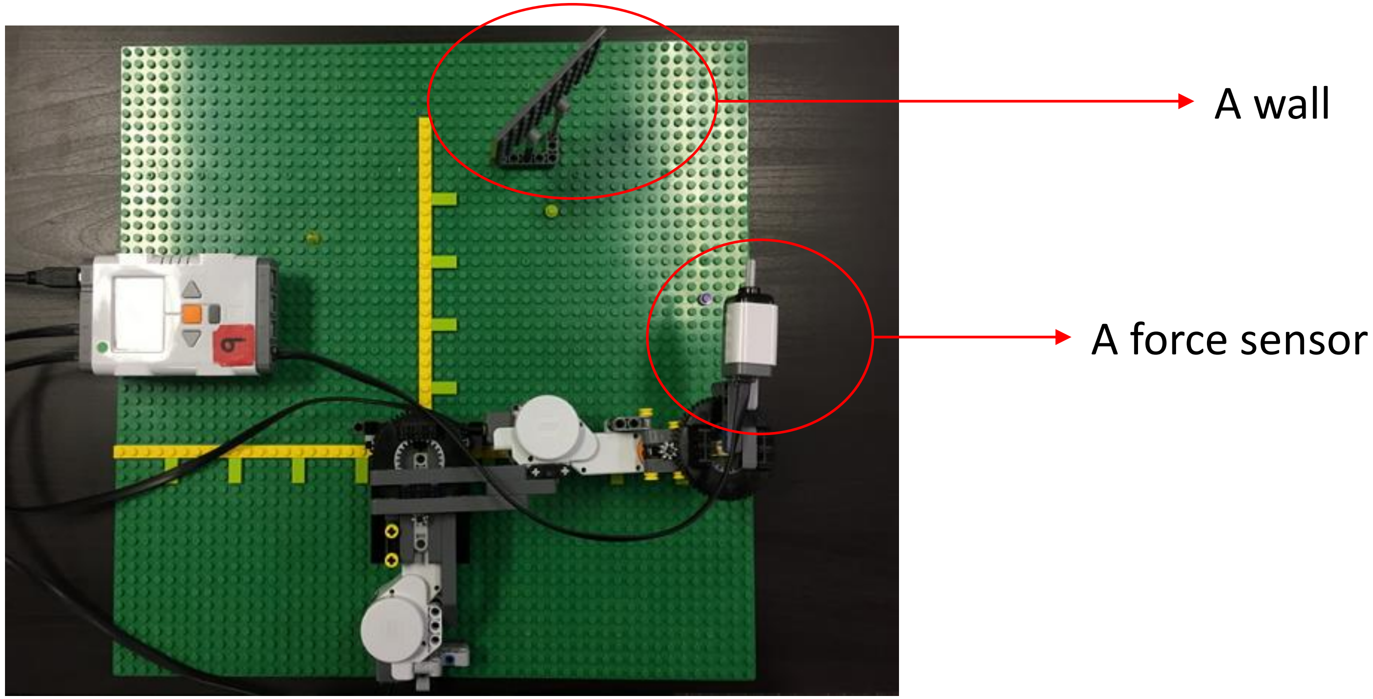
ME729 Advanced Robotics - Lab #8

4/09/2018

Sangsin Park, Ph.D.

- Modify our manipulator.
  - We'll modify two parts.
  - Before experiments, make sure a wall position and the manipulator's configuration.
  - **The wall is positioned at 45 degrees.**
  - **The link 2 of the manipulator is rotated at 90 degrees.**



- ❑ Assemble a force sensor on manipulator's tip (see fig. 1).
    - Before this job, it might need to disassemble a current tip.

- ❑ Build a wall.
    - Make a wall like fig 2.
    - Place a bar on the plate such as fig 3.
    - Then, make the wall stand.

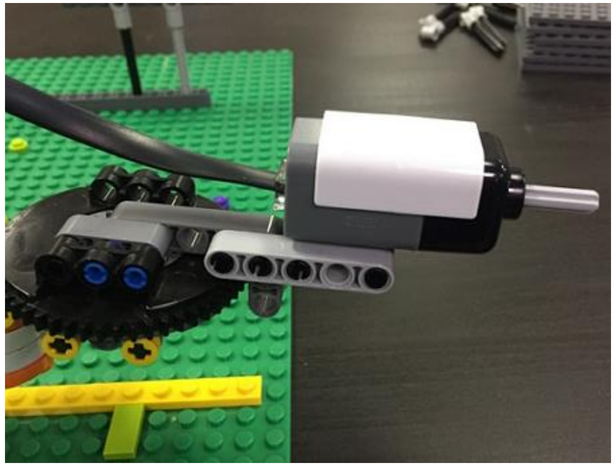


Fig. 1

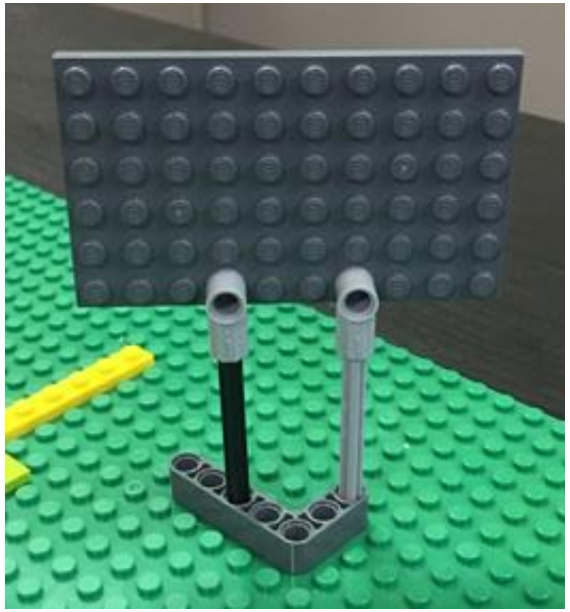


Fig. 2

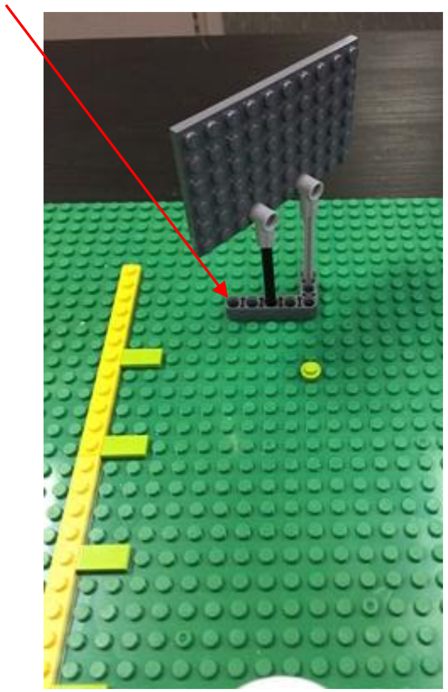
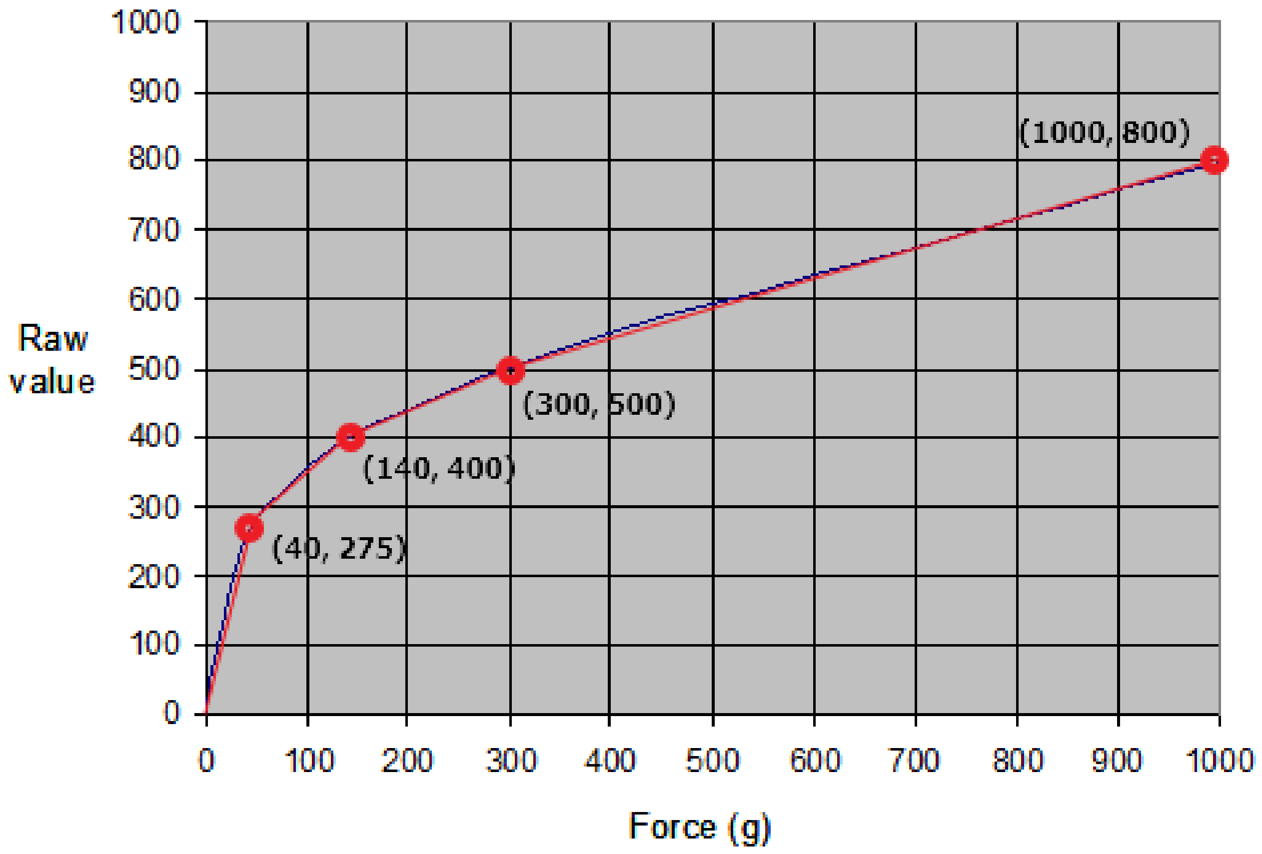


Fig. 3

- About force sensor.
  - The sensor's raw value according to applied force shows nonlinearity.
  - Before curve fitting, the curve is split into 4 segments.
  - Then, fit the curve by linear functions.



❑ Contact situation

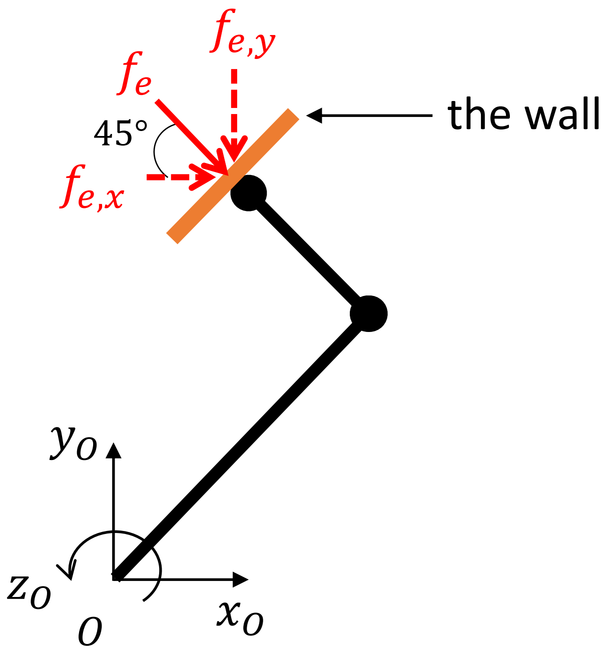
- The contact force f_e is applied to normal direction of the wall.

- Since the wall is positioned at 45 degrees, f_e can be divided into the following.

$$f_{e,x} = f_e \cos 45°$$
$$f_{e,y} = f_e \sin 45°$$

- Therefore, the desired force f_d is also divided by the same way.

$$f_{d,x} = f_d \cos 45°$$
$$f_{d,y} = f_d \sin 45°$$



❑ Source code

- Based on the code to make the joint trajectory.
- It needs a timer to measure joints angle and force.
- In main()

```
// Stop move
if(orangeBtnPushed == TRUE)
{
    orangeBtnPushed = FALSE;

    Off(JNT1);
    Off(JNT2);
}

// Go forward
if(r_ArrowBtnPushed == TRUE)
{
    r_ArrowBtnPushed = FALSE;

    OnFwd(JNT1, 30);
    OnFwd(JNT2, 0);
}

// Go backward
if(l_ArrowBtnPushed == TRUE)
{
    l_ArrowBtnPushed = FALSE;

    OnRev(JNT1, 30);
    OnRev(JNT2, 0);
}

// Check contact
if(force >= 15.0)
{
    forceControl(20);
}

if(flag_readSensor == TRUE)
{
    flag_readSensor = FALSE;
    readSensor();
}

timer();
```

To stop the move, press the orange button.

To rotate counterclockwise, press the right arrow button.

To rotate clockwise, press the left arrow button.

Check if the manipulator is contact with the wall.
If yes, force control is activated.

Measure sensor data in real-time.

❑ Source code – continued

- readSensor()

Force data read

Curve fitting section.

Encoder data read

```
void readSensor()
{
    // Force Sensor
    int rawData = 1023 - SensorHTForce(S1);
    int data = rawData;

    if(data <= 275)
    {
        force = 0.1455*data;
    }
    else if(data > 275 && data <= 400)
    {
        force = 0.8*data - 180;
    }
    else if(data > 400 && data <= 500)
    {
        force = 1.6*data - 500;
    }
    else if(data > 500)
    {
        force = 2.33*data - 865;
    }

    TextOut(0, LCD_LINE3, "                    "); // Clear previous text
    TextOut(0, LCD_LINE3, FormatNum("Force: %.2f", force));

    // Encoder
    enc1 = MotorRotationCount(JNT1)/gearRatio*D2R;
    enc2 = MotorRotationCount(JNT2)/gearRatio*D2R;

    d_enc1 = (enc1 - prev_enc1)/timer_period;
    d_enc2 = (enc2 - prev_enc2)/timer_period;
    prev_enc1 = enc1;
    prev_enc2 = enc2;
}
```

- Source code – continued
  - forceControl()

```
void forceControl(int f_d)
{
    float tau_1 = 0.0;
    float tau_2 = 0.0;
    float err_x = 0.0;
    float err_y = 0.0;
    float fdx = 0.0;
    float fdy = 0.0;
    float kp = 10.0;
    float kv = 10.0;
    float s1 = sin(enc1);
    float c1 = cos(enc1);
    float c2 = cos(enc2);
    float s12 = sin(enc1 + enc2);
    float c12 = cos(enc1 + enc2);
    float M1 = (m1+m2)*L1*L1 + m2*L2*L2 + 2*m2*L1*L2*c2;
    float M2 = m2*L2*L2 + m2*L1*L2*c2;
    float M3 = m2*L2*L2;
    float a = -L1*s1 - L2*s12;
    float b = -L2*s12;
    float c = L1*c1 + L2*c12;
    float d = L2*c12;
    float tau_gain = 50.0;

    fdx = f_d*0.5253;
    fdy = f_d*0.5253;
    err_x = (fdx - force*0.5253);
    err_y = (fdy - force*0.5253);

    tau_1 = (-a*a*M1*d_enc1 + (-2*M2*d_enc1*c - d_enc2*(b*M1 + d*M2))*a
            -c*(M3*d_enc1*c + d_enc2*(b*M2 + d*M3)))*kv
            +(err_x*kp*M1 + err_y*kp*M2 + fdx)*a
            +c*(err_x*kp*M2 + err_y*kp*M3 + fdy);
    tau_2 = (-M1*d_enc2*b*b + (-2*M2*d_enc2*d - d_enc1*(a*M1 + c*M2))*b
            -d*(M3*d_enc2*d + d_enc1*(a*M2 + c*M3)))*kv
            +(err_x*kp*M1 + err_y*kp*M2 + fdx)*b
            +d*(err_x*kp*M2 + err_y*kp*M3 + fdy);
```

$$M(\theta) = \begin{bmatrix} m_1 & m_2 \\ m_2 & m_3 \end{bmatrix}$$

$$J(\theta) = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

```
    if (tau_1 >= 0)
    {
        OnRev(JNT1, (tau_gain*tau_1));
    }
    else
    {
        OnFwd(JNT1, (tau_gain*(-tau_1)));
    }

    if (tau_2 >= 0)
    {
        OnRev(JNT2, (tau_gain*tau_2));
    }
    else
    {
        OnFwd(JNT2, (tau_gain*(-tau_2)));
    }
}
```

Calculate motor output power and run motor.