Trajectory Planning

Path vs. Trajectory

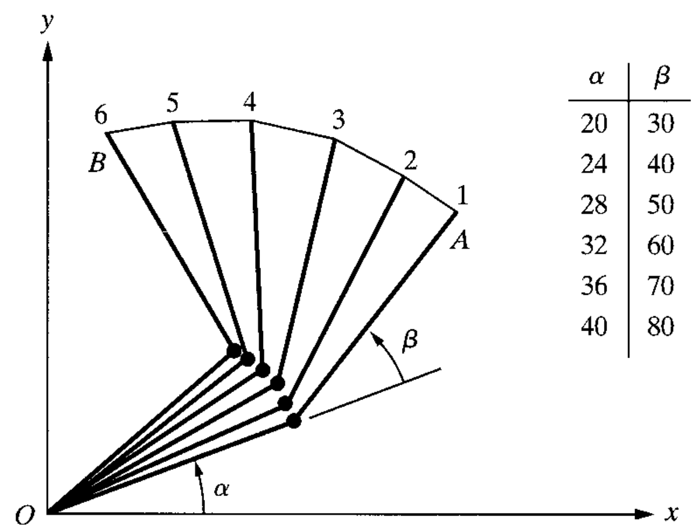
- Path (경로)
  - Sequences of points (configurations)
  - Path planning

- Trajectory (궤적)
  - Time history of position, velocity, and accelerations
  - Trajectory planning

❖ Motion Planning = Path Planning + Trajectory Planning

Joint Space vs. Cartesian Space

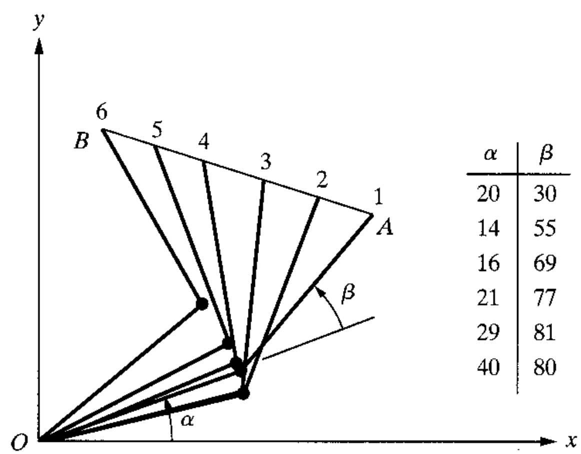
- Joint-space trajectory planning
  - Joint variable 의 interpolation
  - Point-to-Point (PTP) motion
    - Cartesian space 에서의 중간경로 예측 어렵다
    - 명령어: MOVE P1
    - Spot Welding



α	β
20	30
24	40
28	50
32	60
36	70
40	80

Joint Space vs. Cartesian Space

- Cartesian-space trajectory planning
  - Cartesian variable 의 interpolation
  - Linear motion / Circular motion 등
    - 중간 이동경로가 중요한 작업: Arc Welding 등
    - 명령어 예: MOVEL P1, MOVEC P2



α	β
20	30
14	55
16	69
21	77
29	81
40	80

Trajectory Planning 시 고려할 점

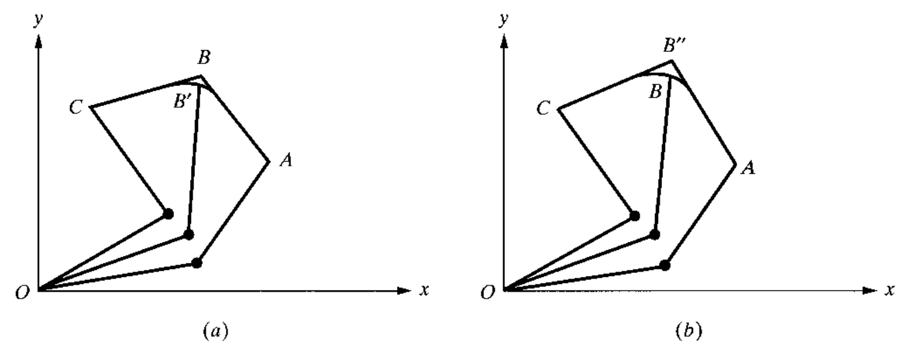
1) Initial & final conditions
   - Initial position, velocity, acceleration
   - Final position, velocity, acceleration

2) Smooth conditions

$$\left|\ddot{\theta}(t)\right| \le \ddot{\theta}_{\max} \qquad \left|\dddot{\theta}(t)\right| \le \dddot{\theta}_{\max}$$

(cf) jerky motion

3) Via points

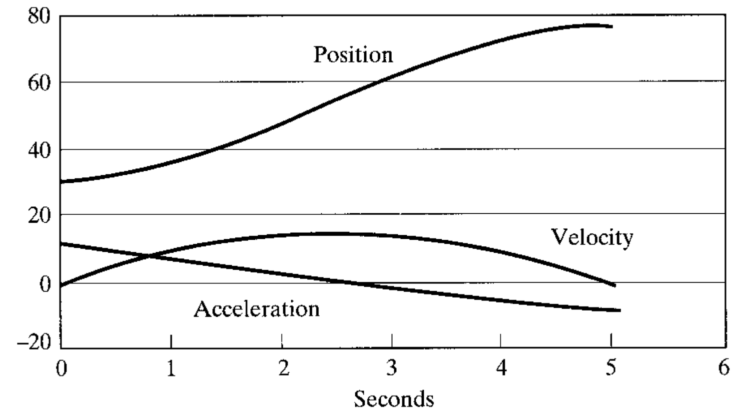


4) Elapsed time

Third-Order Polynomial Trajectory Planning

- Trajectory

$$\theta(t) = c_0 + c_1 t + c_2 t^2 + c_3 t^3$$
$$\dot{\theta}(t) = c_1 + 2c_2 t + 3c_3 t^2$$
$$\ddot{\theta}(t) = 2c_2 + 6c_3 t$$



- Initial / final conditions

$$\theta(t_i) = \theta_i \qquad \theta(t_f) = \theta_f$$
$$\dot{\theta}(t_i) = 0 \qquad \dot{\theta}(t_f) = 0$$

$\Rightarrow$ $c_0 \ , c_1 \ , c_2 \ , c_3$

$$c_0 = \theta_i$$
$$c_1 = 0$$
$$c_2 = 3\ (\theta_f - \theta_i)/\theta_f^2$$
$$c_3 = -2\ (\theta_f - \theta_i)/\theta_f^3$$

$$\left|\ddot{\theta}\right|_{\max} = \left|\frac{6(\theta_f - \theta_i)}{(t_f - t_i)^2}\right|$$

Third-Order Polynomial Trajectory Planning

(Example) $\theta_i = 30$(deg), $\theta_f = 75$(deg), $t_f = 5$(sec)

$\rightarrow$ Find $\theta(t), \dot{\theta}(t), \ddot{\theta}(t)$

3차 곡선 계획법

- C programming

시작점의 값이 0이고, 1초후 최종점의 값이 1000일 때, 3차 곡선 계획법에 의한 위치 중간점을 0.001초마다 구하는 컴퓨터 프로그램을 C언어를 이용하여 작성하라.

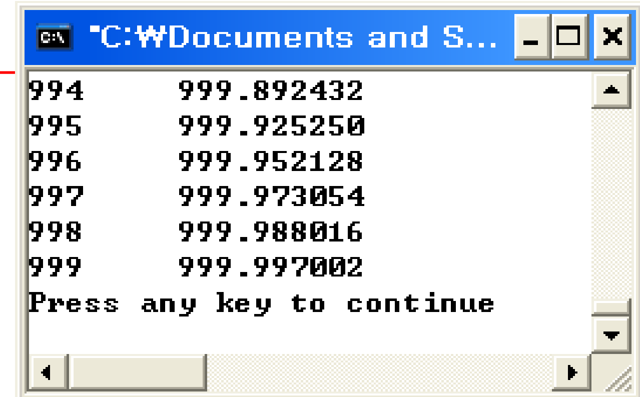
$$a_0 = p_i$$
$$a_1 = 0$$
$$a_2 = 3(p_f - p_i)/t_f^2$$
$$a_3 = -2(p_f - p_i)/t_f^3$$

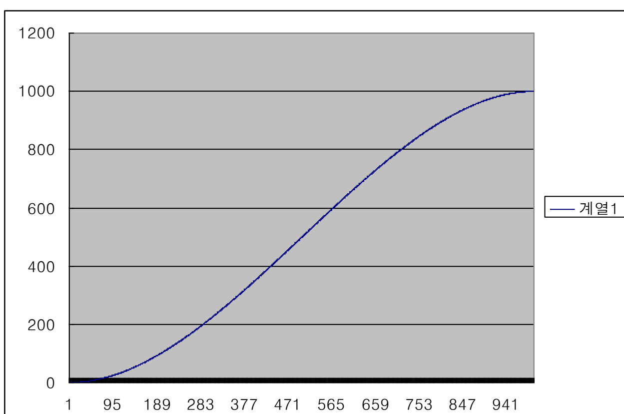
$$p(t) = a_3 t^3 + a_2 t^2 + a_1 t + a_0$$
$$v(t) = 3a_3 t^2 + 2a_2 t + a_1$$
$$a(t) = 6a_3 t + 2a_2$$

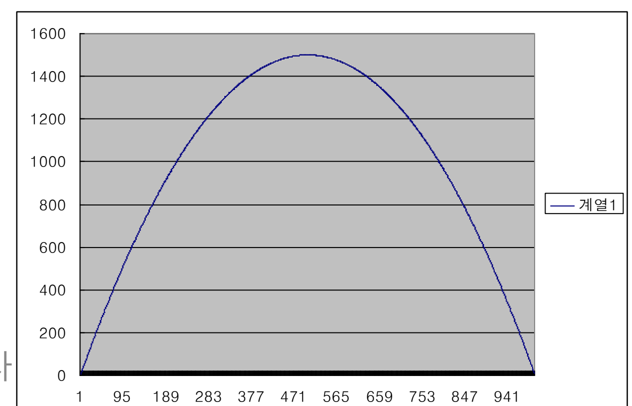
```c
#include<stdio.h>
#include<math.h>
#define FILE_NAME "cubic.xls"
struct motion
{
	double a;
	double v;
	double p;
};
void coeff(double ps,double pf,double tf,double a[])
{
	a[0]=ps;
	a[1]=0.0;
	a[2]=3.0*(pf-ps)/(tf*tf);
	a[3]=-2.0*(pf-ps)/(tf*tf*tf);
}
motion cubic(double t,double a[])
{
	motion m;
	m.p=a[3]*t*t*t+a[2]*t*t+a[1]*t+a[0];
	m.v=3*a[3]*t*t+2*a[2]*t+a[1];
	m.a=6*a[3]*t+2*a[2];
	return m;
}
```

3차 곡선 계획법

```
void main()
{
    FILE*fp=fopen(FILE_NAME,"w");;
    int ps=0,pf=1000,i=0;
    double tf=1.0,ts=0.001,t=0.0;
    double a[4];
    motion m[1000];
    coeff(ps,pf,tf,a);
    while(t<tf)
    {
        m[i]=cubic(t,a);
        printf("%d₩t%f₩n",i,m[i].p);
        fprintf(fp,"%d ₩t%f ₩t%f ₩t%f ₩n",i,m[i].p,m[i].v,m[i].a);
        i=i+1;
        t=ts*i;
    }
}
```







Fifth-Order Polynomial Trajectory Planning

- Trajectory

$$\theta(t) = c_0 + c_1 t + c_2 t^2 + c_3 t^3 + c_4 t^4 + c_5 t^5$$

$$\dot{\theta}(t) = c_1 + 2c_2 t + 3c_3 t^2$$

$$\ddot{\theta}(t) = 2c_2 + 6c_3 t + 12c_4 t^2 + 20c_5 t^3$$

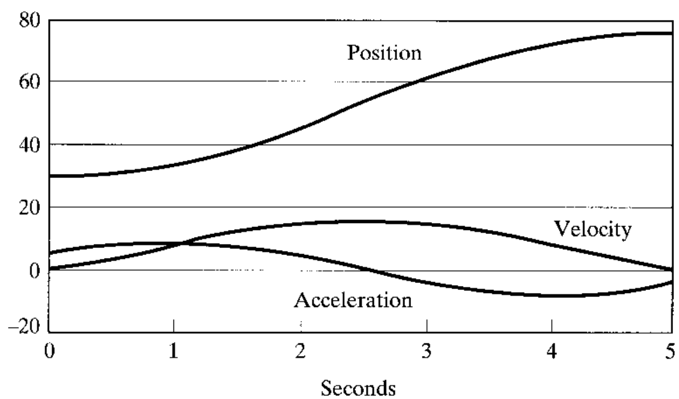
- Initial / final conditions

$$\theta(t_i) = \theta_i \qquad \theta(t_f) = \theta_f$$

$$\dot{\theta}(t_i) = 0 \qquad \dot{\theta}(t_f) = 0$$

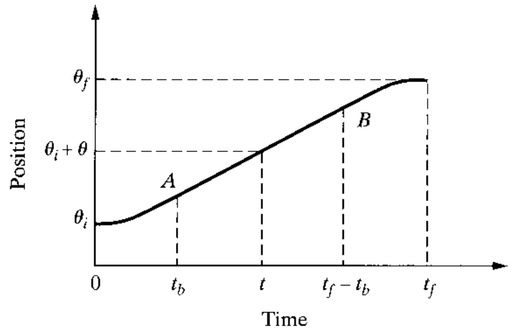
$$\ddot{\theta}(t_i) = \ddot{\theta}_i \qquad \ddot{\theta}(t_f) = \ddot{\theta}_f$$

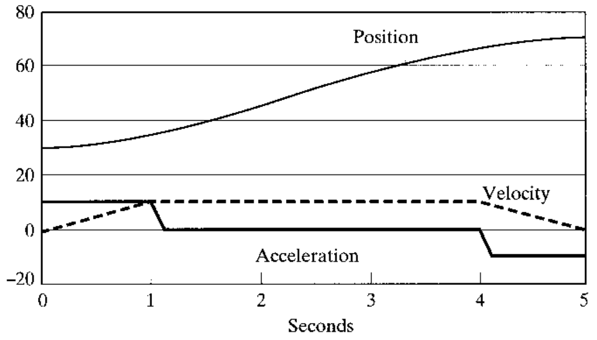
$\Rightarrow$ c_0 , c_1 , c_2 , c_3, c_4, c_5



Linear Segments with Parabolic Blends

- Trajectory





- Initial / final conditions

$$\theta(0) = \theta_i \qquad \theta(t_f) = \theta_f$$

$$\dot{\theta}(0) = 0 \qquad \dot{\theta}(t_f) = 0$$

$$\Rightarrow \quad t_b \text{ , } a \text{ (or } \omega)$$

- Constraints

$$\dot{\theta}_{\max} = \omega \quad (\text{or} \quad \ddot{\theta}_{\max} = a)$$

$$t_b = \frac{\theta_i - \theta_f + \omega \cdot t_f}{\omega} \qquad a = \frac{\omega}{t_b}$$

Linear Segments with Parabolic Blends

(Example) $\theta_i = 30$(deg), $\theta_f = 70$(deg), $t_f = 5$(sec), ω=10(deg/s)

$\rightarrow t_b$

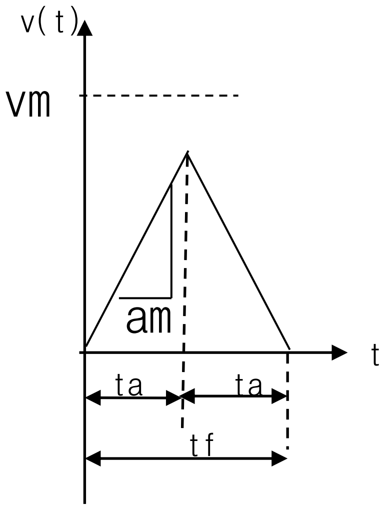
(Example) $\theta_i = 30$(deg), $\theta_f = 70$(deg), $a = 5$(deg/s^2), ω=10(deg/s)

$\rightarrow t_f$

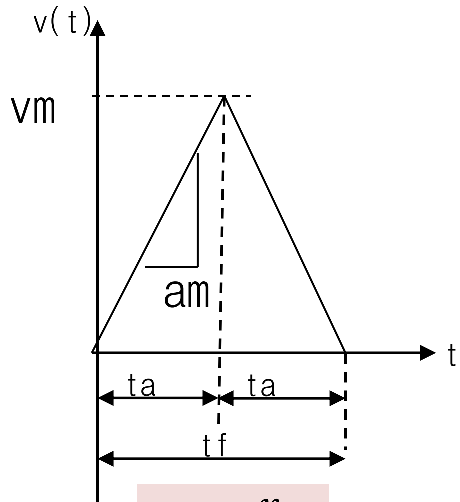
사다리꼴 속도계획법

- 이동해야할 거리의 크기($|p_f - p_s|$)에 따라, 최대속도가 달라진다

$|p_f - p_s| < \frac{v_m^2}{a_m}$ 일 때



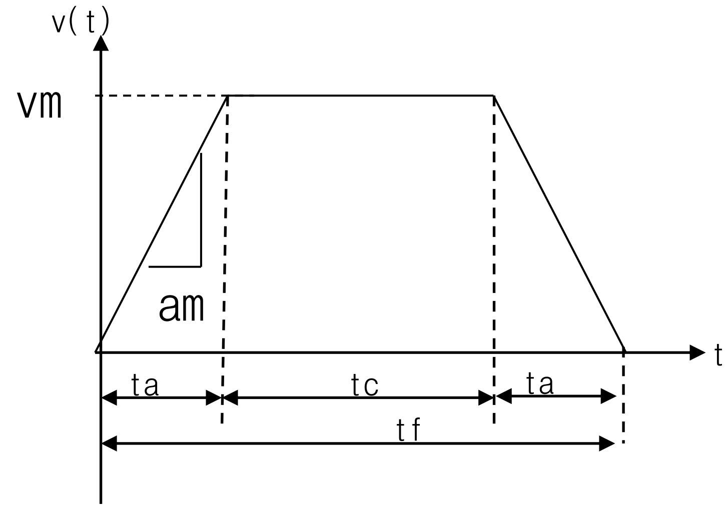
$|p_f - p_s| = \frac{v_m^2}{a_m}$ 일 때



$$a_m = \frac{v_m}{t_a}$$

Case1: 등속구간 없음(tc=0)

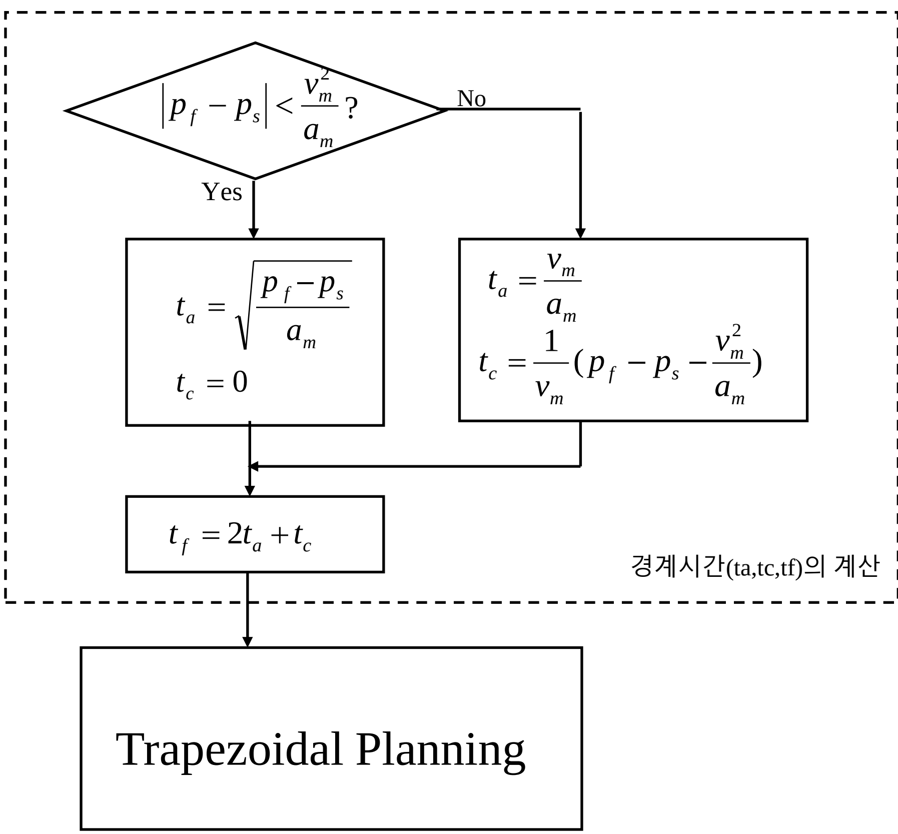
$|p_f - p_s| > \frac{v_m^2}{a_m}$ 일 때



Case2:등속구간 있음(tc>0)

사다리꼴 속도계획법

- 알고리즘(단계1): 경계시간의 계산
  - 주어진 거리 $|p_f - p_s|$에 따라, 경계시간을 먼저 계산한다

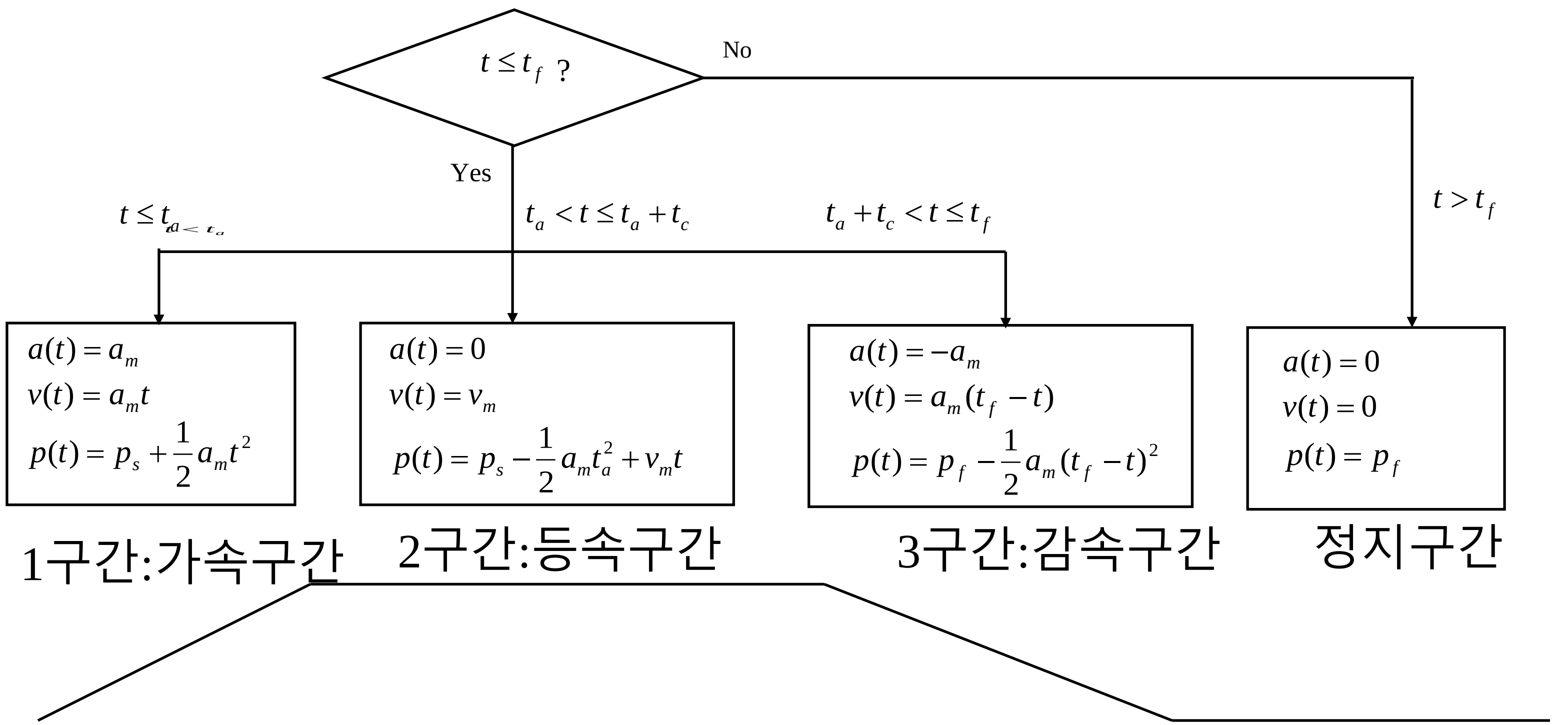


사다리꼴 속도계획법

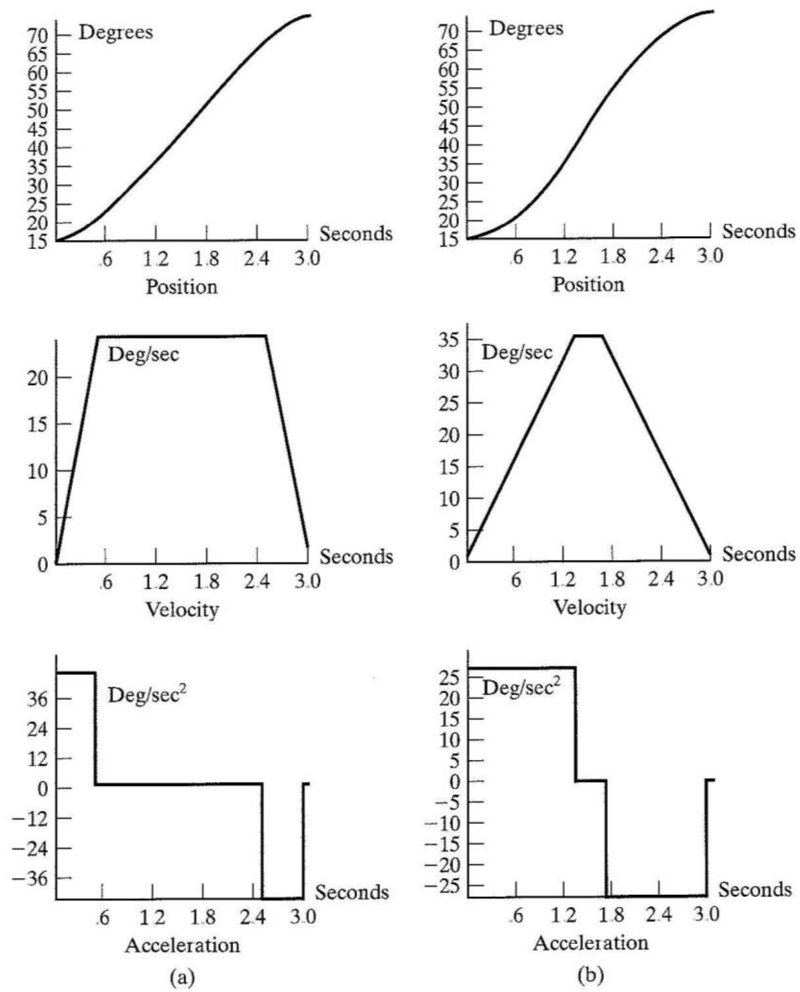
- 알고리즘(단계2) : 구간별 속도,위치 계산
  - 구간마다, 속도/위치 를 계산한다

$$v(t) = v_0 + at$$

$$p(t) = p_0 + v_o t + \frac{1}{2}at^2$$



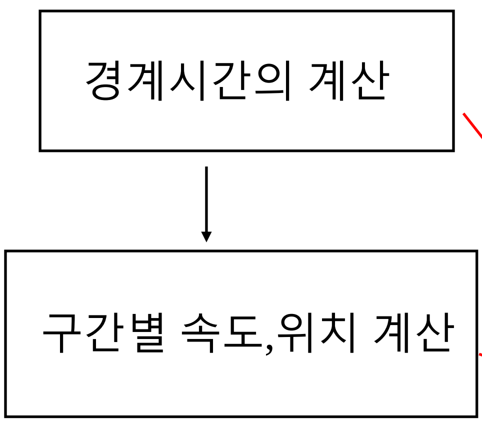
사다리꼴 속도계획법



Degrees
70
65
60
55
50
45
40
35
30
25
20
15
Seconds
.6 1.2 1.8 2.4 3.0
Position
Deg/sec
20
15
10
5
0
Velocity
Deg/sec²
36
24
12
0
−12
−24
−36
Acceleration
(a)
Deg/sec
35
30
25
Deg/sec²
−5
−10
−15
−20
−25
(b)

사다리꼴 속도계획법

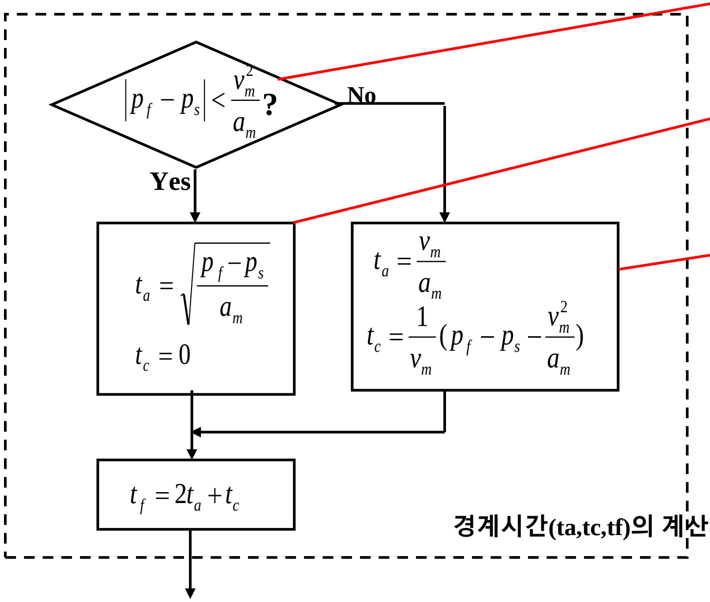
시작점의 위치/속도가 (0,0)이고, 최종점의 위치/속도가 (1500,0)일 때(단위 엔코더 펄스), 최대가속도가 1000 pulse/sec^{2}이고 최대속도가 1000pulses/sec일 때, 사다리꼴 계획법에 의한 운동계획을 0.001초마다 구하는 컴퓨터 프로그램을 C언어를 이용하여 작성하라.




```
#include<stdio.h>
#include<math.h>
#define FILE_NAME "traipzoidal.xls"
double vm=1000.0,am=1000.0;
double ta,tc,tf,ts=0.001,ps=0,pf=1500;
struct motion
{
    double a;
    double v;
    double p;
};
void coeff();
motion traipzoidal(double t);
void main()
{
    FILE*fp=fopen(FILE_NAME,"w");;
    int i=0;
    double t=0.0;
    motion m[3000];
    coeff();
    while(t<=tf)
    {
        m[i]=traipzoidal(t);
        printf("%d₩t%f₩t%f₩n",i,m[i].p,m[i].v);
        fprintf(fp,"%d ₩t%f ₩t%f ₩t%f ₩n",i,m[i].p,m[i].v,m[i].a);
        i=i+1;
        t=ts*i;
    }
}
```

사다리꼴계획법

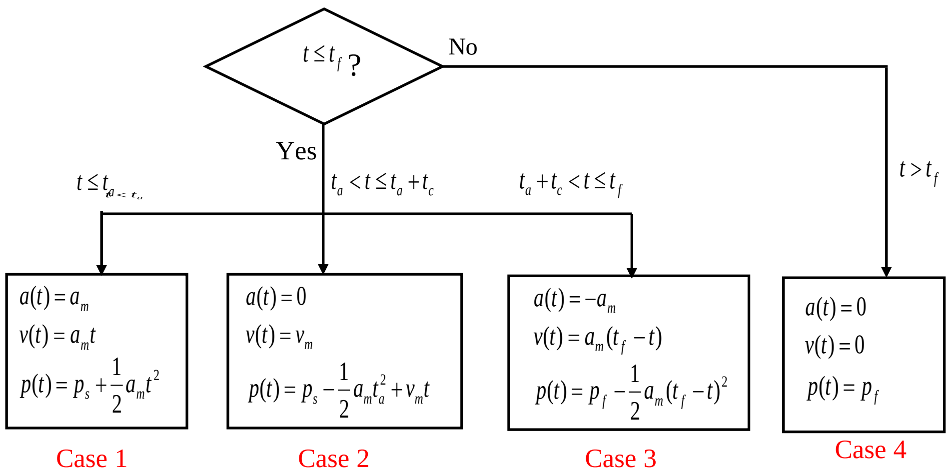
함수 coeff() : 경계시간의 계산




```
void coeff()
{
    if((pf-ps)<=vm*vm/am)
    {
        ta=sqrt((pf-ps)/am);
        tc=0.0;
    }
    else
    {
        ta=vm/am;
        tc=(pf-ps-vm*vm/am)/vm;
    }
    tf=tc+ta*2;
}
```

사다리꼴 속도계획법

함수 trapezoidal() : 구간별 속도/위치 계산

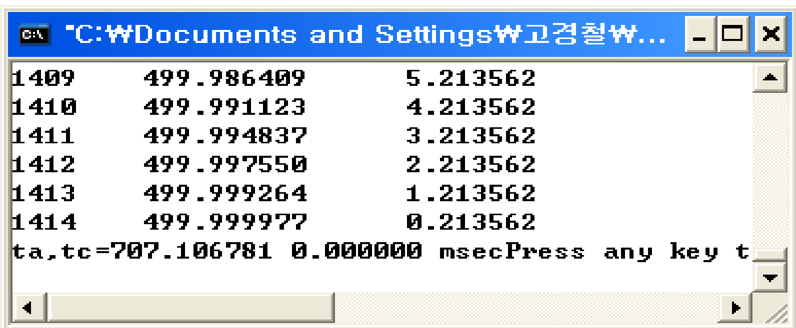


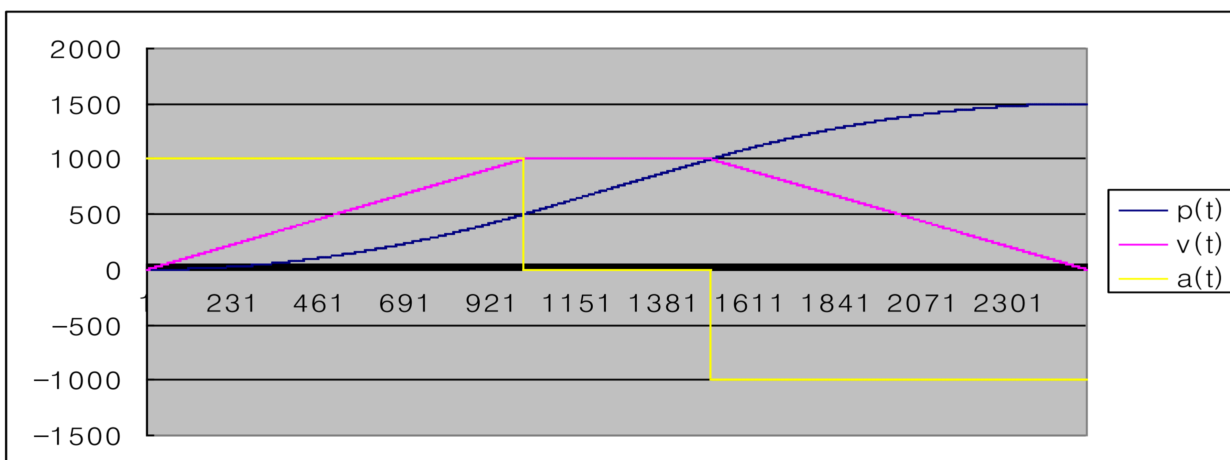

```
motion trapezoidal(double t)
{
    motion m;
    if(t<=ta)                          Case 1
    {
        m.a=am;
        m.v=am*t;
        m.p=ps+am*t*t/2.0;
    }
    else if(t<=tc+ta)                  Case 2
    {
        m.a=0;
        m.v=vm;
        m.p=ps-am*ta*ta/2.0+vm*t;
    }
    else if(t<tf)                      Case 3
    {
        m.a=-am;
        m.v=am*(tf-t);
        m.p=pf-am*(tf-t)*(tf-t)/2.0;
    }
    else                               Case 4
    {
        m.a=0;
        m.v=0;
        m.p=pf;
    }
    return m;
}
```

사다리꼴 속도계획법

시작점의 위치/속도가 (0,0)이고, 최종점의 위치/속도가 (1500,0)일 때

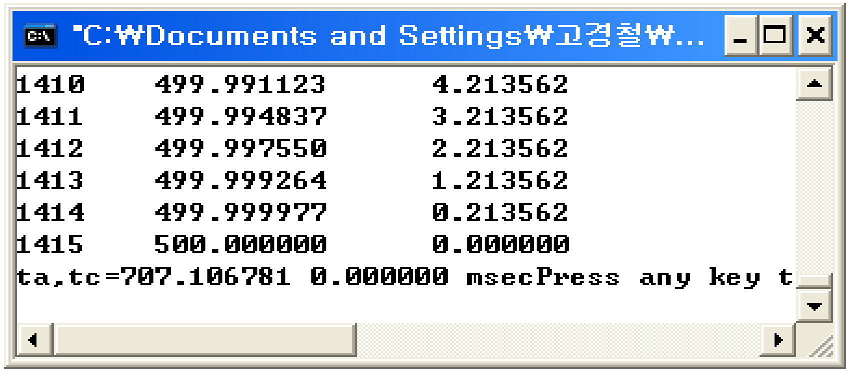
실행결과:

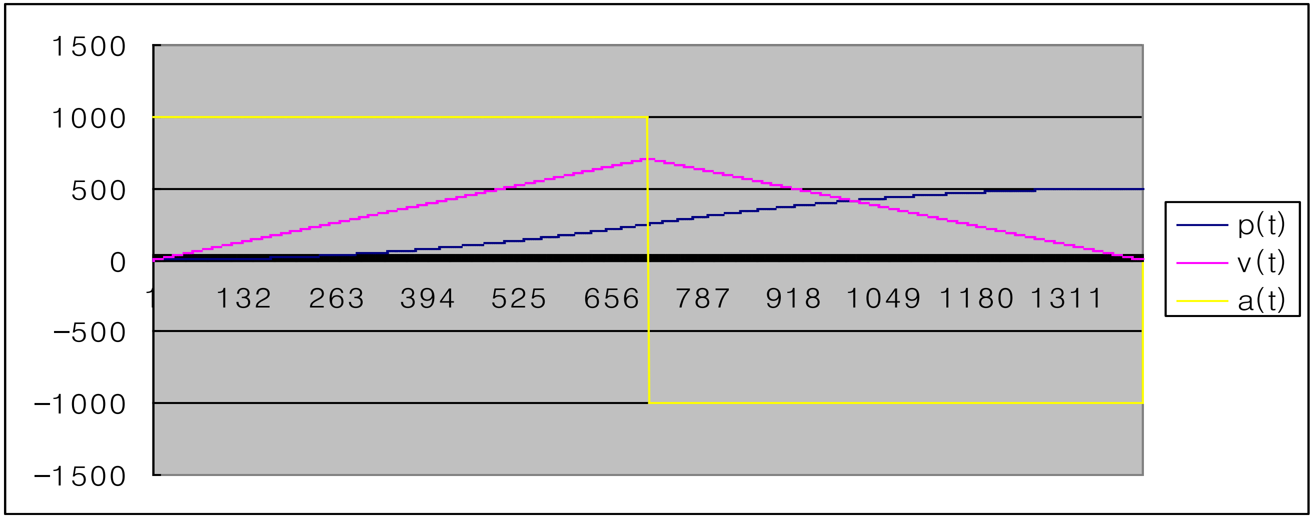




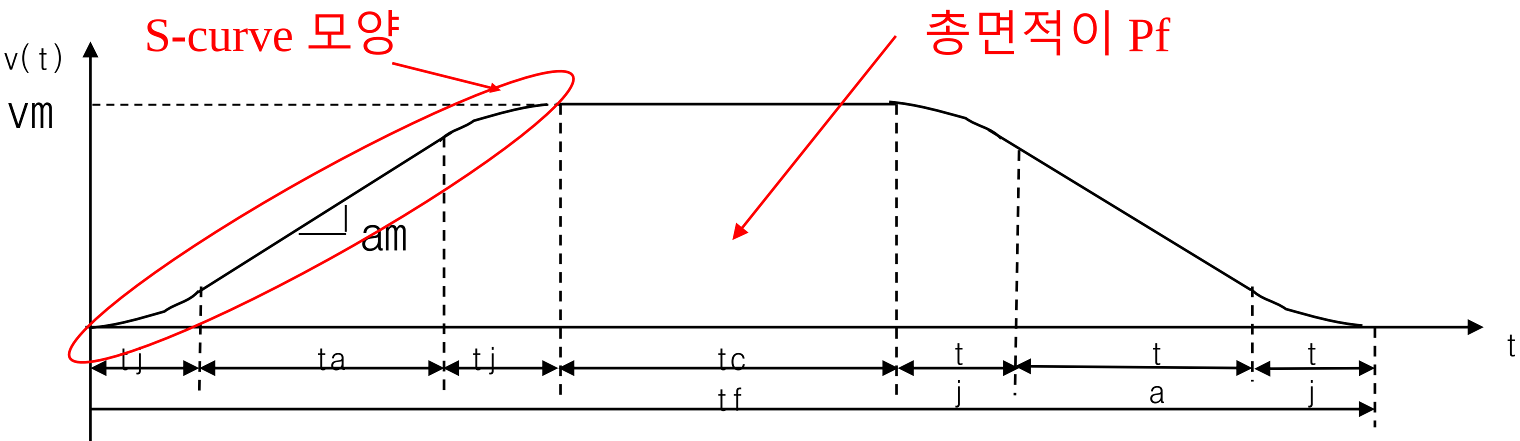
사다리꼴 속도계획법

시작점의 위치/속도가 (0,0)이고, 최종점의 위치/속도가 (500,0)일 때 실행결과:

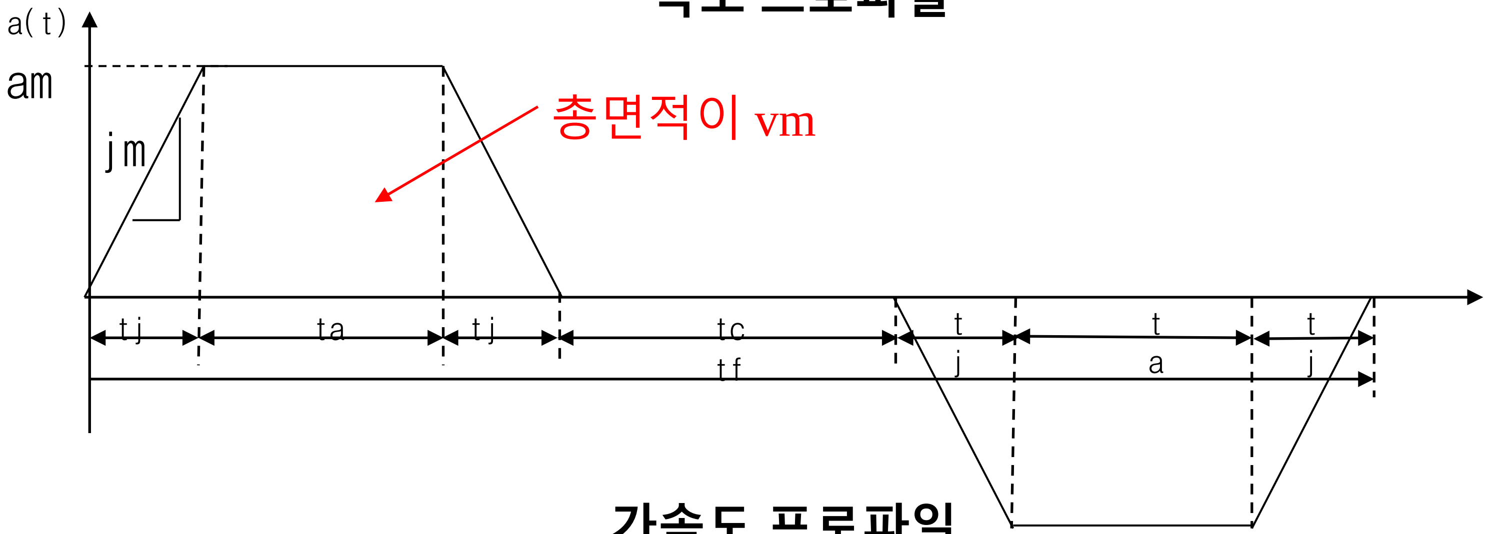




가감속 S-곡선 사다리꼴 속도계획



속도 프로파일



가속도 프로파일

Higher Order Trajectories

$$\theta(t) = c_0 + c_1 t + c_2 t^2 + c_3 t^3 + \cdots\cdots + c_{n-1} t^{n-1} + c_n t^n$$

- Via points 고려
- 가속도 조건 고려

– 4-3-4 trajectory

$$\theta(t)_1 = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 \quad (0 \le t \le \tau_{1f})$$

$$\theta(t)_2 = b_0 + b_1 t + b_2 t^2 + b_3 t^3 \quad (0 \le t \le \tau_{2f})$$

$$\theta(t)_3 = c_0 + c_1 t + c_2 t^2 + c_3 t^3 + c_4 t^4 \quad (0 \le t \le \tau_{3f})$$

1) Initial position
2) Initial velocity
3) Initial acceleration
4) Position of 1st via point
5) Position continuity at 1st via point
6) Velocity continuity at 1st via point
7) Acceleration continuity at 1st via point
8) Position of 2nd via point
9) Position continuity at 2nd via point
10) Velocity continuity at 2nd via point
11) Acceleration continuity at 1st via point
12) Final position
13) Final velocity
14) Final acceleration

Higher Order Trajectories

$$
\begin{bmatrix} \theta_1 \\ \dot{\theta}_1 \\ \ddot{\theta}_1 \\ \theta_2 \\ \theta_2 \\ 0 \\ 0 \\ \theta_3 \\ \theta_3 \\ 0 \\ 0 \\ \theta_4 \\ \dot{\theta}_4 \\ \ddot{\theta}_4 \end{bmatrix}
=
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & \tau_{1f} & \tau_{1f}^{2} & \tau_{1f}^{3} & \tau_{1f}^{4} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 2\tau_{1f} & 3\tau_{1f}^{2} & 4\tau_{1f}^{3} & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 2 & 6\tau_{1f} & 12\tau_{1f}^{2} & 0 & 0 & -2 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & \tau_{2f} & \tau_{2f}^{2} & \tau_{2f}^{3} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 2\tau_{2f} & 3\tau_{2f}^{2} & 0 & -1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 6\tau_{2f} & 0 & 0 & -2 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \tau_{3f} & \tau_{3f}^{2} & \tau_{3f}^{3} & \tau_{3f}^{4} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2\tau_{3f} & 3\tau_{3f}^{2} & 4\tau_{3f}^{3} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 6\tau_{3f} & 12\tau_{3f}^{2}
\end{bmatrix}
\times
\begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ b_0 \\ b_1 \\ b_2 \\ b_3 \\ c_0 \\ c_1 \\ c_2 \\ c_3 \\ c_4 \end{bmatrix}
$$

$$[\theta] = [M][C] \qquad [C] = [M]^{-1}[\theta]$$

Higher Order Trajectories

Example 5.5

Conditions

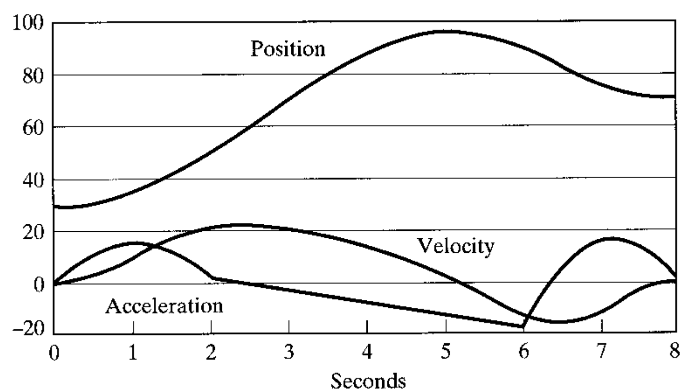
$$\theta_1 = 30^\circ, \quad \dot{\theta}_1 = 0, \quad \ddot{\theta}_1 = 0, \quad \tau_{1i} = 0, \quad \tau_{1f} = 2,$$
$$\theta_2 = 50^\circ, \quad \tau_{2i} = 0, \quad \tau_{2f} = 4,$$
$$\theta_3 = 90^\circ, \quad \tau_{3i} = 0, \quad \tau_{3f} = 2,$$
$$\tau_4 = 70^\circ, \quad \dot{\theta}_4 = 0, \quad \ddot{\theta}_4 = 0.$$

Coefficients

$$a_0 = 30, \quad b_0 = 50, \quad c_0 = 90,$$
$$a_1 = 0, \quad b_1 = 20.477, \quad c_1 = -13.81,$$
$$a_2 = 0, \quad b_2 = 0.714, \quad c_2 = -9.286,$$
$$a_3 = 4.881, \quad b_3 = -0.833, \quad c_3 = 9.643,$$
$$a_4 = -1.191, \quad c_4 = -2.024.$$

4-3-4 Trajectory

$$\theta(t)_1 = 30 + 4.881t^3 - 1.191t^4, \qquad 0 < t \le 2,$$
$$\theta(t)_2 = 50 + 20.477t + 0.714t^2 - 0.833t^3, \qquad 0 < t \le 4,$$
$$\theta(t)_3 = 90 - 13.81t - 9.286t^2 + 9.643t^3 - 2.024t^4, \qquad 0 < t \le 2.$$



Cartesian-Space Trajectory Planning

S1. Increment the time $t = t + \Delta t$

S2. Calculate the position and orientation of hand

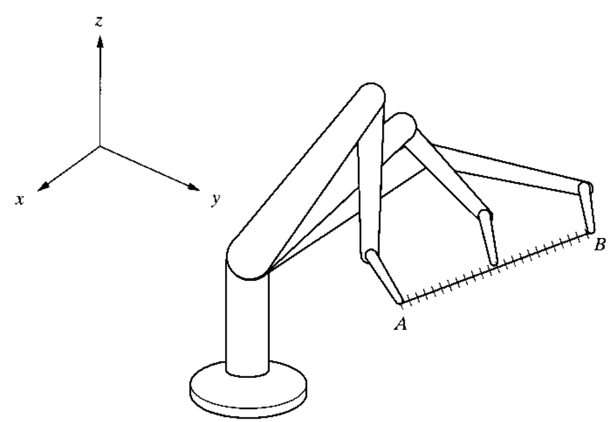
$$P = P(t), \qquad R = R(t)$$

S3. Calculate joint values : inverse kinematics

$$\theta_1, \dots , \theta_n$$

S4. Send the joint values to joint (motor) controller

S5. Go to S1

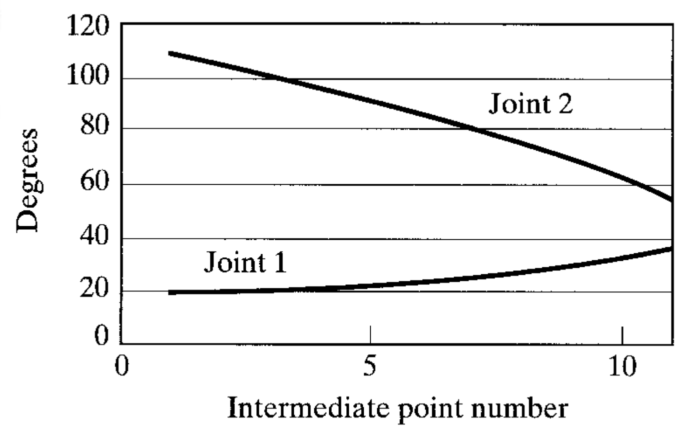


Cartesian-Space Trajectory Planning

(Example 5.6) 2 d.o.f robot

TABLE 5.1 THE COORDINATES AND THE JOINT ANGLES FOR EXAMPLE 5.6.

#	x	y	θ_1	θ_2
1	3	10	18.8	109
2	3.5	10.4	19	104.0
3	4	10.8	19.5	100.4
4	4.5	11.2	20.2	95.8
5	5	11.6	21.3	90.9
6	5.5	12	22.5	85.7
7	6	12.4	24.1	80.1
8	6.5	12.8	26	74.2
9	7	13.2	28.2	67.8
10	7.5	13.6	30.8	60.7
11	8	14	33.9	52.8

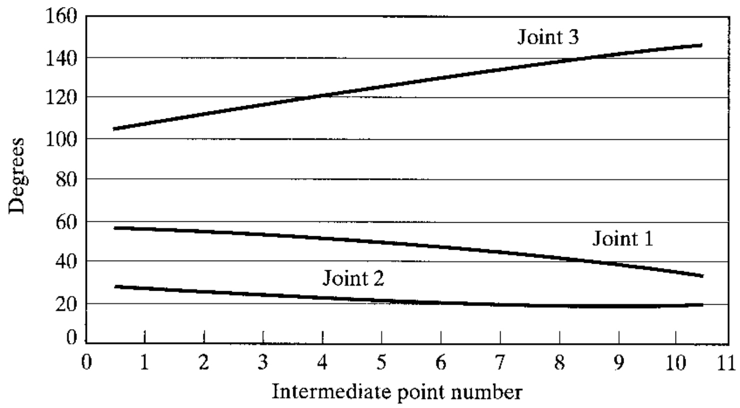


Cartesian-Space Trajectory Planning

(Example 5.7) 3 d.o.f robot

TABLE 5.2 THE HAND-FRAME COORDINATES AND JOINT ANGLES FOR THE ROBOT OF EXAMPLE 5.7

X	Y	Z	θ_1	θ_2	θ_3
9	6	10	56.3	104.7	27.2
8.4	5.9	9.8	54.9	109.2	25.4
7.8	5.8	9.6	53.4	113.6	23.8
7.2	5.7	9.4	51.6	117.9	22.4
6.6	5.6	9.2	49.7	121.9	21.2
6	5.5	9	47.5	125.8	20.1
5.4	5.4	8.8	45	129.5	19.3
4.8	5.3	8.6	42.2	133	18.7
4.2	5.2	8.4	38.9	136.3	18.4
3.6	5.1	8.2	35.2	139.4	18.5
3	5	8	31	142.2	18.9



Cartesian-Space Trajectory Planning

- Cartesian space planning 시 고려사항

1. 많은 계산량: inverse kinematics
2. Intermediate points unreachable
3. Sudden joint-angle change around sigularities