연속형 RHC 에 대한 개선된 구현 알고리즘

Improved Implementation Algorithm for Continuous-time RHC

이 영 삼*, 김 태 신, 김 창 유
(Young Sam Lee, Tae Shin Kim, and Chang You Kim)

Abstract : This paper proposes an improved implementation algorithm for the continuous-time receding horizon control (RHC). The proposed algorithm has a feature that it has better control performance than the existing algorithm. Main idea of the proposed algorithm is that we can approximate the original RHC problem better by assuming the predicted input trajectory on the prediction horizon has a continuous form, which is constructed from linear interpolation of finite number of vectors. This, in turn, leads to improved control performance. We derive a predictor such that it takes linear interpolation into account and proposes the method by which we can express the cost exactly. Through simulation study for an inverted pendulum, we illustrate that the proposed algorithm has the better control performance than the existing one.

Keywords: receding horizon control, implementation algorithm, linear interpolation, predictor

I. 서론

이동구간제어(Receding Horizon Control, RHC)는 개념의 명료성, 입출력 제약조건을 다룰 수 있는 강점, 다변수로의 확장 용이성과 같은 장점으로 인해 학계나 산업계에서 많은 관심을 받아왔다[1-3]. 제약조건을 효과적으로 다루고 안정성을 보장하기 위해 최근에는 RHC를 볼록 최적화(convex optimization) 문제의 일종인 선형행렬부등식(Linear Matrix Inequality, LMI)문제로 변화하는 연구들이 많이 발표되었다[4-6]. RHC는 개념상 이산형 시스템에 대해 많이 연구되어져 왔으나, 실제의 시스템은 대부분 연속형 시스템이라는 사실에 입각해 연속형 시스템에 대한 RHC에 대한 연구도 최근 많이 이루어지고 있다[7-9]. 참고문헌 [7]과 [8]에서는 비선형 시스템에 대한 RHC를 제안하여 RHC가 비선형 최적화 문제로 설정되는데 비해 [9]에서는 선형 시스템에 대한 연속형 RHC를 제안하고 해를 구하기 위해 LMI를 이용한 최적화 문제를 제안하였다. 하지만 연속형 시스템에 대한 RHC는 최종 문제가 무한차원의 최적화 문제로 변환되기 때문에 해를 구할 수 없다는 단점을 가지고 있다. 따라서 [9]에서는 이러한 단점을 우회하기 위해 원래의 무한차원 최적화 문제를 계산이 가능한 유한차원 최적화 문제로 근사화시켜 LMI문제로 설정하는 방법을 제시하였다. [9]에서 제안된 구현 알고리즘은 무한차원 최적화 문제를 유한차원 최적화 문제로 만들기 위해 예측구간에서의 입력궤적을 유한개의 벡터에 의해 정의되는 불연속적인 형태로 가정한다. 이를 바탕으로 예측자를 구성하고 비용을 유도하는데 비용을 유도하는 과정에서 근사화 과정이 한 번 더 도입된다. 결과적으로 최종적으로 설정된 문제는 비록 유한차원 최적화문제이기는 하지만 본래의 연속형 RHC문제를 효과적으로 근사화시키지 못하는 단점을 가지고 있었다. 본 논문에서는 기존의 연속형 RHC구현 알고리즘이 갖는 단점을 개선하여 유한차원 최적화 문제로 변형하되 최대한 본래의 연속형 RHC문제를 근사화 시킬 수 있는 구현 알고리즘을 제안하는 것을 목표로 한다.

* 책임저자(Corresponding Author)
논문접수 : 2005. 3. 29., 채택확정 : 2005. 6. 7.
이영삼, 김태신, 김창유 : 인하대학교 전자전기공학부
(lys@inha.ac.kr/wtigerw@hanmail.net/changyu1@hanmail.net)

II. 개선된 연속형 RHC 구현 알고리즘

개선된 알고리즘을 제안하기에 앞서 기존의 연속형 RHC 문제를 간략히 살펴보자[9]. 연속형 시스템의 동특성이

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{1}$$

와 같다고 가정하자. 여기서 $x \in R^n$ 은 상태변수, $u \in R^m$ 은 제어입력을 나타낸다. 시스템 (1)은 다음과 같은 입력제약조건을 갖는다.

$$u_{j,\min} \le u_j(t) \le u_{j,\min},\ t \ge 0,\ \ j = 1, 2, \cdots, m$$

여기서 아래첨자 j 는 벡터의 j 번째 원소를 의미한다. 벡터에서의 원소별 대소 관계가 위와 같이 일률적으로 성립할 때 표기상의 편의를 위해 다음과 같이 나타내기로 한다.

$$u_{\min} \le u(t) \le u_{\max},\ t \ge 0 \tag{2}$$

연속형 RHC는 동특성 (1)과 입력 제약조건(2)를 만족하며 다음과 같이 표현되는 비용함수 (cost function)

$$V = \int_t^{t+T_p} \left[x^T(s)\,Q\,x(s) + u^T(s)\,R\,u(s) \right] ds + x^T(t+T_p)\,H\,x(t+T_p)$$

를 최소화하는 입력 궤적 $u(s),\ s \in \left[t, t+T_p\right]$을 구한 후에 $s = t$ 일 때의 입력, 즉 $u(t)$ 를 제어입력으로 사용하는 제어기법이다. 여기서 H 는 종단가중 행렬이고 T_p 는 예측구간이다. 연속형 RHC를 구하기 위해서는 최적화 문제를 풀어야 하는데, 이때의 최적화 문제를 설정하기 위하여 아래의 관계식을 만족하는 새로운 변수 γ_1과 γ_2를 도입한다.

$$J_1 = \int_t^{t+T_p} \left[x^T(s)\,Q\,x(s) + u^T(s)\,R\,u(s) \right] ds \le \gamma_1 \tag{3}$$

$$J_2 = x^T(t+T_p)\,H\,x(t+T_p) \le \gamma_2 \tag{4}$$

최적화 문제는 다음과 같이 표현된다.

$$\min\ \gamma_1+\gamma_2 \quad \text{subject to}$$

$$\dot{x}(s)=Ax(s)+Bu(s),\quad s\in\left[t,\ t+T_p\right] \tag{5}$$

$$S>0 \tag{6}$$

$$\begin{bmatrix} (AS+BY)+(AS+BY)^T & SQ^{\frac{1}{2}} & Y^TR^{\frac{1}{2}} \\ * & -\gamma_2 I & 0 \\ * & * & -\gamma_2 I \end{bmatrix}\le 0 \tag{7}$$

$$\int_t^{t+T_p}\left[x^T(s)Q\,x(s)+u^T(s)R\,u(s)\right]ds\le\gamma_1 \tag{8}$$

$$\begin{bmatrix} 1 & x^T(t+T_p) \\ * & S \end{bmatrix}\ge 0 \tag{9}$$

$$\begin{bmatrix} Z & Y \\ * & S \end{bmatrix}\ge 0,\quad Z_{jj}\le u_{j,\lim}^2 \tag{10}$$

$$u_{\min}\le u(s)\le u_{\max},\quad s\in\left[t,t+T_p\right] \tag{11}$$

여기서 S,Y,Z,γ_1,γ_2 는 모두 선형행렬부등식(Linear Matrix Inequality) 변수이다. Z_{jj} 는 Z 의 j 행 j 열 원소를 나타내며 $u_{j,\lim}$ 은 벡터 $u_{\lim}$ 의 j 번째 원소를 뜻한다. $u_{j,\lim}$ 은 다음과 같이 정의된다.

$$u_{j,\lim}=\min\left(\left|u_{j,\min}\right|,\left|u_{j,\max}\right|\right)$$

그리고 $H=\gamma_1 S^{-1}$ 의 관계가 성립한다. 연속형 RHC에 대한 좀더 자세한 내용은 참고문헌 [9]을 참조하라.

하지만 이 문제는 (11)에서 알 수 있듯이 무한개의 최적화 변수를 갖는 무한차원 최적화 문제가 되어 현실적으로 풀 수 없다. 이런 이유로 연속형 RHC를 구현하기 위해서는 본래의 무한차원 최적화 문제를 유한차원 최적화 문제로 근사화 시켜 풀이가 가능한 형태로 변형시키는데, (5),(8),(9),(11)이 변형된다.

그림 1은 기존의 알고리즘에 사용하고 있는 근사화 방식의 기본 개념을 나타내고 있다. 기존에 사용되는 구현 알고리즘에서는 $u(s),\ s\in\left[t,t+T_p\right]$ 를 그림 1과 같이 근사화 시킨다. 즉 다음과 같이 가정한다.

$$u(s)=u(t_i),\ t_i\le s<t_{i+1},\ i=0,1,\cdots,N-1. \tag{12}$$

여기서 $t_i=t+ih$ 이고 h 는 최적화 문제 계산주기를 나타낸다. 결과적으로 N 개의 예측 입력 시퀀스 $u(t_0),\cdots,u(t_{N-1})$ 를 이용하여 t 에서 $t+T_p$ 구간의 입력을 근사화시킨다. 비용함수는 다음과 같이 근사화시킨다.

$$\begin{aligned} J_1&=\int_t^{t+T_p}\left[x^T(s)Q\,x(s)+u^T(s)R\,u(s)\right]ds \\ &\approx h\left[\sum_{i=0}^{N-1}x^T(t_i)Q\,x(t_i)+u^T(t_i)R\,u(t_i)\right] \end{aligned} \tag{13}$$

따라서 최적화문제는 $u(s),\ s\in\left[t,t+T_p\right]$ 대신 $u(t_0),u(t_1),\cdots,u(t_{N-1})$ 를 구하면 되는 유한차원 최적화 문제로 변형되고, RHC는 이 중 첫번째인 $u(t_0)$ 를 t 에서 $t+h$ 동안 사용한다. $t+h$ 시점에서는 위의 과정이 반복된다.

하지만 (12)과 (13)을 기반으로 한 기존 알고리즘의 근사화 방법은 2가지 측면에서 본래의 RHC 문제를 충분히 반영하지 못하고 있다. 먼저 (12)과 같이 $u(s),\ s\in\left[t,t+T_p\right]$ 를 근사화시키는 것이 효과적이지 못하다. 둘째는 (12)로 가정하였을 경우 발생하는 상태변수에 대한 비용을 (13)이 정확히 나타내지 못한다는 점이다. $t_i<s<t_i+h$ 일때의 $x(s)$ 에 의해 발생하는 비용이 정확히 계산된 것이 아니고 $x(t_i)$ 의 값만을 이용해 상태변수에 대한 비용을 근사화시켰기 때문이다. 본 논문에서는 위에서 지적한 2가지 사항을 개선함으로써 제어성능이 향상된 구현 알고리즘을 제안하고자 한다.

III. 제안되는 구현 알고리즘

이 본 논문에서 제안하는 RHC 구현 알고리즘은 유한한 개수의 예측 입력 시퀀스를 구한다는 면에서는 기존의 알고리즘과 같지만 입력의 형태를 유한개의 입력 벡터를 선형 보간하여 그림 2와 같이 연속적인 형태로 가정한다. 그림 3은 선형보간 방식으로 입력형태를 가정하는 것을 설명한다. 즉 입력은 다음과 같이 가정한다.

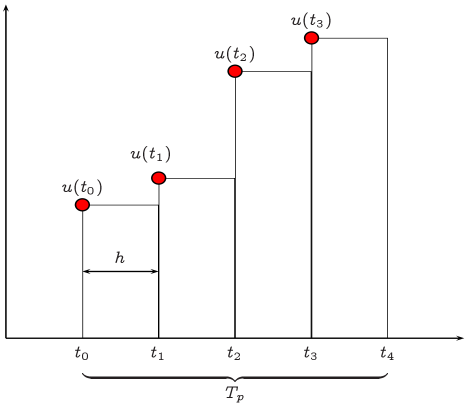

그림 1. 기존의 연속형 RHC 구현 기본 개념도.
Fig. 1. Concept of the conventional method.

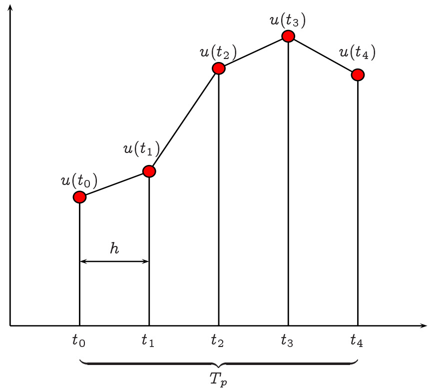

그림 2. 제안되는 연속형 RHC의 구현 기본 개념도.
Fig. 2. Concept of the proposed method.

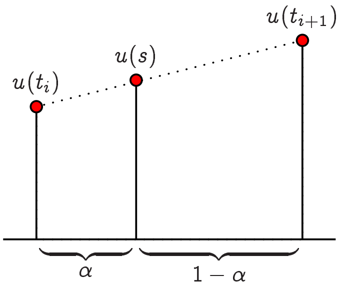

그림 3. 선형보간된 입력.
Fig. 3. Linearly interpolated input.

$$u(s)=(1-\alpha)u(t_i)+\alpha u(t_{i+1}),\ t_i\le s\le t_{i+1},\ i=0,1,\cdots,N-1. \tag{14}$$

여기서 α 는 다음과 같다.

$$\alpha=\frac{s-t_i}{h}$$

위와 같이 가정한 입력이 (12)에 비해 본래의 연속형 RHC 문제를 더 효과적으로 근사화하는 것을 4절에서 수치예제를 통해 뒷받침한다. 아래에서는 예측 입력을 (14)와 같이 가정하였을 때 비용함수를 예측자를 이용하여 정확히 나타내는 식을 유도하기로 한다.

1. 예측자(predictor) 유도

$x(t_0)$ 가 주어졌을 때, $x(t_1),x(t_2),\cdots,x(t_N)$ 에 대한 예측식을 나타내는 예측자를 구해보자. 먼저 $x(t_i)$ 가 주어졌다고 가정하면 $x(t_{i+1})$ 은 다음과 같이 나타낼 수 있다.

$$x(t_{i+1})=e^{Ah}x(t_i)+\int_{t_i}^{t_i+h}e^{A(t_i+h-s)}Bu(s)ds$$

$t_i\le s\le t_{i+1}$ 이므로 $0\le\alpha\le 1$ 이 성립한다. 예측 입력을 (14)와 같이 가정하면 다음과 같은 관계식을 얻는다.

$$\begin{aligned}x(t_{i+1})&=e^{Ah}x(t_i)\\&\quad+h\int_0^1 e^{(1-\alpha)hA}B\left[(1-\alpha)u(t_i)+\alpha u(t_{i+1})\right]d\alpha\\&=A_dx(t_i)+B_{1d}u(t_i)+B_{2d}u(t_{i+1}).\end{aligned} \tag{15}$$

여기서 A_d, B_{1d}, B_{2d} 는 다음과 같다.

$$\begin{aligned}A_d&=e^{Ah},\ B_{1d}=\left[h\int_0^1(1-\alpha)e^{(1-\alpha)hA}Bd\alpha\right]\\B_{2d}&=\left[h\int_0^1\alpha e^{(1-\alpha)hA}Bd\alpha\right]\end{aligned} \tag{16}$$

상태 예측 벡터와 입력 예측 벡터를 다음과 같의 정의하자.

$$\begin{aligned}X&=\left[x^T(t_0)\quad x^T(t_1)\quad\cdots\quad x^T(t_N)\right]^T\\U&=\left[u^T(t_0)\quad u^T(t_1)\quad\cdots\quad u^T(t_N)\right]^T\end{aligned} \tag{17}$$

(15)로부터 예측자를 다음과 같이 유도할 수 있다.

$$X=\bar{A}x(t_0)+\bar{B}U \tag{18}$$

$$x(t_N)=\bar{A}_Nx(t_0)+\bar{B}_NU \tag{19}$$

여기서,

$$\bar{A}=\left[I\quad A_d^T\quad\left(A_d^2\right)^T\quad\cdots\quad\left(A_d^N\right)^T\right]^T$$

$$\bar{B}=\begin{bmatrix}0_{n\times mN}&0_{n\times m}\\\Omega(A_d,B_{1d},N)&0_{nN\times m}\end{bmatrix}+\begin{bmatrix}0_{n\times m}&0_{n\times mN}\\0_{nN\times m}&\Omega(A_d,B_{2d},N)\end{bmatrix}$$

$$\bar{A}_N=A_d^N$$

$$\begin{aligned}\bar{B}_N&=\left[A_d^{N-1}B_{1d}\quad A_d^{N-2}B_{1d}\quad\cdots\quad B_{1d}\quad 0_{n\times m}\right]\\&\quad+\left[0_{n\times m}\quad A_d^{N-1}B_{2d}\quad A_d^{N-2}B_{2d}\quad\cdots\quad B_{2d}\right]\end{aligned}$$

이며, 이때 $\Omega(A_d,B_{id},N)$ 은 행렬 A_d,B_{id} 및 양의 정수 N 에 대해 다음과 같이 정의된다.

$$\Omega(A_d,B_{id},N)=\begin{bmatrix}B_{id}&0&\cdots&0\\A_dB_{id}&B_{id}&\cdots&0\\\vdots&\vdots&\ddots&\vdots\\A_d^{N-1}B_{id}&A_d^{N-2}B_{id}&\cdots&B_{id}\end{bmatrix}$$

참고로 $t_N=t+Nh=t+T_p$ 가 성립한다.

2. 비용(cost) 유도

앞에서 유도한 예측자식을 이용하여 비용(cost)을 유도할 수 있다. 비용은 (3), (4)와 같이 $\boldsymbol{J_1}$, $\boldsymbol{J_2}$ 로 나눌 수 있으며 다시 J_1 은 다음과 같이 2개의 부분으로 나눌 수 있다.

$$J_1=J_x+J_u$$

여기서 J_x 과 J_u 는 다음과 같다.

$$J_x=\int_t^{t+T_p}x^T(s)Rx(s)ds,\ J_u=\int_t^{t+T_p}x^T(s)Ru(s)ds$$

J_2 는 다음과 같이 나타낼 수 있다.

$$J_2=x^T(t_N)Hx(t_N) \tag{20}$$

(17)의 U 를 이용하면 J_u 는 다음과 같이 나타낼 수 있다.

$$\begin{aligned}J_u&=\int_t^{t+T_p}x^T(s)Ru(s)ds\\&=h\sum_{i=0}^{N-1}\int_0^1\left[(1-\alpha)u(t_i)+\alpha u(t_{i+1})\right]^TR\left[(1-\alpha)u(t_i)+\alpha u(t_{i+1})\right]d\alpha\\&=\frac{h}{3}\sum_{i=0}^{N-1}\left\{u^T(i)Ru(t_i)+u^T(i)Ru(t_{i+1})+u^T(i+1)Ru(t_{i+1})\right\}\\&=U^T\bar{R}U\end{aligned} \tag{21}$$

$$\bar{R}=\frac{h}{3}\underbrace{\begin{bmatrix}R&\frac{1}{2}R&0&0&\cdots&0\\\frac{1}{2}R&R&\frac{1}{2}R&0&\cdots&0\\0&\frac{1}{2}R&2R&\frac{1}{2}R&\cdots&0\\\vdots&\vdots&\vdots&\vdots&\ddots&\vdots\\0&\cdots&0&\frac{1}{2}R&2R&\frac{1}{2}R\\0&\cdots&0&0&\frac{1}{2}R&R\end{bmatrix}}_{N+1}$$

U 와 예측자 (18)과 (19)를 이용하여 J_x 를 유도해보자. s 와 r 을 다음과 같이 정의하자.

$$s = t_i + \theta h,\ 0 \le \theta \le 1,\ r = t_i + \alpha h,\ 0 \le \alpha \le \theta$$

(14)에 의해 $u(r) = (1-\alpha)u(t_i) + \alpha u(t_{i+1})$ 이 성립한다. 따라서 다음과 같이 정리할 수 있다.

$$\begin{aligned} x(s) &= e^{\theta hA}x(t_i) + \int_{t_i}^{t_i+\theta h} e^{A(t_i+\theta h-r)}Bu(r)dr \\ &= e^{\theta hA}x(t_i) \\ &\quad + h\int_0^\theta e^{(\theta-\alpha)hA}B\left[(1-\alpha)u(t_i) + \alpha u(t_{i+1})\right]d\alpha \end{aligned} \tag{22}$$

(22)를 이용하면

$$\begin{aligned} \int_{t_i}^{t_{i+1}} x^T(s)Qx(s)ds &= h\int_{t_i}^{t_{i+1}} x^T(t_i+\theta h)Qx(t_i+\theta h)d\theta \\ &= x^T(t_i)Q_1x(t_i) + u^T(t_i)Q_2u(t_i) + u^T(t_{i+1})Q_3u(t_{i+1}) \\ &\quad + 2u^T(t_i)Q_4u(t_{i+1}) + 2x^T(t_i)Q_5u(t_i) \end{aligned}$$

여기서 $Q_1, \cdots, Q_5$ 는 다음과 같다.

$$Q_1 = \left[h\int_0^1 (e^{\theta hA})^T Q(e^{ehA})d\theta\right] \tag{23}$$

$$Q_2 = h^3\int_0^1\left[\int_0^\theta (1-\alpha)e^{(\theta-\alpha)hA}Bd\alpha\right]^T Q\left[\int_0^\theta (1-\alpha)e^{(\theta-\alpha)hA}Bd\alpha\right]d\theta \tag{24}$$

$$Q_3 = h^3\int_0^1\left[\int_0^\theta \alpha e^{(\theta-\alpha)hA}Bd\alpha\right]^T Q\left[\int_0^\theta \alpha e^{(\theta-\alpha)hA}Bd\alpha\right]d\theta \tag{25}$$

$$Q_4 = h^3\int_0^1\left[\int_0^\theta (1-\alpha)e^{(\theta-\alpha)hA}Bd\alpha\right]^T Q\left[\int_0^\theta \alpha e^{(\theta-\alpha)hA}Bd\alpha\right]d\theta \tag{26}$$

$$Q_5 = h^2\int_0^1 (e^{\theta hA})^T Q\left[\int_0^\theta (1-\alpha)e^{(\theta-\alpha)hA}Bd\alpha\right]d\theta \tag{27}$$

결국 J_x 는 다음과 같이 나타낼 수 있게 된다.

$$\begin{aligned} J_x &= \sum_{i=0}^{N-1}\int_{t_i}^{t_{i+1}} x^T(s)Qx(s)ds \\ &= X^TQ_xX + 2X^TQ_{xu}U + U^TQ_uU \end{aligned} \tag{28}$$

여기서 Q_x, Q_{xu}, Q_u 는 다음과 같다.

$$Q_x = diag\{\underbrace{Q_1, \cdots, Q_1}_{N}, 0_{n\times n}\}$$

$$Q_{xu} = diag\{\underbrace{Q_5, \cdots, Q_5}_{N}, 0_{n\times m}\}$$

$$Q_u = \begin{bmatrix} Q_2 & Q_4 & 0 & 0 & \cdots & 0 \\ Q_4^T & Q_2+Q_3 & Q_4 & 0 & \cdots & 0 \\ 0 & Q_4^T & Q_2+Q_3 & Q_4 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & Q_4^T & Q_2+Q_3 & Q_4 \\ 0 & \cdots & 0 & 0 & Q_4^T & Q_3 \end{bmatrix}$$

(21), (28)로부터 J_1 을 다음과 같이 나타낼 수 있게 된다.

$$J_1 = X^TQ_xX + 2X^TQ_{xu}U + U^T[Q_u + \overline{R}]U$$

예측자 (18)과 (19)를 위 식과 (20)의 J_2 에 대입하면 다음과 같이 정리된다.

$$J_1 = U^T\Phi U + 2x^T(t_0)\Psi U + x^T(t_0)\Lambda x(t_0)$$

$$J_2 = [\overline{A}_N x(t_0) + \overline{B}_N U]^T H[\overline{A}_N x(t_0) + \overline{B}_N U]$$

여기서 Λ, Φ, Ψ 는 다음과 같다.

$$\begin{aligned} \Lambda &= \overline{A}^T Q_x \overline{A}, \\ \Phi &= \overline{B}^T Q_x \overline{B} + Q_u + \overline{R} + \overline{B}^T Q_{xu} + Q_{xu}^T \overline{B}, \\ \Psi &= \overline{A}^T[Q_x\overline{B} + Q_{xu}]. \end{aligned}$$

3. 최적화 문제

결국 본래의 최적화 문제중에서 동특성 (5)는 예측자 (18), (19)을 통해 만족되며 (8), (9), (11) 조건은 각각 다음과 같은 LMI조건으로 근사화된다.

$$\begin{bmatrix} \gamma_1 - x^T(t_0)\Lambda x(t_0) - 2x^T(t_0)\Psi U & U^T\Phi^{1/2} \\ * & -I \end{bmatrix} \ge 0 \tag{29}$$

$$\begin{bmatrix} 1 & [\overline{A}_N x(t_0) + \overline{B}_N U]^T \\ * & S \end{bmatrix} \ge 0 \tag{30}$$

$$U_{\min} \le U \le U_{\max} \tag{31}$$

여기서 $U_{\min}$ 과 $U_{\max}$ 는 다음과 같다.

$$U_{\min} = [\underbrace{u_{\min}^T \ \cdots \ u_{\min}^T}_{N+1}]^T,\ U_{\max} = [\underbrace{u_{\max}^T \ \cdots \ u_{\max}^T}_{N+1}]^T.$$

따라서 제안되는 연속형 RHC 구현 알고리즘은 h 의 시간 간격마다 다음과 같은 유한차원 최적화문제를 풀게 된다.

$$\boxed{\begin{gathered} \min\ \gamma_1 + \gamma_2 \\ \text{subject to (6), (7), (10), (29), (30), (31)} \end{gathered}} \tag{32}$$

위의 최적화 문제를 풀어 U 를 구하면 그 중에서 $u(t_0)$ 와 $u(t_1)$ 을 이용하여 t 에서 $t+h$ 의 시간 동안 RHC를 다음과 같이 적용한다.

$$u(s) = (1-\alpha)u(t_0) + \alpha u(t_1),\ \alpha = \frac{s-t_0}{h},\ s \in [t, t+h]$$

$t+h$ 시점에는 다시 최적화 문제를 풀어 위의 과정을 반복한다. 기존의 구현 알고리즘이 $u(t_0)$ 만을 이용하여 RHC를 구현하는 것과는 달리 제안되는 알고리즘에서는 $u(t_0)$ 와 $u(t_1)$ 을 이용하는 것이 특징이다.

Remark 1: (16)의 B_{1d}, B_{2d} 및 (23)~(27)의 $Q_1, \cdots Q_5$ 는 Simpson방법과 같은 수치해석적인 적분법을 이용하여 구할 수 있다 [10]. 이것은 h 간격 마다 풀어야 하는 최적화 문제와 달리 매번 구할 필요 없이 사전에 미리 계산할 수 있는 값으로 실시간 계산량에는 영향을 주지 않는다.

Remark 2: 동일한 예측구간 T_p 를 사용하였을 경우 제안되

는 알고리즘은 기존 알고리즘에 비해 최적화변수가 하나 더 많다. 이것은 기존 알고리즘은 최적화문제를 통해 $u(t_0), \cdots, u(t_{N-1})$ 을 구하지만, 제안되는 알고리즘은 $u(t_0), \cdots, u(t_N)$ 을 구하기 때문이다.

따라서 동일한 계산량을 갖도록 최적화문제를 설정하려면 제안되는 알고리즘의 예측구간을 기존 알고리즘의 예측구간 보다 h 만큼 짧게 선정하면 된다. RHC는 예측구간이 길면 길수록 비용이 줄어드는 비용단조감소(cost monotonicity) 성질이 있기 때문에 제안되는 알고리즘의 예측구간을 기존 알고리즘의 예측구간에 비해 짧게 선정하면 비록 계산량은 같다고 해도 제안되는 알고리즘의 제어성능이 나쁘리라고 생각할 수도 있다. 하지만 다음 절에서 다룰 수치 예제에서 볼 수 있듯이 동일한 계산량을 갖게 하기 위해 제안되는 알고리즘의 예측구간을 기존 알고리즘의 예측구간보다 짧게 선정해도 더 우수한 제어 성능을 나타낸다.

Remark 3: 본 논문에서는 연속형 RHC의 구현 알고리즘을 다루므로 안정성 부분을 심도있게 다루지는 않는다. 하지만 제안된 알고리즘을 적용하였을 경우 스템의 안정성 여부는 선형행렬부등식(LMI) 문제의 가해성 조건으로 판별 할 수 있다. 즉 (32) 의 LMI 문제가 해를 갖는 경우에는 제안되는 알고리즘을 통해 제어되는 되먹임 제어시스템은 안정하다고 말 할 수 있다.

IV. 수치 예제

제안되는 알고리즘의 개선된 성능을 알아보기 위해 다음과 같은 동특성을 갖는 도립진자 시스템에 적용해 보자.

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 29.43 & 0 & 0 & 0.3075 \\ 0 & 0 & 0 & 1 \\ -196.2 & 0 & 0 & -12.3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ -0.9547 \\ 0 \\ 38.1875 \end{bmatrix} u$$

$$x_1 = \theta,\ x_2 = \dot{\theta},\ x_3 = x,\ x_4 = \dot{x}$$

여기서 θ 는 도립진자가 지표면과 수직인 선과 이루는 각도, x 는 진자를 싣고 있는 수레가 초기의 중심점으로부터의 떨어진 변위를 나타낸다. $h = 0.2$ 일 때, 동일한 계산량을 갖는 최적화문제를 풀도록 하기 위해 기존의 알고리즘은 예측구간을 $T_p = 1.2$, 제안되는 알고리즘은 $T_p = 1$ 로 하여 각각을 도립진자의 제어에 적용하였다. 성능 비교를 위해 다음과 같은 기준을 사용하기로 한다.

$$J_{INT} = \int_0^5 \left[x^T(t) Q x(t) + u^T(t) R u(t) \right] dt$$

도립진자의 초기 상태는 $x(0) = \begin{bmatrix} 0.05 & 0 & 0 & 0 \end{bmatrix}^T$ 으로 주어지고 입력 제약조건은 다음과 같다.

$$u_{\min} = -10,\ \ u_{\max} = 10$$

그림 4는 기존의 알고리즘과 제안되는 알고리즘, 그리고 제약조건을 고려하지 않은 LQR (LQ Regulator)을 적용하여 얻은 도립진자의 각도 θ 를 각각의 알고리즘을 통해 얻은 J_{INT} 와 함께 비교하여 나타내고 있다. 제안되는 알고리즘이 기존의 알고리즘보다 20%이상 더 적은 J_{INT} 갖는 것을 볼 수 있다. 이는 더 좋은 제어 성능을 갖는 것을 의미한다. RHC는 예측구간 T_p 가 길면 길수록 제어 성능이 더 좋아지는 비용단조감소성을 가지고 있다. 따라서 기존 알고리즘의 예측구간을 $T_p = 1.2$ 로 정한것은 제안되는 알고리즘에 비해 더 유리한 조건이다. 즉 더 유리한 조건의 예측구간을 가진 기존 알고리즘보다 제안되는 알고리즘의 제어 성능이 더 우수하다는 것을 도립진자 예제를 통해서 살펴볼 수 있다. 또한 제안되는 알고리즘은 LQR 보다도 더 적은 J_{INT} 를 갖는 것을 볼 수 있다. LQR의 경우 제약조건을 고려할 수 없는 제어기법이기 때문에 설계된 제어기를 그대로 제약조건이 있는 시스템에 적용하면 제약조건을 고려하여 설계한 제어기보다 제어성능이 떨어질 수 있기 때문에 이런 현상이 발생하는 것으로 볼 수 있다. 그림 5는 각각의 경우 계산된 제어입력을 비교하여 나타내고 있다. 기존의 알고리즘과 제안된 알고리즘의 경우 둘 다 제약조건을 고려하였기 때문에 입력 제

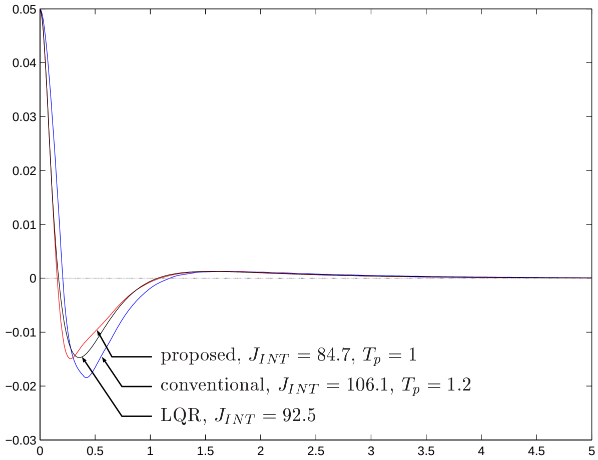

그림 4. 도립진자의 각도 (θ).

Fig. 4. Angle of a pendulum.

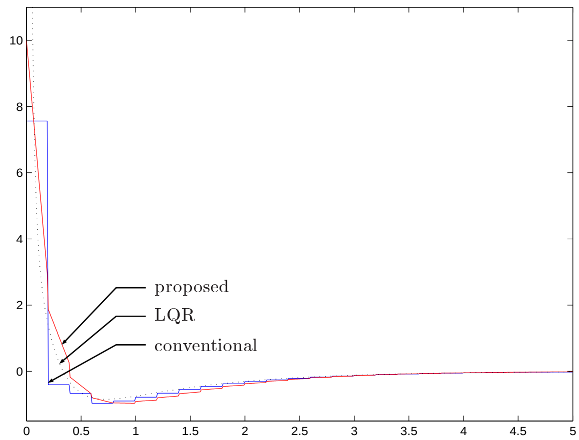

그림 5. 계산된 제어입력 (u).

Fig. 5. Control input.

약조건 ± 10 을 만족하고 있는 것을 볼 수 있다. 하지만 LQR의 경우 입력제약조건을 만족하지 못함을 알 수 있다. 또한 제안된 알고리즘을 통해 계산된 제어입력은 기존의 알고리즘에 의해 계산된 제어입력에 비해 좀 더 연속적인 형태를 나타냄을 볼 수 있는데, 결국 이러한 차이가 성능에서의 차이를 유발하였음을 짐작할 수 있다.

Remark 4: 위의 예제에서는 $h = 0.2$ 로 비교적 큰 값을 사용하였다. 하지만 h 의 값을 점점 감소시키면 기존의 알고리즘과 제안되는 알고리즘을 적용하였을 때 발생하는 제어성능의 차이는 차츰 감소하게 된다. 즉 제안되는 알고리즘은 h 가 비교적 큰 경우 유용하게 사용할 수 있다.

V. 결론

본 논문에서는 연속형 시스템의 RHC에 대해 개선된 구현 알고리즘을 제안하였다. 기존의 구현 알고리즘은 예측구간에서 불연속적인 형태의 제어입력 궤적을 가정하여 예측자를 유도하고 비용함수를 근사화시켜 구현 알고리즘을 구하였다. 이에 비해 제안되는 알고리즘은 예측구간 동안의 입력궤적을 유한개의 벡터를 선형보간시켜 연속적인 값을 갖도록 가정하여 예측자를 유도하고 이 예측자를 이용하여 비용을 계산하였다. 이 결과 제안되는 알고리즘은 기존의 알고리즘에 비해 본래의 연속형 RHC 문제를 보다 효과적으로 근사화시킬 수 있고 더 향상된 제어 성능을 나타낼 수 있다. 제안되는 구현 알고리즘을 도립진자 시스템에 모의실험을 통하여 적용함으로써 제안되는 구현 알고리즘의 성능 개선을 예시하였다.

참고문헌

[1] J. Richalet, A. Rault, J. L. Testud, and J. Papon, "Model predictive heuristic contro: application to industrial process," *Automatica*, vol. 14, pp. 413-420, 1978.

[2] D. W. Clarke and C. Mohtadi, "Properties of generalized predictive control," *Automatica*, vol. 25, pp. 859-875, 1989.

[3] R. Soeterboek, *Predictive control: A unified approach*, Prentice Hall, New York, 2002.

[4] S. Boyd, L. E. Chaoui, E. Feron, and V. Balakrishnan, *Linear matrix inequalities in system and control theory*, SIAM, Philadelphia, 1994.

[5] M. V. Kothare, V. Balakrishnan, and M. Marari, "Robust constrained model predictive control using linear matrix inequalities," *Automatica*, vol. 32, pp. 1361-1379, 1996.

[6] J. W. Lee, W. H. Kwon, and J. H. Choi, "On the stability of constrained receding horizon control with finite terminal weighting matrix," *Automatica*, vol. 34, no. 12, pp. 1607-1612, 1998.

[7] H. Michalska and D. Q. Mayne, "Robust receding horizon control of constrained nonlinear systems," *IEEE Trans. on Automatica Control*, vol. 38, no. 11, pp. 1623-1633, 1993.

[8] H. Chen and F. Allgöwer, "A quasiinfinite horizon nonlinear model predictive control scheme with guaranteed stability," *Automatica*, vol. 14, pp. 1205-1217, 1998.

[9] K. B. Ki, T. W. Yoon, and W. H. Kwon, "Receding horizon guidance laws for constrained missiles with autopilot lags," *Control Engineering Practice*, vol. 9, no. 10, pp. 1107-1115, 2001.

[10] S. C. Chapra and R. P. Canale, *Numerical methods for engineers*, McGraw Hill, 2002.

이 영 삼

1997년 인하대학교 전기공학과 졸업. 1999년 동 대학원 석사. 2003년 서울대 전기 컴퓨터 공학부 박사. 2004년~현재 인하대학교 전자전기공학부 전임강사. 관심분야는 임베디드 시스템, 로봇공학.

김 창 유

2005년 인하대학교 전자공학과 졸업. 2005년~현재 인하대 전기공학과 석사과정 재학중. 관심분야는 임베디드 시스템, 시스템 식별.

김 태 신

2002년 인하대학교 전자공학과 졸업. 2004년 동 대학원 석사. 2004년~현재 인하대 전기공학과 박사과정 재학중. 관심분야는 시스템식별, 다변수 시스템제어, 필터설계.