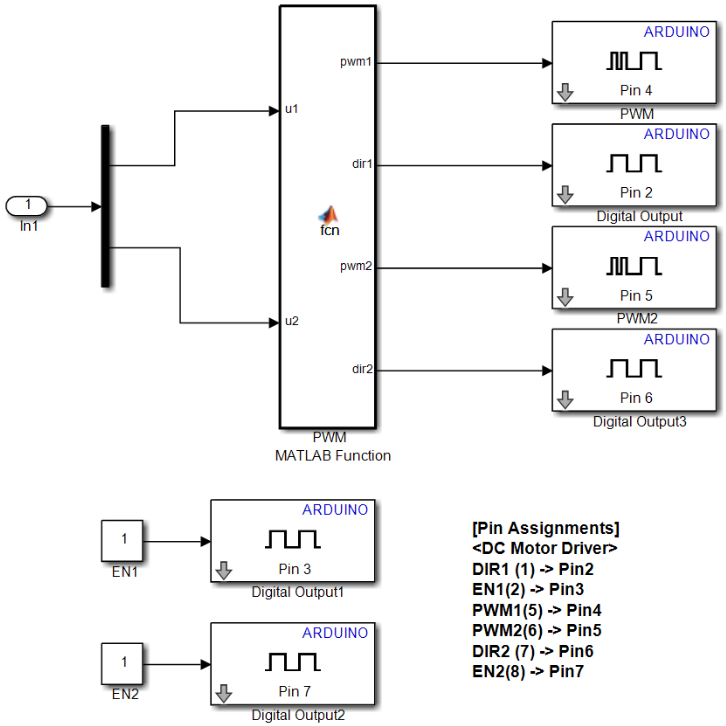


[Connection]

[5-2] Mechatronics Practice

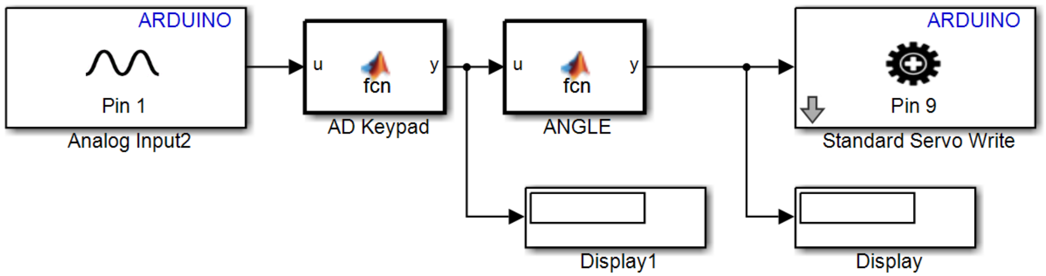
DC Motor:





[PWM Subsystem]

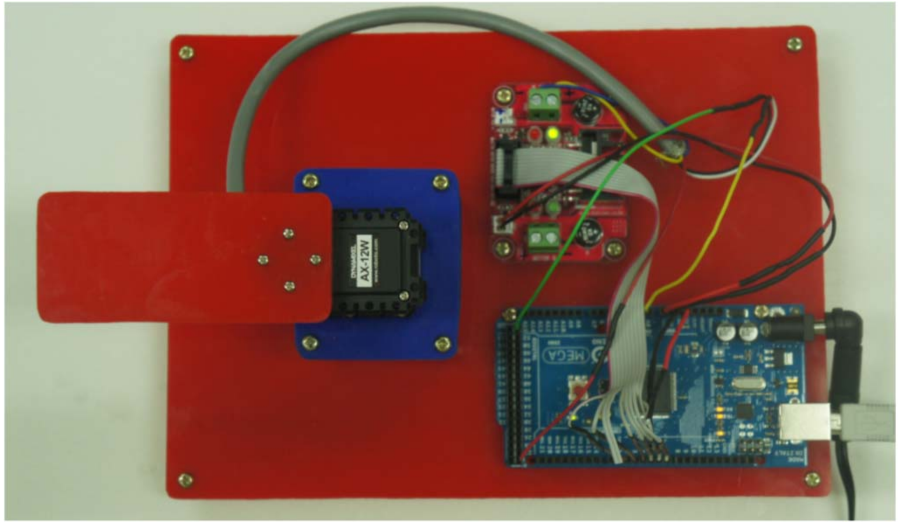
Servo Motor:

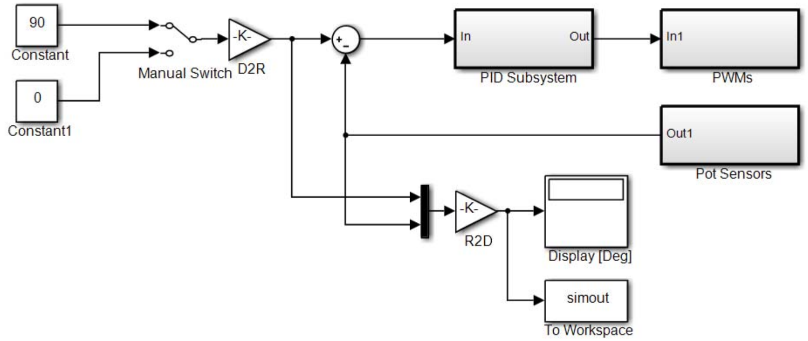


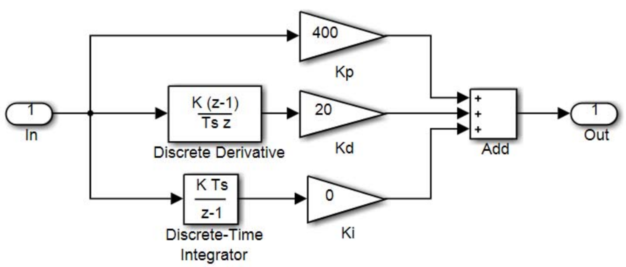
MOVIE

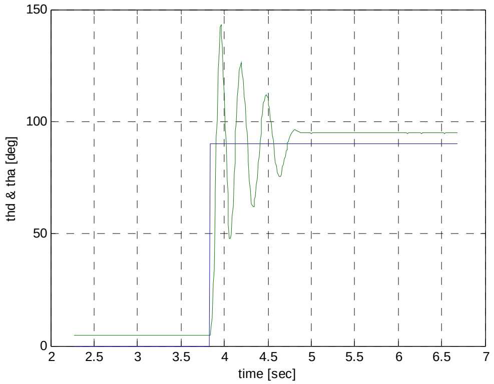
[5-3] PID Control Practice

◆ **PID Control (10msec):**

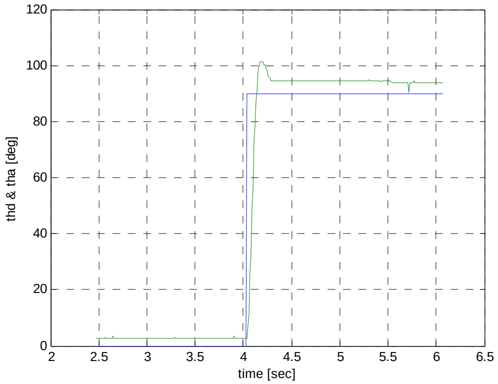




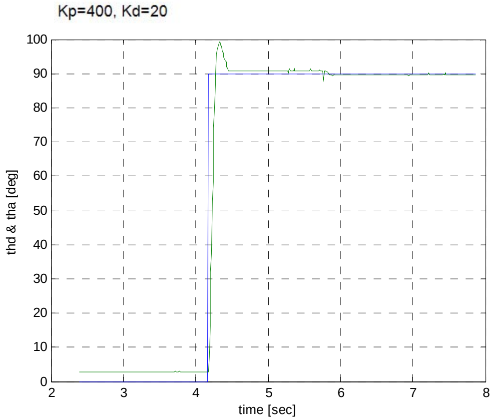




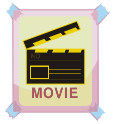
[Kp=300, Kd=0, Ki=0]



[Kp=300, Kd=12, Ki=0]

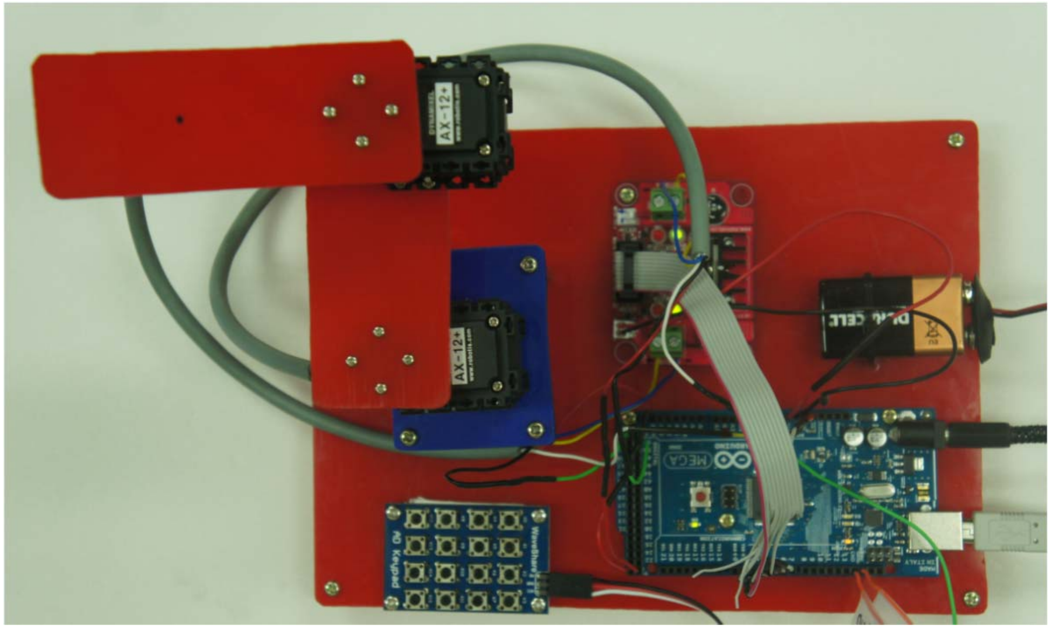


[Kp=300, Kd=12, Ki=100]

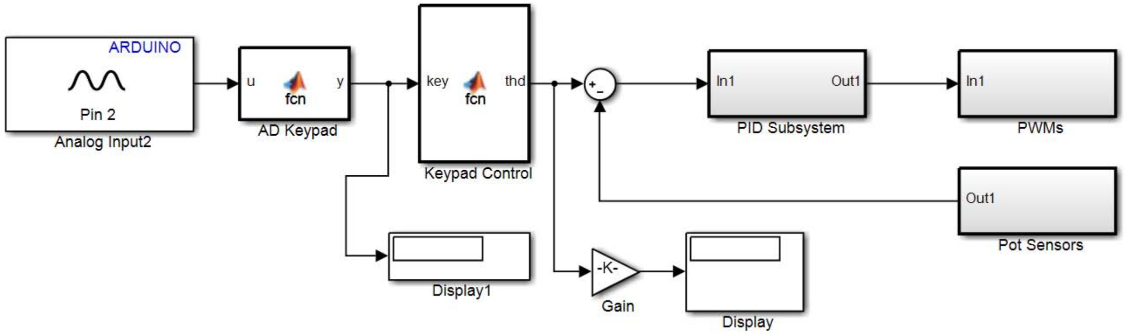


[5-4] 2-DOF Robot Arm Project

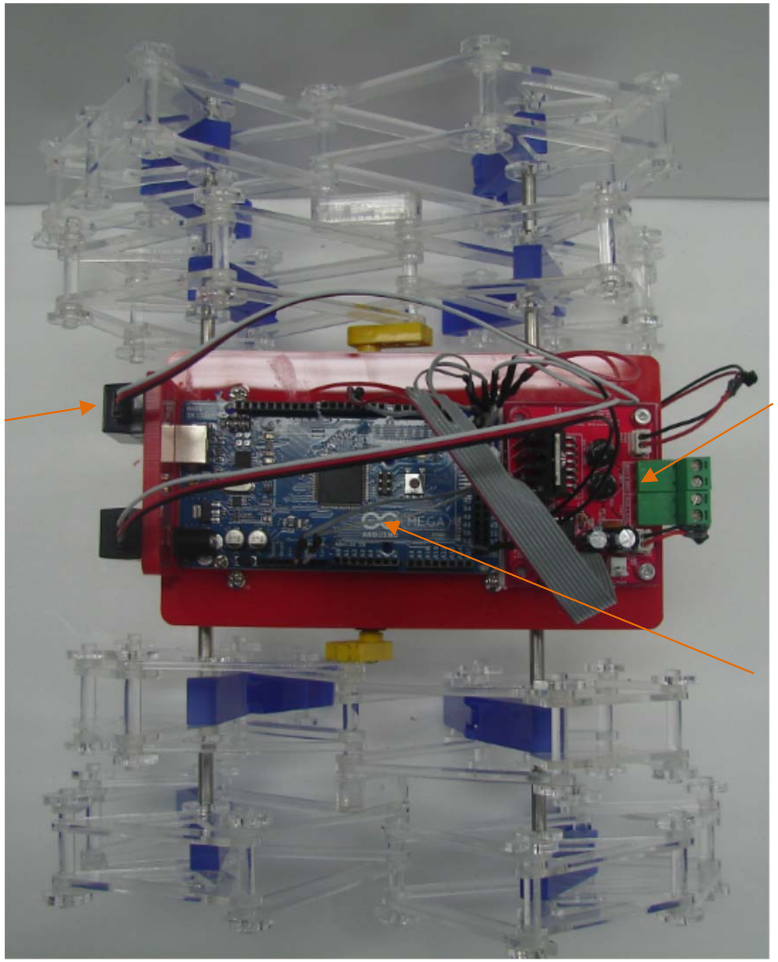
2-DOF Robot Arm with Keypad Control:



Forward Kinematics:
K0→+th1, K4→-th1,
K1→+th2, K5→-th2
Inverse Kinematics:
K2→+x, K6→-x,
K3→+y, K7→-y



[5-5] Walking Robot



3 IR Sensors

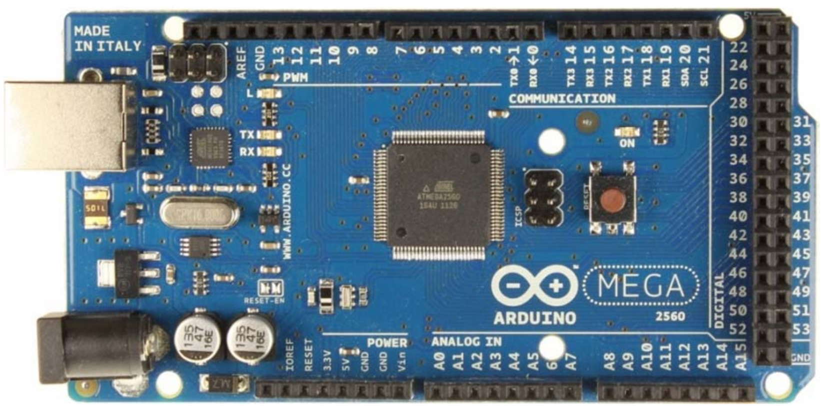
2-ch PWM AMP

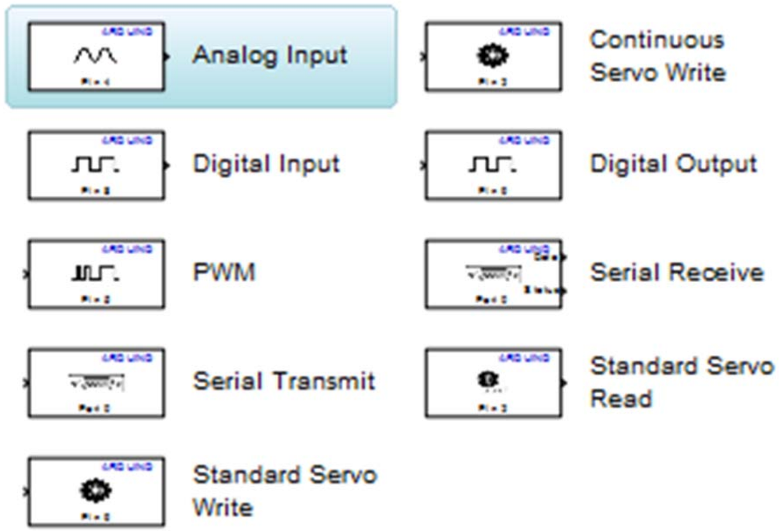
Arduino Mega 2560

Arduino Interface: Arduino Mega 2560

Arduino Mega 2560

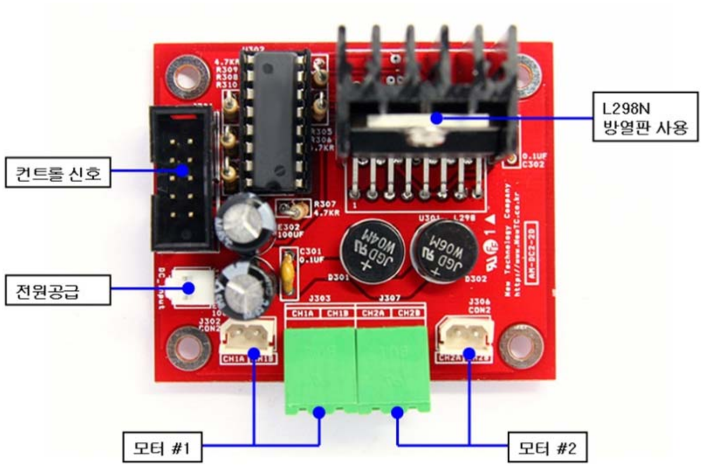
- ATmega 2560: 16MHz, 256kB Flash, 54-DIO, 14-PWM, 16-AI, 4-UART, USB connector
- Supported I/O drivers: AI, DI, DO, AI, PWM, Servo Read/Write, Serial Receive/Transmit
- MathWorks webpage: http://www.mathworks.co.kr/products/simulink/simulink-targets/

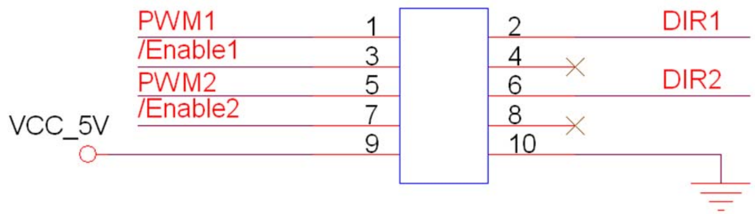




Arduino Interface: Parts

- **DC Motor Amplifier(AM-DC2-2D)** (2ch, 2A, 5~45V)



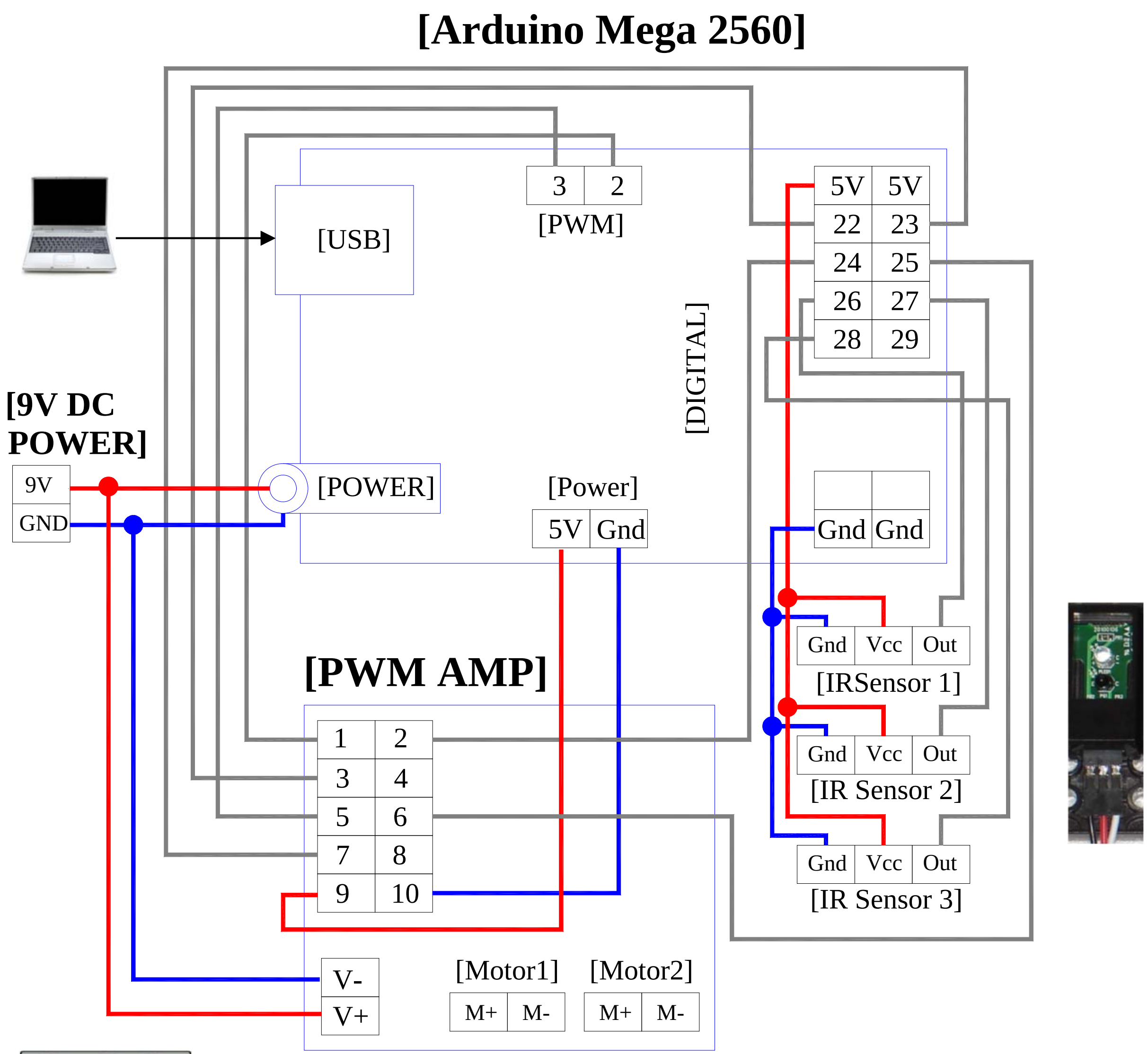


- Geared DC Motor (60RPM, 3.5kgf.cm at DC 6V)

■ SPECIFICATION

Motor Type	3 Bole Ferrite
Bearing Type	Nylon Bushing
Torque 4.8/6.0v	3.0 / 3.5 kg.
Speed 4.8/6.0v	0.19 / 0.15 second
Size	41 x 20 x 37mm
Weight	48.5 g.

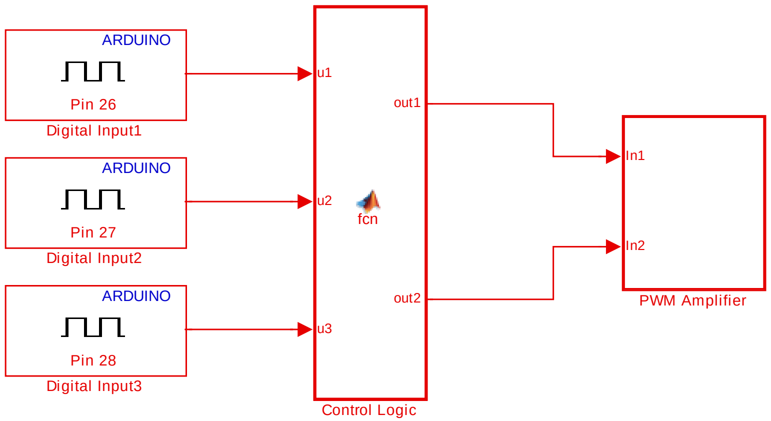
Arduino Interface: Connection

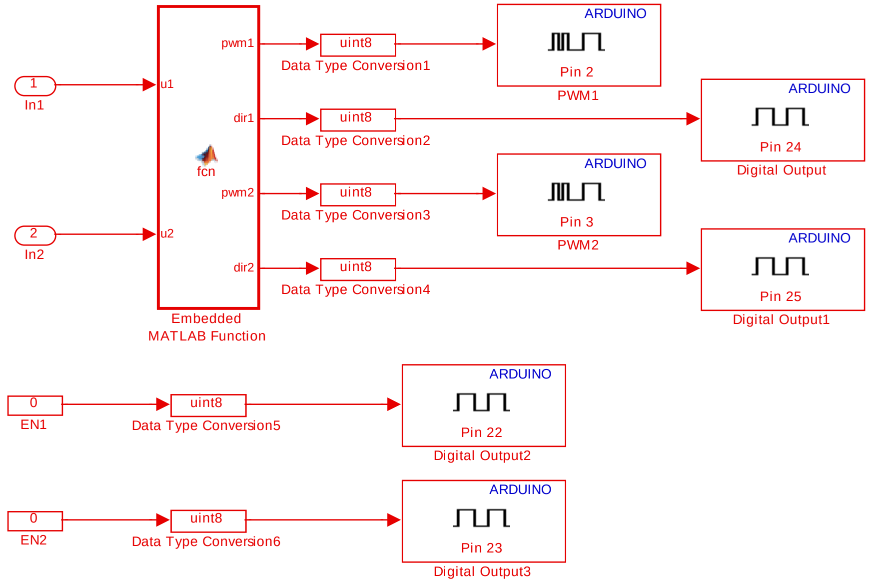


Arduino Interface: Control Program

Control of 2 DC motors by 3 IR sensors

If IR3=ON, Enable
If IR1=OFF& IR2=OFF, then move forward
If IR1=ON & IR2=OFF, then turn left
If IR1=OFF & IR2=ON, then turn right
If IR1=ON & IR2=ON, then move backward



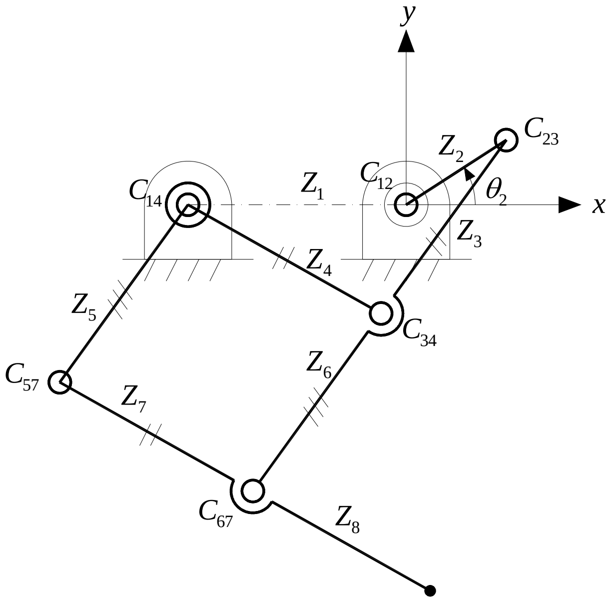



```
function [out1, out2] = fcn(u1, u2, u3)

% When IR sensor is blocked (or reflected), u=0
% When IR sensor is not blocked (or not reflected), u=1
out1=0; out2=0;
%sp1=250; sp2=-250;
sp1=-190; sp2=170;      % 0~255 %
if u3==0,
   if u1==1 && u2==1,  out1=sp1; out2=sp2; end     % Forward %
   if u1==0 && u2==1,  out1=0; out2=sp2; end     % Left %
   if u1==1 && u2==0,  out1=sp1; out2=0; end     % Right %
   if u1==0 && u2==0,  out1=-sp1; out2=-sp2; end     % Backward %
else
   out1=0; out2=0;
end
```

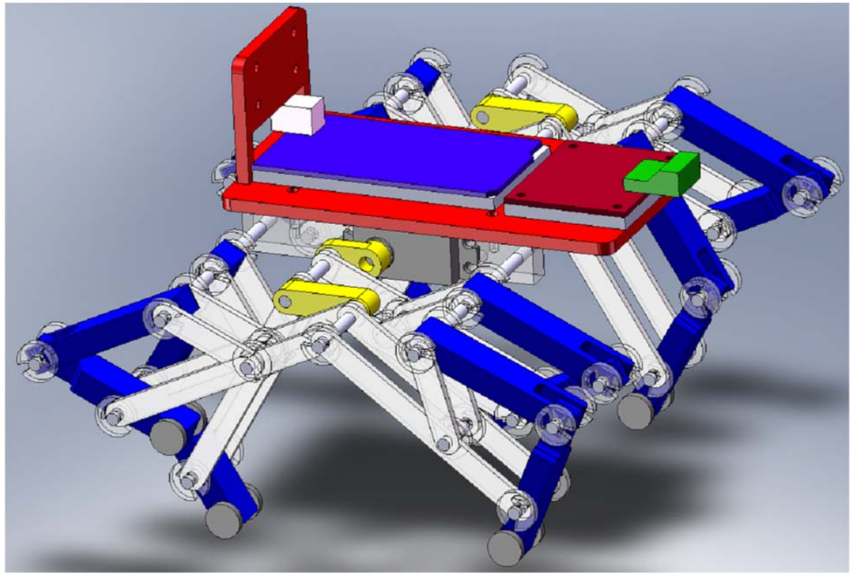
[PWM Amplifier]

6-bar Leg Mechanisms

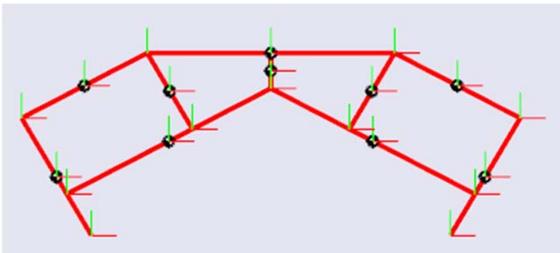


Z_1	Z_2	Z_3	Z_4	Z_5	Z_6	Z_7	Z_8
45	15	37.5	37.5	60	60	37.5	22.5

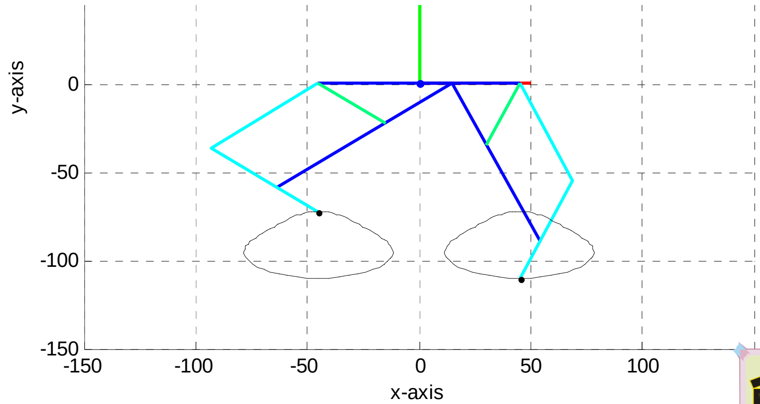
[6-bar mechanism]



[3-D Modeling]



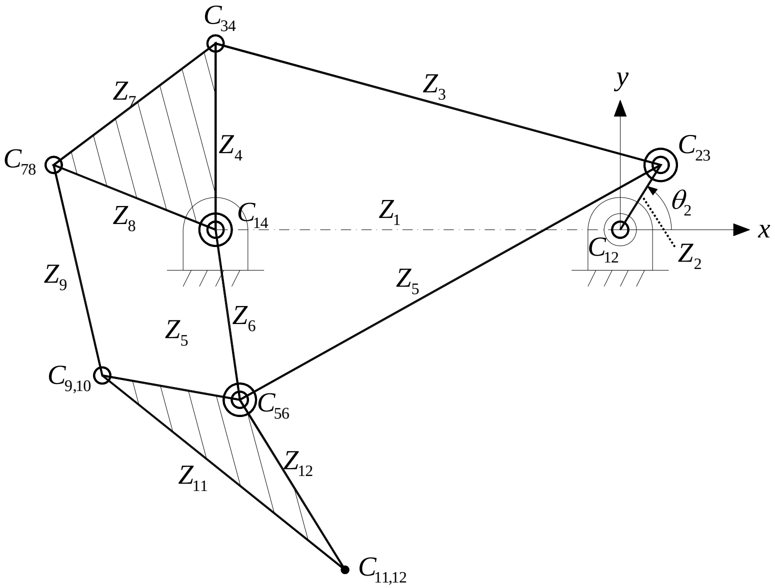
[SimMechanics]

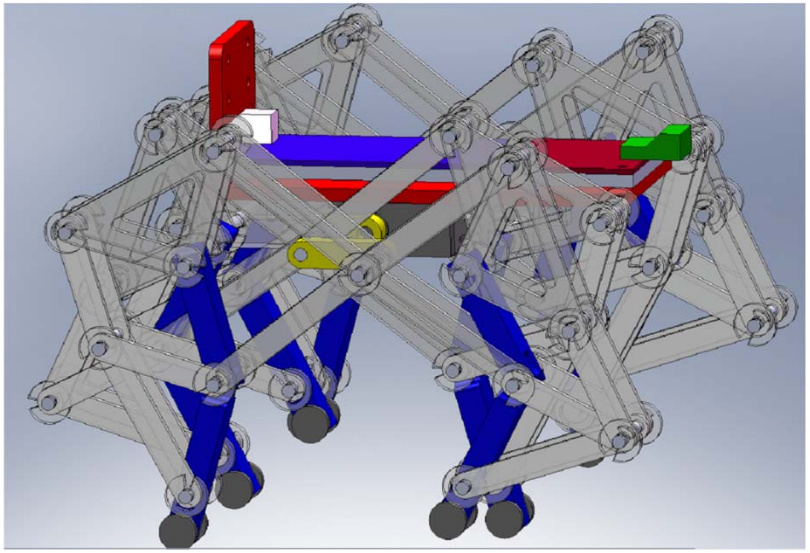


[MATLAB Simulation]

MOVIE

8-bar Leg Mechanisms

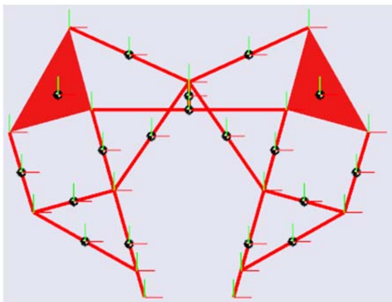




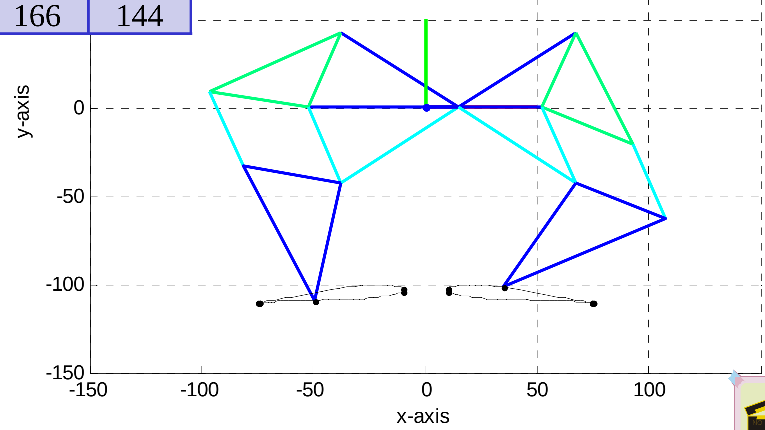
[3-D Modeling]

Z_1	Z_2	Z_3	Z_4	Z_5	Z_6	Z_7	Z_8	Z_9	Z_{10}	Z_{11}	Z_{12}
90	28	128	83	128	83	115	83	83	83	166	144

[8-bar mechanism (Theo mechanism)]



[SimMechanics]

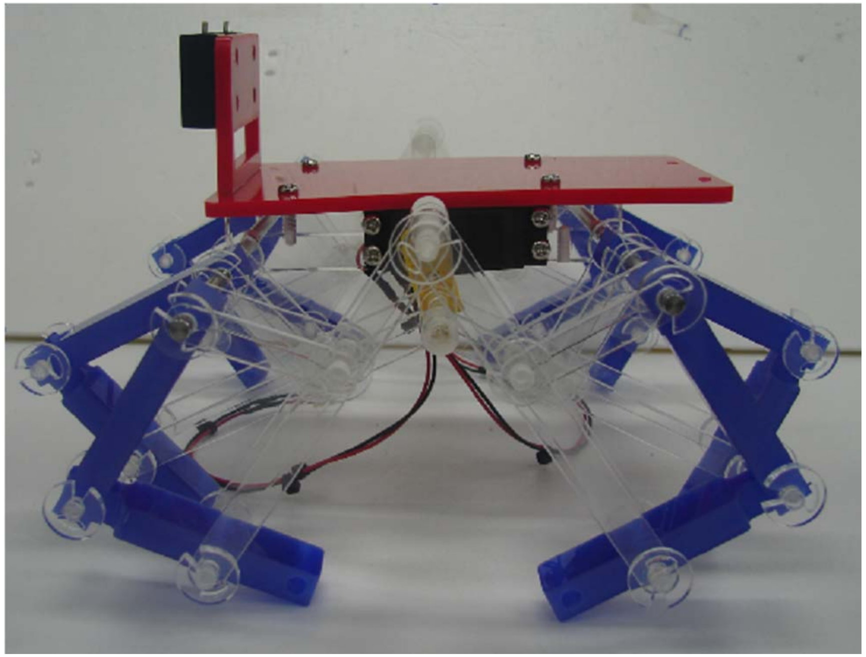


[MATLAB Simulation]

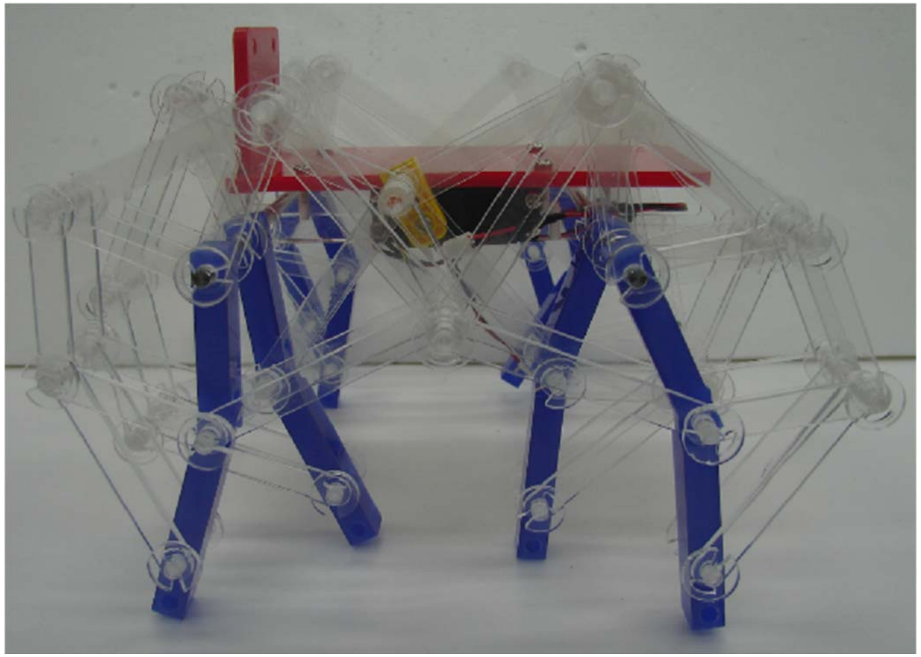
MOVIE

MathWorks

Walking Robot Prototypes



[6-bar Legs]

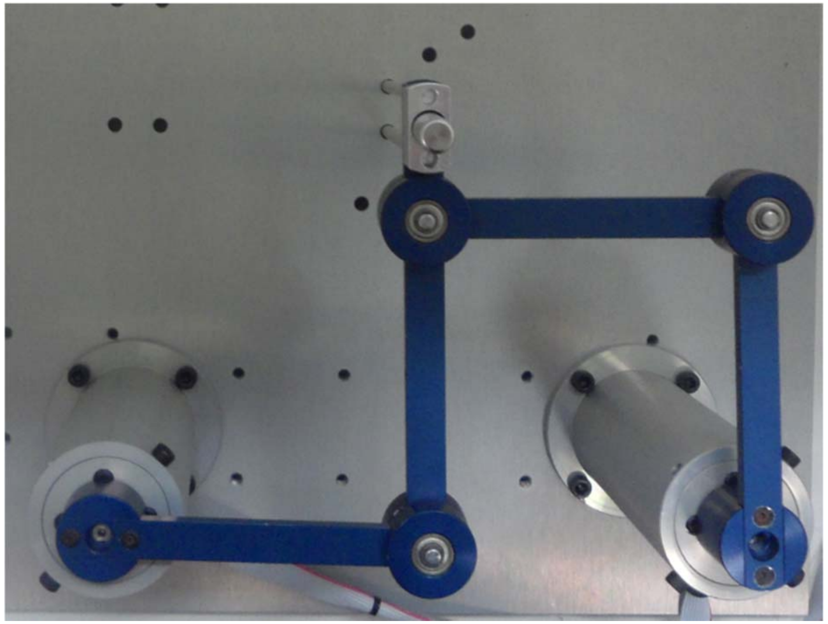


[8-bar Legs]

- ❖ Operated by two DC motors and Move forward/backward/right/left
- ❖ Statically balanced walking robot

[5-6] Haptic Interface

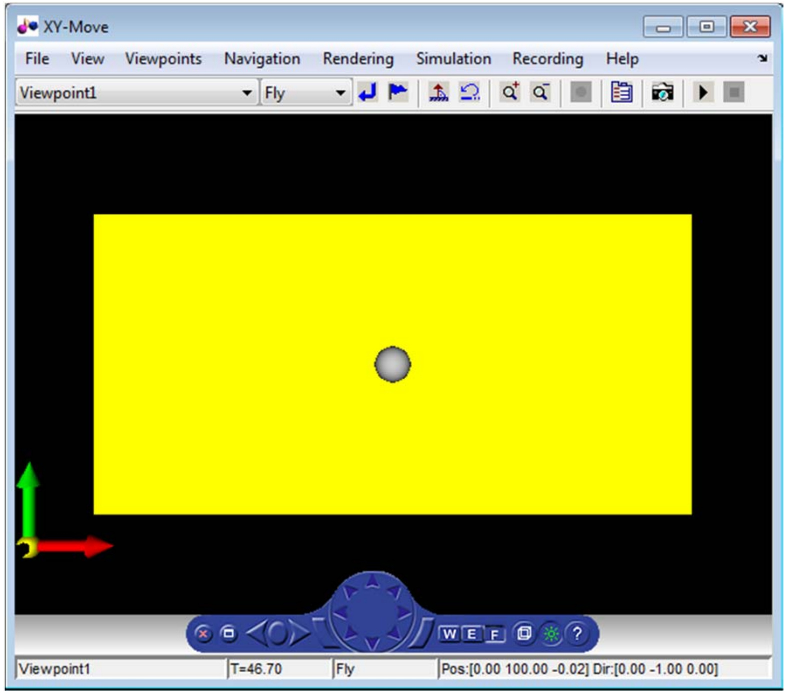
- **2-DOF Parallel-type Haptic Device**
  - ✓ Forward kinematics module
  - ✓ Force-Reflecting module
  - ✓ VRML graphic module



[Haptic Device]

position

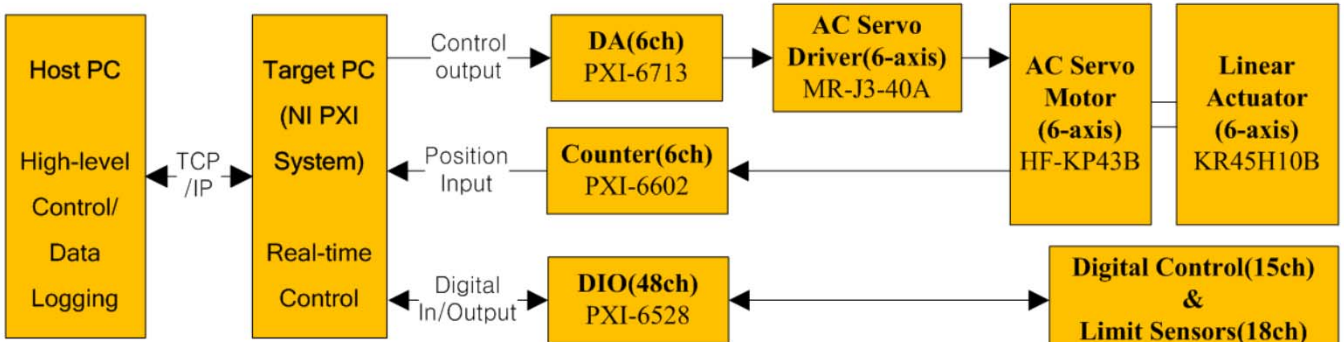
force

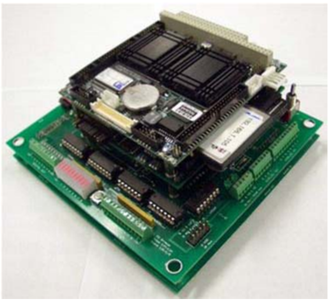


[Virtual Environment]

[5-7] PC-based Controller with xPC Target

[PC-based, DAQ-based]





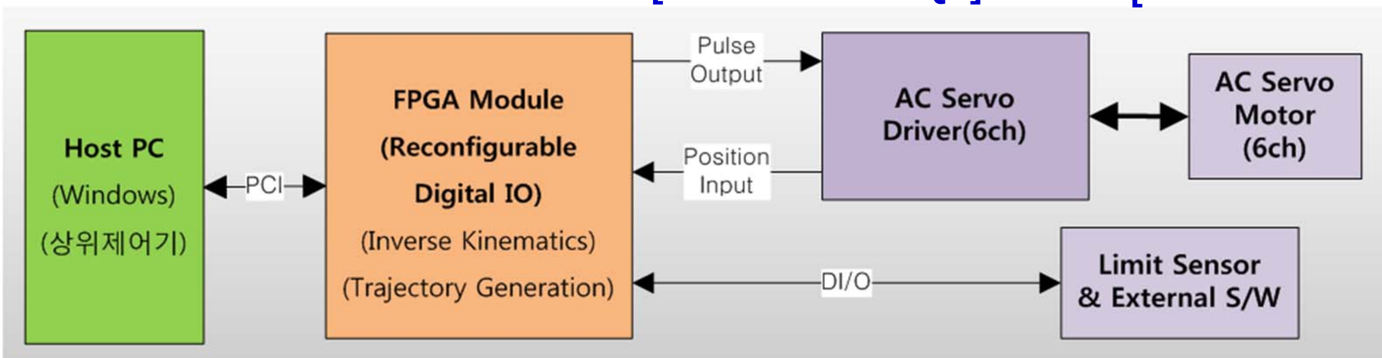
[PC-104]

[speedgoat]

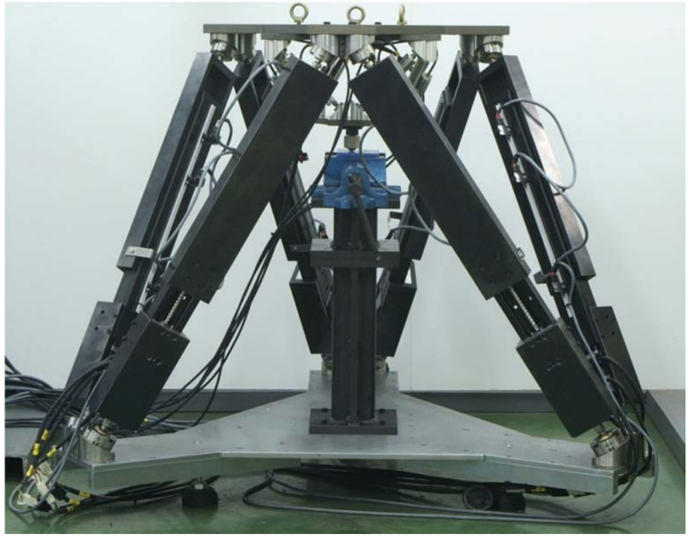
[NI PXI with DAQs]

[AC Motor Drivers]

[DSP-based]
[FPGA-based]

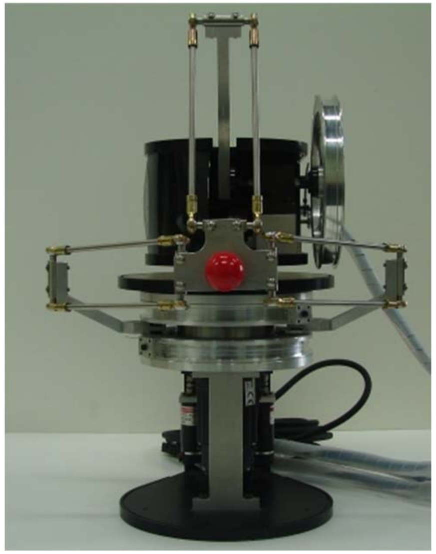


Robot Developments by xPC Target



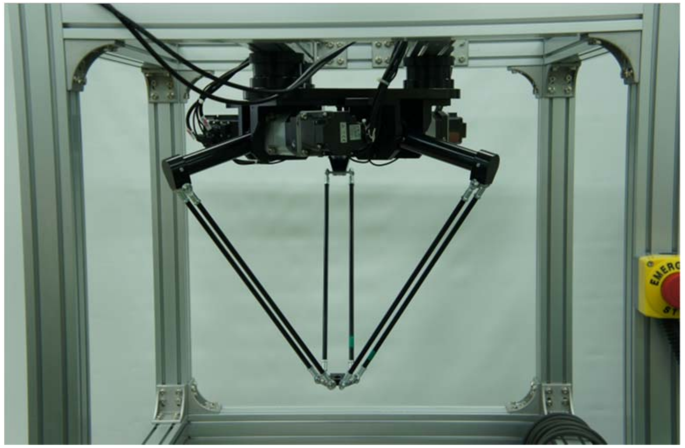
[6-DOF Parallel Robot]

[6-axis Motion Simulator]



[3-DOF Haptic Device]

[6-axis Force/Torque Sensor]



[Delta Parallel Robot]

[Walking Robot]

[6] Conclusions

- **This hand-on notes can be used partially or fully in Robotics, Control Design, Mechatronics system, and Capstone Design.**

- **The hand-on notes with drawings and MATLAB & Simulink examples will be published at the following website.**
  - ✓ http://imecha.kyungnam.ac.kr/rndlab

- **More project practices will be developed.**