

뱅뱅 제어법을 변형한 중간 경로 제동이 가능한
최단시간 제어기의 개발

A Study on the Trajectory Controllable Minimum-Time Controller
Using Modified Bang-Bang Control Law

이 현 오 양 우 석

홍익대학교 전기공학과 (Tel: 60-2376; Fax: 867-6146; E-mail: wsyang@wow.hongik.ac.kr)

Abstracts Bang-bang control law provides the optimal solution for a minimum-time control problem, but ignores the intermediate path except for the initial and final points. In this paper, a near minimum-time suboptimal fuzzy logic controller is introduced that can control the intermediate path.

A dynamic model for a system is established using the average dynamics method of linearization. System model is continuously updated over the control time periods. This makes it suitable for high speed or variable payload applications. Bang-bang control theory is modified and used to derive the preliminary control law. A fuzzy logic algorithm is then applied to adjust and find the best solution. The solution will provide the suboptimal minimum-time control law which can avoid obstacles in the workspace.

Keywords trajectory controllable, minimum-time controller, fuzzy logic, average dynamics

1. 서론

하나의 구간을 최단 시간 내에 이동하고자 할 경우 흔히들 bang-bang 제어가 사용된다. 대상 system의 dynamics가 주어 진다면 bang-bang 제어는 최적의 최단시간제어를 위한 해답을 제공한다. 그러나 일반적으로 system dynamics는 상태(state)들이 서로 couple된 비선형방정식으로 이루어져 bang-bang 제어를 위한 switching curve를 closed form으로 산출하기가 어렵다. 더욱이 bang-bang 제어는 시작점과 최종위치만 관리를 하는 제어방식으로서 이동경로의 제어를 필요로 하는 경우에는 거의 적용이 되지 않고 있다.

이동경로 제어가 필요한 일반적인 제어 방법은 off-line으로 설계된 경로(Trajectory)를 따라서 on-line상에서 경로조종(Path Tracking)을 하는 것이다. 이 경우, 일단, 시간의 함수로 구성된 계획된 최적의 trajectory를 구하면 현재 많은 연구가 이루어진 여러 가지 on-line path tracking기법을 사용하여 쉽게 제어를 할 수 있다.

최적의 trajectory란 task 측면뿐만 아니라 제작시 설계된 system의 능력까지도 최고로 발휘되도록 설계가 되어야 한다. 그러나 불행히도 system dynamics가 고려된 최적의 trajectory 설계방법은 아직까지 나타나지 않고 있다. 운전시 최고 운전가능 속도 및 가속도는 순간위치, 부하 등, 여러 가지 조건에 따라 변한다. Trajectory를 설계할 경우에 이들 제약조건을 일일이 미리 분석하여 설계하기란 매우 어렵다. 따라서 trajectory는 최악의 경우를 대비한 속도 및 가속도 등을 기준으로(Global Least Upper Bound) 설계하게 된다. 이 경우 system은 최악의 상황을 고려하여 미리 예정된 속도 및 가속도에 근거하여 미리 계획된 trajectory를 따라 움직이게 된다. 이는 system의 실제

운전능력에 비하여 매우 낮은 기준인바, system의 기능을 최대한 발휘하기 어렵게 설계가 된다.

본 연구는 system을 최단 시간 내에 원하는 위치로 이동시키기 위한 Minimum-Time Suboptimal Controller를 개발하고자 하는 것이다. 즉 중간경로의 제동이 가능하고 최적의 시간제어가 가능한 제어기 알고리즘을 개발하고자 하는 것이다.

실시간 제어에 응용될 수 있는 새로운 dynamics 선형화 방법의 개발과 함께 bang_bang 제어법을 수정하여 plant의 중간 경로에 대한 제동을 가능하게 하며, 최단 시간 내에 Plant의 상태를 원하는 위치에 이동시킨다. 이러한 요소기술의 개발은 최적의 시간제어가 필요한 HDD의 제어분야 뿐만 아니라 무인반송차의 모델링과 동작제어, 각종 센서와 장비와의 결합, 2개 이상의 로봇트를 동시에 사용할 경우의 로봇트제어 등 자동화 관련 여러 분야의 이론적인 요소기술로도 활용이 될 것이다.

2. 최단시간 제어알고리즘

2.1 System dynamics의 선형화 : Average dynamics

Average dynamics에 의한 dynamics 선형화(linearization)의 기본 개념은 현재(current) state의 system dynamics와 최종 목표 state의 dynamics를 산술 평균하여 선형화 시켜 현재의 system dynamics로 사용하고자 하는 것이다. 다른 선형화 방법과 마찬가지로 average dynamics도 시간이 흘러 목표 state에 system이 근접할수록 실제 dynamics와 가까워진다. Average dynamics는 물론 실제의 dynamics와는 많은 차이가 있다. 그러나 본 연구에서 사용하고자 하는 기본 제어법은 bang-bang 제어이다. Bang-bang 제어에서는 Switching curve를 이용하여 적용할 제어값의 최고 혹은 최저를 구한다. 이 경우 dynamics는 제어값의 부호를 제대로 구할 수 있는 정도의 정밀도만 갖

본 연구는 한국 전력 공사의 지원에 의하여 기초 전력 공학 공동 연구소 주관으로 수행되었음.

추면 충분하다.

system의 freedom을 n 이라 하고 \mathbf{x} 를 system의 각 위치 변수를 나타내는 $n \times 1$ vector라고 정의하자. system dynamics는 아래와 같이 표현된다

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \dot{\mathbf{x}}, \mathbf{u}, t) \quad (1)$$

\mathbf{x} 의 i 번째 원소를 x_i 라 하자. $\dot{\mathbf{x}}$ 와 $\ddot{\mathbf{x}}$ 는 각각 속도와 가속도를 의미하는 $n \times 1$ vector들이다. \mathbf{u} 는 $m \times 1$ control vector이다. 우리는 임의의 시간 t 에서 해당 변수 이외의 모든 변수에 대한 실험데이터를 대입하여 (1)의 방정식을 다음과 같이 정리할 수 있다.

$$\ddot{x}_i = f_i U_i + g_i, \quad \text{for } i = 1, 2, \dots, n \quad (2)$$

f_i 와 g_i 는 scalar 값을 갖게된다. Average dynamics에 의한 선형화된 dynamics는 아래와 같은 식으로 정의한다.

$$\ddot{x}_i = \alpha_i U_i + \beta_i \quad \text{for } i^{\text{th}} \text{ state} \quad (3)$$

$$\alpha_i = \frac{\hat{a}_i + a_{if}}{2} \quad (4)$$

$$\beta_i = \frac{\hat{\beta}_i + \beta_{if}}{2} \quad (5)$$

α_i 와 β_i 를 구하기 위하여 우선 \hat{a}_i 와 $\hat{\beta}_i$ 가 식 (3), (4)에 의하여 구하여진다. \hat{a}_i 와 $\hat{\beta}_i$ 는 현재의 state와 $U(t-\sigma)$ 를 이용하여 구한 f_i 와 g_i 이다. 여기서 t 는 현재 시간이며, σ 는 sampling interval을 의미한다. 결국 \hat{a}_i 와 $\hat{\beta}_i$ 는 현시점에서의 f_i 와 g_i 의 근사치라 할 수 있다. a_{if} and β_{if} 는 최종 목표 state를 대입한 f_i 와 g_i 이다.

2.2 Bang-bang 최단시간 제어 알고리즘

우선 average dynamics를 이용한 일반적인 최단시간 제어 알고리즘을 고찰하자. 최단시간 제어 문제는 i 번째의 변수($i = 1, 2, \dots, n$)를 초기값(x_{i0}, \dot{x}_{i0})에서 최종값(x_{if}, \dot{x}_{if})으로 이동시키기 위한 다음의 dynamics를 만족시키는 제어 U_i 를 구하는 것이다.

$$\ddot{x}_i = \alpha_i U_i + \beta_i \quad (6)$$

이때 제어 U_i 는 $|U_i| \leq U_{iMAX}$ 조건을 갖게된다.

System variable에 대하여 state를 $\mathbf{x}_i = (x_{1i}, x_{2i})^T$ 라고 하면, Bang-bang 제어에 의한 최소시간제어를 위한 해답은 다음과 같이 구하여진다.

$$U_i^* = \begin{cases} U_{iMAX} & \text{if } S_i(\mathbf{x}_i) < 0 \\ -U_{iMAX} & \text{if } S_i(\mathbf{x}_i) > 0 \end{cases} \quad (7)$$

여기서

$$S_i(\mathbf{x}_i) = \begin{cases} x_{1i} - x_{2i}^2/(2a_i^-) & \text{if } x_{2i} \geq 0 \\ x_{1i} - x_{2i}^2/(2a_i^+) & \text{if } x_{2i} \leq 0 \end{cases} \quad (8)$$

$$a_i^- = -\alpha_i U_{iMAX} + \beta_i \quad (9)$$

$$a_i^+ = \alpha_i U_{iMAX} + \beta_i \quad (10)$$

이다. 또한 이 결과 나타나는 trajectory는 다음의 식으로 나타난다.

$$x_{1i} = \begin{cases} x_{2i}^2/(2a_i^-) + C_i^- & \text{if } U_i = -U_{iMAX} \\ x_{2i}^2/(2a_i^+) + C_i^+ & \text{if } U_i = U_{iMAX} \end{cases} \quad (11)$$

C_i^+ 와 C_i^- 는 초기값과 관련이 있는 상수이다. 초기 state \mathbf{x}_{i0} 로부터 최종목표 state \mathbf{x}_{if} 로의 이동을 완료하기 위한 수행시간

(execution time) T_i 는 다음식 (12)에 의하여 구하여진다.

$$T_i = \begin{cases} K^-(1/a_i^- - 1/a_i^+) - x_{20}/a_i^- & \text{if } S_i(\mathbf{x}_i) > 0 \\ K^+(1/a_i^+ - 1/a_i^-) - x_{20}/a_i^+ & \text{if } S_i(\mathbf{x}_i) < 0 \end{cases} \quad (12)$$

여기서

$$K^- = -\{2[x_{10} - x_{20}^2/(2a_i^-)] [a_i^- a_i^+ / (a_i^- - a_i^+)]\}^{1/2}$$

$$K^+ = \{2[x_{10} - x_{20}^2/(2a_i^+)] [a_i^+ a_i^- / (a_i^+ - a_i^-)]\}^{1/2}$$

이다.

2.3 Control Law

Bang-Bang 제어법은 최단시간 제어를 위한 일반적인 해결책이다. 최적제어를 위한 제어값은 switching 곡선에 의해 결정되는 sign값을 부호로 하는 최대의 제어 가능치를 갖게 된다.

그림 1은 초기 state \mathbf{x}_{i0} 에 대한 최종 목표 state \mathbf{x}_{if} 를 기준으로 하는 perturbation model에서의 bang-Bang 제어에 의한 trajectory를 나타낸다. State plane상에서의 trajectory는 A-B-O로 나타난다. System variable은 초기값에 따라 그림 1.1의 경우에는 a-â-a-b-o, 그림 1.2의 경우에는 a-b-o로 변한다.

만약 모든 state의 초기 상태 \mathbf{x}_{i0} 가 그림 1.2의 perturbation model에서와 같이 x_{1i} 축 위에 존재한다면 각 state의 trajectory는 state plane상에서 그림 1.2의 A-B-O와 유사한 형태를 가질 것이다. 이 경우 그림 1.1의 경우의 a-â-a-b-o 처럼 궤도의 재추적(retrace)은 발생할 수 없게 된다. 더욱이 각 state 이동의 실행 시간이 같아진다면, 제어의 결과로 생기는 state variable position은 거의 직선의 형태를 갖게 될 것이다. 이러한 경우 장애물 회피를 위한 bang-bang 제어[6] 까지도 가능하게 된다. 만약에 무인반송차 위에 놓여진 로봇과 반송차를 동시에 제어한다고 할 경우도 각 joint의 실행시간을 조정함으로써 state를 원하는 값 즉, $\mathbf{x}_f(t) = \mathbf{x}(t)$ 와 $\mathbf{v}(t) = \dot{\mathbf{x}}(t)$ 으로의 제어가 가능하게 될 것이다.

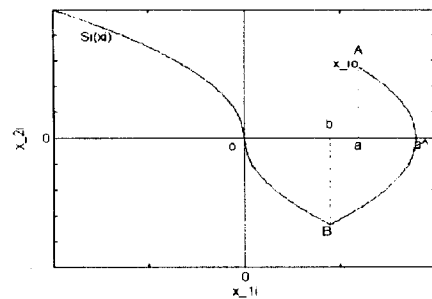


그림 1.1 Case 1

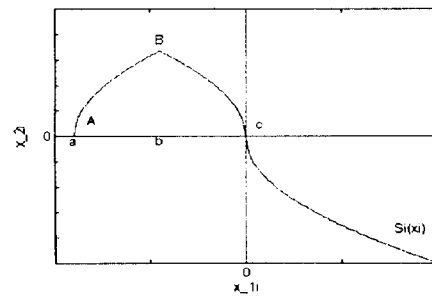


그림 1.2 Case 2

그림 1. 상태변수의 궤적

작업 실행시간은 n개의 position state의 실행시간 중 가장 늦게 완료되는 시간이라 할 수 있다. 동시에 작업을 끝마치기 위한 최단시간은 모든 실행시간 $T_i, i = 1, 2, \dots, n$, 중의 최대값 T_{MAX} 에 의하여 결정된다.

$$T_{MAX} = \text{MAX}(T_1, T_2, \dots, T_n) \quad (13)$$

T_i 는 식 (14)에서 구하여진 i번째 position variable에 대한 일반적인 bang-bang 제어를 할 경우 소요되는 실행시간을 의미한다. 각 state는 $T_{MAX} - T_i$ 의 여유작업시간을 갖고 있다. U_i 를 조절하면 그림 1의 switching curve를 변화시켜 T_i 와 trajectory를 변화하게 한다. T_i 는 U_i 를 $\pm U_{MAX}$ 로 제어하였을 때의 결과이다. $|U_i|$ 의 최대치를 U_{MAX} 에서 U_i^* 로 ($U_{imax} > U_i^*$) 적절히 줄인다면 실행시간 T_i^* 도 T_i 와 T_{MAX} 사이의 값으로 늘일 수 있고 각 position variable의 trajectory도 조정이 가능하다. T_i^* 와 U_i^* 는 퍼지 제어로서 보정이 되어 원하는 최적의 제어를 이룩하게 된다.

본 논문에서 소개하고자 하는 개략적인 제어 알고리즘은 아래와 같다.

Step 1. 식(4)과 (5)로부터 α_i 와 β_i 가 구하여진다.

Step 2. 식 (12)을 사용하여 state별 T_i 를 구한다. T_{MAX} 도 동시에 구하여진다.

i번째 state에서 $T_i = T_{MAX}$ 라 하자.

Step 3. $T_i = T_{MAX}$ 로 하는 U_i^* 를 구한다. $i \neq j$ 이다.

Step 4. U_i^* 를 사용하여 새로운 switching curve를 구하여 제어값을 구한다.

Step 5. 퍼지이론을 사용하여 구한 보정치를 Step 4에서 구한 제어값에 더하여 최적의 제어값을 구한다.

Step 6. Step 1에서 step 5를 $\|x_k - x^* \| \cdot \| \dot{x}_k - \dot{x}^* \| \leq \epsilon$ 이 될 때까지 반복한다. ϵ 는 임의로 정한 위치 오류이다.

Bang-bang 제어를 사용할 경우 최종 목표 state 근처에서는 제어입력의 빈번한 switching으로 정확한 최종 목표점에 도달하기 위하여 추가로 많은 시간이 소요된다. 그러므로, 제어기는 end-effector가 최종 목표점 부근에 도달하면 다음의 step으로 마무리된다.

Step 7. 제어는 PID제어로 변환되어 system을 최종 목표 state로 도달시킨다.

2.4 Fuzzy Logic

앞에서 소개한 바와 같이 time-synchronization으로 구해진 제어값은 fuzzy logic에 의하여 보정 된다. 이러한 보정에 의하여 최종목표에 도달하는 위치오차와 시간 지연이 억제된다. 본 논문에서는 그림 2와 같이 state plane 상의 position이 switching curve에 가까운 정도를 판단하여 제어값을 변화시킬 수 있는 퍼지제어 시스템을 사용하였다.

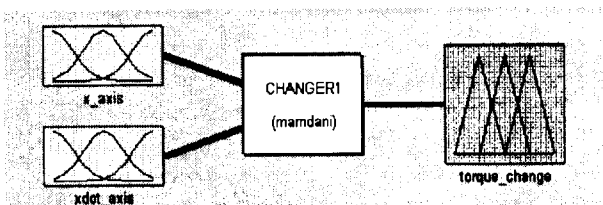


그림 2. Black diagram으로 나타낸 전체 퍼지제어 시스템

그림 2는 전체 퍼지제어 시스템에 대한 개념을 black

diagram을 사용하여 나타낸 것이다. 좌측의 입력은 state들이 switching curve에 가까운 정도를 fuzzify하기 위한 membership function을 나타낸다. 여기서 x_axis는 x 값의 switching curve에의 근접도를 의미하며, xdot_axis는 \dot{x} 값의 switching curve에의 근접도를 나타낸다. 퍼지제어 시스템은 switching curve에 대한 근접도를 입력으로 받아 Mamdani가 제시한 방식으로 결과를 추론한다.[7]

3. 모의실험

본 논문에서 개발된 제어 알고리즘을 점성 유체(viscous fluid) 안에서 움직이는 물체를 모델로 하여 모의실험을 하였다. 모의실험에 사용된 plant의 상태방정식은 다음과 같이 수식화 된다. Plant의 제어변수 u_1 과 u_2 는 x와 y방향으로의 추력을 나타낸다.

$$\begin{aligned} \dot{x}_1(t) &= x_2(t) \\ \dot{x}_2(t) &= -\alpha x_2(t) \sqrt{x_2^2(t) + x_4^2(t)} + u_1(t) \\ \dot{x}_3(t) &= x_4(t) \\ \dot{x}_4(t) &= -\alpha x_4(t) \sqrt{x_2^2(t) + x_4^2(t)} + u_2(t) \end{aligned} \quad (14)$$

그림 3과 그림 4는 각각 모의실험에 사용된 퍼지제어 시스템의 입, 출력 membership function을 보여준다. 수정 시간을 보정하기 위한 fuzzy rule은 다음과 같다.

Rule 1. If (x_axis is close) and (xdot_axis is close) then (torque_change is large)

Rule 2. If (x_axis is near) and (xdot_axis is near) then (torque_change is average)

Rule 3. If (x_axis is fall_away) and (xdot_axis is fall_away) then (torque_change is small)

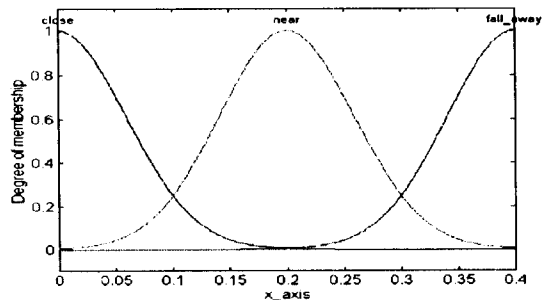


그림 3. 입력의 membership function

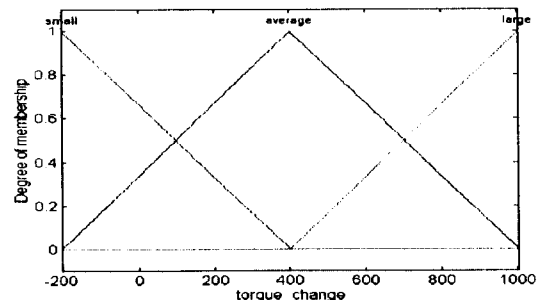


그림 4. 출력의 membership function

앞의 rule은 x 와 \dot{x} 의 위치를 switching curve에 가까운 정도에 따라 fuzzify하여 membership function으로 나타내고 그 값이 작아질수록(switching curve에 가까운 경우) 제어값의 수정치가 커지게 만든 것이다. 이렇게 산출된 추론 결과는 centroid method로 defuzzify 시켰다.

식 (14)에 본 논문에서 제시한 제어 알고리즘과 fuzzy logic을 적용하여 모의 실험한 결과를 그림 5부터 그림 8에 보였다. 그림 5와 그림 6은 각각 x 와 \dot{x} , y 와 \dot{y} 에 대한 상태변수 궤적을 나타낸다. 그림 7은 시간에 따른 최대 제어값의 변화를 보여준다. Bang-bang 제어는 항상 최대값이 제어값으로 되는 방식이다. 그러나 본 알고리즘은 그림에서 보듯이 제어값이 최대값(U_{MAX})과 최소값($-U_{MAX}$) 사이에서 결정되는 것을 볼 수 있다. 그림 8은 xy 평면상에서 제어값에 따라 물체가 움직이는 실제 궤적을 나타내고 있다. 비록 bang-bang 제어에 근거를 두었으나 위치 변수가 선형으로 변하고 있음을 볼 수 있다.

4. 결론

본 논문에서 system을 최단 시간 내에 원하는 위치로 이동시키기 위한 중간경로의 제동이 가능한 제어기 알고리즘이 제시되었다. 주어진 system의 dynamics를 averaging dynamics에 의해 선형화 시키고 선형화된 방정식을 이용하여 실행시간을 계산하여 각 state의 실행 시간을 synchronize하기 위해 계산된 실행 시간을 조정한다. 조정된 실행 시간으로 switching curve를 고려하여 새로운 제어값을 구한다. 구해진 제어값은 퍼지 제어 시스템을 사용하여 보정 된다. 모의실험 결과에서도 알 수 있듯이 수정된 bang-bang control law에 의한 plant의 궤적은 거의 직선으로 나타난다. 각 위치 변수의 실행 시간을 synchronize하여 제어값을 조정함으로써 중간 경로에 대한 제동이 가능하다.

본 논문의 결과를 무인반송차에 적용한다면 생산 현장에서의 무인반송차 효율을 극대화시킬 수 있을 것이다. 무인반송차 및 로봇 등 동작계획을 용이하게 수립할 수도 있다. 또한 본 연구결과는 동적 환경 하에서 움직이는 물체의 동작예측과 함께 다이나믹 시스템 시뮬레이션 시스템과 결합되어, 부품 정렬 시스템, 로봇 조립 시스템 및 보행 로봇 시스템 등의 동작계획의 수립 및 해석 등에 활용될 수 있을 것이다. 본 논문에서 소개한 bang-bang 제어를 수정한 중간 경로의 제동이 가능한 제어법의 개발은 자동화 요소기술로서 각종 생산 시스템의 최적설계에도 응용이 가능할 것이다.

참고문헌

[1] Kirk, *Optimal Control*, Prentice-Hall, New Jersey
 [2] B. K. Kim, K. G. Shin(1985) "Suboptimal Control of Industrial Manipulators with Weighted Minimum Time-Fuel Criterion", *IEEE Trans. on Automatic Control*, Jan. 1985.
 [3] R. P. Paul, *Robot Manipulator -Mechanics, Programming and Control*, MIT Press.
 [4] S. Yoshida, N. Wakabayashi(1992) "A Fuzzy Logic Controller for a Rigid Disk Drive", *IEEE Control System Magazine*, June 1992.
 [5] S. Yoshida, M. ToKura, M. Imura, N. Wakabayashi (1991) "Bang-Bang Seek Control for HDD with Fuzzy Algorithm"

, *IEEE Translate. J. Mag. Japan*, Mar 1991.

[6] S. Yang, J. Jang, "Minimum-Time Control of Robotic System with Obstacles in the Workspace", *Proc. in ASCC*, July 1994.
 [7] N. Gulley, J. R. Jang, *Fuzzy Logic Toolbox*, Math Works, Mass.

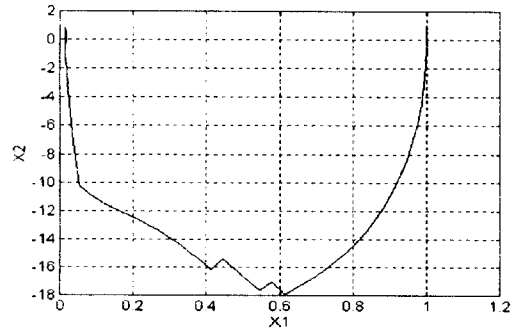


그림 5. x 와 \dot{x} 의 궤적

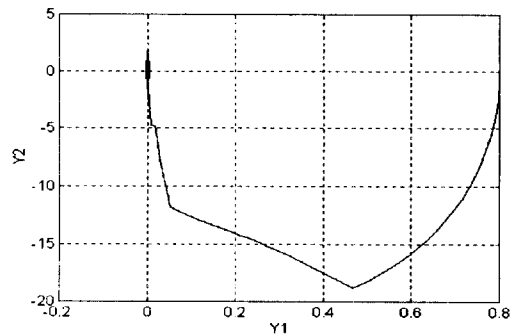


그림 6. y 와 \dot{y} 의 궤적

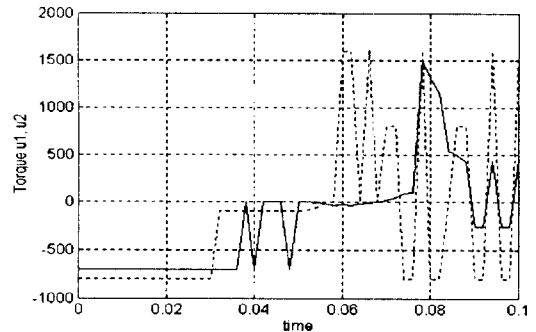


그림 7. 최대 제어값

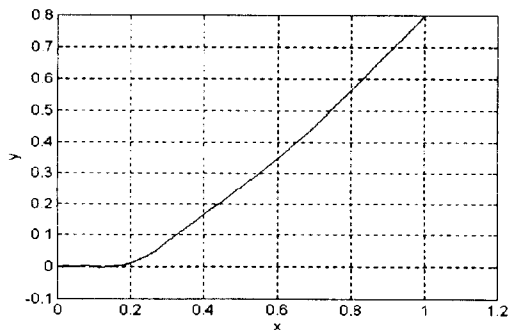


그림 8. x 와 y 의 2차원 평면 궤적