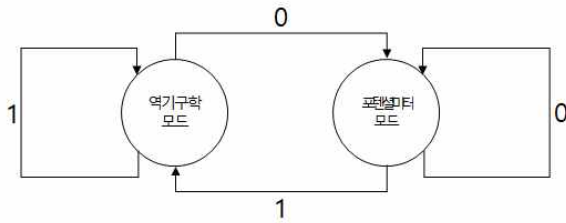


임베디드 로보틱스 최종보고서

1. 개요

- 1) 기본적으로 본 프로젝트에서는 포텐셜 미터모드와 역기구학 모드를 지원합니다.
- 2) 본 프로젝트는 FSM (밀리머신)형태로 구성되어 있습니다.

가)



나) 아두이노 환경에서 구현하기 위해 'void loop()' 구문안에 FSM 코드를 삽입해두었습니다.

다) 각 모드에 해당하는 모드함수를 호출하는 형태로 구성되어있습니다.

```

void loop() {
  switch (readModeSW()) {
    case POTMODE:
      potMode();
      break;
    case INVERSEKINEMATICMODE:
      inverseMode();
      break;
  }
}
  
```

3) 해당 모드는 토글 아두이노 우노 D9 핀에 풀업저항과 연결된 토글스위치를 통해 모드를 전환합니다.

4) 이를 인식하는 'int readModeSW()' 라는 함수를 만들어졌습니다

```

int readModeSW(){
  bool modeSW = digitalRead(9);
  if(modeSW == 0){
    digitalWrite(5, HIGH);
    return POTMODE;
  }
  else if(modeSW == 1){
    digitalWrite(5, LOW);
    return INVERSEKINEMATICMODE;
  }
  else
    return ERROR;
}
  
```

가)

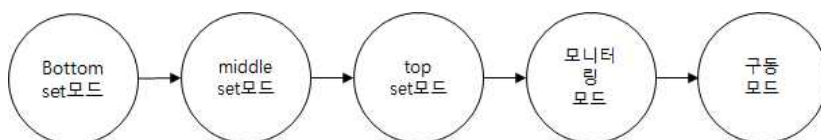
2. 포텐셜미터 모드

가. 포텐셜미터 모드는 3DoF로 구성된 로봇팔의 Bottom, middle, top 모터의 각도값을 포텐셜미터로 값을 받고 넘겨주는 구조입니다.

나. 포텐셜미터를 돌려주고 D10에 연결된 토글스위치를 꺾어주면 각도값이 저장됩니다. Bottom, middle, top 모터의 각도값을 지정하기 위해 해당과정을 3회 반복해야합니다.

다. 포텐셜미터 모드는 FSM 구조로 작동합니다. 이를 구현하는 'void potMode()'를 선언했습니다.

1)



2) 코드는 아래와 같습니다. 아두이노 우노 ADC 10bit 해상도입니다. 하지만, 다이내믹셀의 각도 해상도는 12bit입니다. 이에 포텐셜미터 ADC 값에 4를 곱해줌으로서 맵핑했습니다.

```

void potMode(){
    static int state = 0;
    switch (state)
    {
    case 0: //Bottom set 모드
        while(1){
            potVal[POT_BOTTOM] = analogRead(A2)*4;
            Serial.print("POT_BOTTOM : ");
            Serial.println(potVal[POT_BOTTOM]);
            if (toggleSW() == 1) {
                state = 1;
                break;
            }
        }
        break;
    case 1: //middle set 모드
        while(1){
            potVal[POT_MIDDLE] = analogRead(A2)*4;
            Serial.print("POT_MIDDLE : ");
            Serial.println(potVal[POT_MIDDLE]);
            if (toggleSW() == 1) {
                state = 2;
                break;
            }
        }
        break;
    case 2: //top set모드
        while(1){
            potVal[POT_TOP] = analogRead(A2)*4;
            Serial.print("POT_TOP : ");
            Serial.println(potVal[POT_TOP]);
            if (toggleSW() == 1) {
                state = 3;
                break;
            }
        }
        break;
    }
}

```

가) Bottom set, middle set, top set 모드 전환은 'bool toggleSW()'함수를 통해 전환했습니다. 해당함수는 토글스위치의 state가 변환되면 1를 출력합니다

<pre> bool toggleSW(){ static bool prev_btn; static bool current_btn; current_btn = digitalRead(10); if ((prev_btn ^ current_btn) == 1){ prev_btn = current_btn; Serial.println("SW"); return 1; } } </pre>	<pre> else{ prev_btn = current_btn; return 0; } } </pre>
--	--

3) 다음은 모니터링 모드와 구동모드입니다. 모니터링 모드에서 입력된 각도 정보를 확인하고 바로 다이내믹셀에게 각도 정보를 송신하여 구동하게 됩니다.

```

case 3: //모니터링 모드
    Serial.print(potVal[POT_BOTTOM]);
    Serial.print("\t");
    Serial.print(potVal[POT_MIDDLE]);
    Serial.print("\t");
    Serial.println(potVal[POT_TOP]);
    state = 4;
    break;

case 4: //구동 모드
    motorDeg[POT_BOTTOM] = potVal[POT_BOTTOM];
    motorDeg[POT_MIDDLE] = potVal[POT_MIDDLE];
    motorDeg[POT_TOP] = potVal[POT_TOP];

    ref_0[0] = bottomPacket(motorDeg[0]);
    ref_0[1] = topPacket(motorDeg[0]);
    dx_tx_packet_for_position_control(BOTTOM_ID, ref_0);
    delay(1000);
    ref_1[0] = bottomPacket(motorDeg[1]);
    ref_1[1] = topPacket(motorDeg[1]);
    dx_tx_packet_for_position_control(MIDDLE_ID, ref_1);
    delay(1000);
    ref_2[0] = bottomPacket(motorDeg[2]);
    ref_2[1] = topPacket(motorDeg[2]);
    dx_tx_packet_for_position_control(TOP_ID, ref_2);
    delay(1000);

    while (1) {
        if (toggleSW() == 1) {
            state = 0;
            break;
        }
    }
    break;
}
}

```

가) 다이나믹셀은 각도값을 상단 unsigned char, 하단 unsigned char로 나눠서 전송받습니다. 이를 구현해 주기 위해 아래의 두함수를 선언해주었습니다

나)

```

unsigned char bottomPacket(int REF){
    return REF % (16*16);
}

```

다)

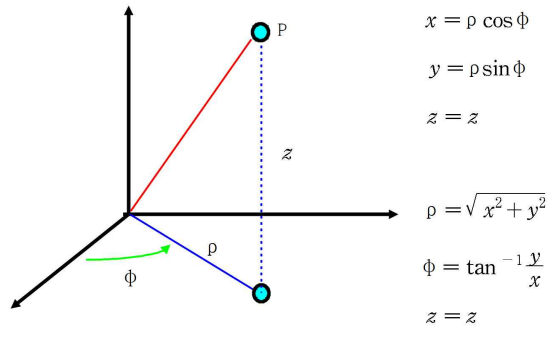
```

unsigned char topPacket(int REF){
    return (REF/(16*16))%(16*16);
}

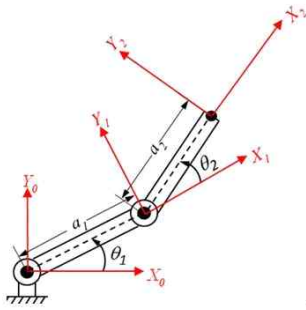
```

3. 역기구학 모드

가. 3차원 역기구학 문제를 해결하기 어려웠습니다. 이에 문제를 쉽게 해결 하기 위해 직교좌표계로 들어오는 좌표정보를 원통좌표계로 변환하였습니다. 변환과정에서 얻은 Pi정보를 첫번째 관절에 제공해줍니다. 그럼 문제는 rho(코드 상에는 r로 표기하였습니다)와 z로 이뤄진 2차원 역기구학 문제로 단순화 됩니다.



나. 단순화된 2차원 역기구학 문제와 답은 아래와 같습니다.



$$\theta_1 = \tan^{-1} \frac{P_y}{P_x} - \tan^{-1} \frac{a_2 \sin \theta_2}{a_1 + a_2 \cos \theta_2}$$

$$\theta_2 = \pm 2 \tan^{-1} \sqrt{\frac{(a_1 + a_2)^2 - (P_x^2 + P_y^2)}{(P_x^2 + P_y^2) - (a_1 - a_2)^2}}$$

다. theta1은 theta2에 의존성이 있습니다. 즉, 직교좌표계->원통좌표계->theta2->theta1 순으로 계산해야합니다.
라. 이를 계산해주는 'void findInverseKinematic(float x, float y, float z)'를 아래와 같이 구성해줬습니다.

<pre>void findInverseKinematic(float x, float y, float z){ cartsia2cylinder(x, y, z); float theta2 = findTheta2(r,z); float theta1 = findTheta1(r,z,theta2); motorDeg[0] = int(pi*11); motorDeg[1] = int(theta1*11); motorDeg[2] = int(theta2*11); }</pre>	<pre>void cartsia2cylinder(double x, double y, double _z){ double cord[3]; pi = atan2(y,x) * 57.2958; r = sqrt(x*x + y*y); z = _z; } double findTheta2(double r, double z){ double innerSqrt = ((L1 + L2)*(L1 + L2) - (r*r + z*z))/ ((r*r + z*z) - (L1 - L2)*(L1 - L2)); return 2*atan(sqrt(innerSqrt))* 57.2958; } double findTheta1(double r, double z, double th2){ double firstTerm = atan2(z,r)*57.2958; double secondTerm = atan2(L2*sin(th2),(L1 + L2*cos(th2)))* 57.2958; return firstTerm - secondTerm; }</pre>
--	--

마. 이러한 역기구학 문제를 계산하고 모터를 구동하는 'void inverseMode()'를 선언해주었습니다. UART로 좌표정보를 수신받고, 해당좌표 정보를 'void findInverseKinematic()'를 통해 각 관절의 각도 정보를 변환합니다. 이 변환된 각도정보를 서보에 전달하여 구동하는 구조입니다.

<pre>void inverseMode(){ char inCom; static int state = 0; switch (state) { case 0: for (int i = 0; i < 3; i++) { Serial.print("INPUT (since 3Sec) : \n"); delay(3000); if (Serial.available()){ cartsianCord[i] = Serial.parseInt(); } } while (1) { char inCome; Serial.print("Press 'a' to start\n"); if (Serial.available()){ inCome = Serial.read(); } if (inCome == 'a') { break; } } state = 1; break; </pre>	<pre>case 1: findInverseKinematic(cartsianCord[0],cartsianCord[1], cartsianCord[2]); ref_0[0] = bottomPacket(motorDeg[0]); ref_0[1] = topPacket(motorDeg[0]); dx_tx_packet_for_position_control(BOTTOM_ID, ref_0); delay(1000); ref_1[0] = bottomPacket(motorDeg[1]); ref_1[1] = topPacket(motorDeg[1]); dx_tx_packet_for_position_control(MIDDLE_ID, ref_1); delay(1000); ref_2[0] = bottomPacket(motorDeg[2]); ref_2[1] = topPacket(motorDeg[2]); dx_tx_packet_for_position_control(TOP_ID, ref_2); delay(1000); state = 2; break; case 2: while (1) { char inCome; Serial.print("Press 'a' to restart\n"); if (Serial.available()){ inCome = Serial.read(); } if (inCome == 'a') { state = 0; break; } } } }</pre>
---	---



5. [Github link](https://github.com/js4ngu/embedded_robotics_2022_MJU/tree/master/Project/upload)

https://github.com/js4ngu/embedded_robotics_2022_MJU/tree/master/Project/upload