

실수코딩 유전알고리즘을 이용한 모델 예측 제어 시스템 설계

Model Predictive Control System Design with Real Number Coding Genetic Algorithm

방현진*, 박종천*, 홍진만*, 이흥기*

* 서울시 동작구 중앙대학교 전자전기공학부
E-mail : hglee@cau.ac.kr

요 약

모델 예측 제어 시스템은 이동 제어 구간에서 원하는 출력과 예측된 출력의 차이를 최소화하는 현재의 제어 입력을 적용하는 방식을 사용한다. 제약조건이 있는 경우이거나 비선형 시스템 문제의 경우는 주어진 함수를 최소화하는 최적화 문제를 풀기가 힘들다. 본 논문에서는 모델 예측 제어 시스템의 최적화 문제를 실수 코딩 유전 알고리즘을 이용하여 효율적으로 구할 수 있음을 보인다. 또한 실수코딩 유전알고리즘이 여러 가지 면에서 디지털코딩 유전알고리즘보다 더 자연스럽고 유리함을 모의실험을 통해 보인다.

Key Words : genetic algorithm, real number coding, predictive control, receding horizon control

1. 서 론

모델 예측 제어(Model Predictive Control)는 전 구간을 한꺼번에 고려하는 기존의 제어 방식과 달리 정해진 가까운 미래 구간의 입출력만 고려하면서 제어하는 것이다. 예를 들어, 서울에서 부산까지 가는데 100미터 앞까지만 보고 운전하는 것과 아주 비슷하다. 지금 순간 앞으로 10초 후까지의 운전자의 행동을 결정하였다더라도 1초 후에 다시 전방 100미터 앞을 보고 지금 취하여야 할 행동을 결정하는 방식이다. 이를 이동 구간 제어(receding horizon control)라고 하고 다음 절에서 간단히 설명한다. MPC에 관한 자세한 내용은 참고문헌 [3,4]에서 발견할 수 있다.

제약조건이 없는 선형시스템의 경우는 전 구간을 고려하던 일부 구간만을 고려하던 최적화하는 제어 규칙을 발견하는 데 어려움이 없다. 또한 이동 구간 제어를 사용하면 전 구간을 고려한 결과보다는 못하지만 그에 버금가는 결과를 얻을 수 있다는 것도 잘 알려져 있다. 비선형 시스템이거나 제약조건이 있는 선형시스템의 경우는 전 구간을 고려하여 최적화 하는 제어 규칙을 발견하기는 힘들다. 그렇지만 이동

구간 제어 방식을 사용하면 최적화 문제의 독립변수가 획기적으로 적어지기 때문에 훨씬 더 유리하다. 물론 독립 변수가 적어지더라도 여전히 최적화 문제를 풀기가 쉽지는 않고 이에 관한 내부 점 방법(interior point method) 등의 연구들은 [4] 등에서 발견할 수 있다.

본 논문에서는 이 최적화 문제를 해결하기 위해 실수코딩 유전알고리즘을 적용하는 것을 제안하고 이의 효용성에 대해 논한다. 유전알고리즘(genetic algorithm)은 유전적 계승과 적자생존이라는 자연현상을 알고리즘형태로 모델링한 확률적 탐색 방법으로 함수의 최적화 등에서 계산 시간을 획기적으로 줄일 수 있다. 유전알고리즘에 관한 자세한 내용은 참고문헌 [1,2]에서 발견할 수 있다. 참고문헌 [5,6]에서 디지털 코딩을 사용하는 보통의 유전알고리즘을 적용하여 모델 예측 제어를 구하였다. 본 논문의 경우처럼 실수 함수의 최적화 문제는 실수코딩이 코딩과 디코딩 과정이 필요치 않는 등 자연스러우면서 여러 가지 면에서 유리하다.

본 논문은 다음과 같이 구성되어 있다. 2절에서는 모델 예측 제어에 대해 간단히 소개하

고 3절에서 실수코딩 유전알고리즘을 이용한 모델 예측 제어를 구한다. 4절에서는 우리의 제어 시스템과 기존의 방법들을 모의실험을 통해서 비교하고 마지막으로 5절에서 이 논문의 결론을 내린다.

2. 모델 예측 제어기

MPC 시스템의 핵심은 이동 구간 제어 방식을 적용한다. 즉, 다음의 시스템을 고려하자.

$$y_{k+1} = f(y_k, y_{k-1}, \dots, y_{k-n}, u_k, u_{k-1}, \dots, u_{k-n}) \quad (1)$$

여기서 y_k 와 u_k 는 각각 시스템의 출력과 입력이다. 우리의 제어 목적은 출력 $\{y_k\}$ 가 주어진 원하는 출력 함수 $\{r_k\}$ 를 잘 추종하게 하는 것이다. 이를 위하여 평가함수(cost function)

$$J = \sum_{i=1}^{\infty} \|r_i - y_i\|^2 + \sum_{i=0}^{\infty} \|\Delta u_i\|^2 \quad (2)$$

를 최소화하는 $\{u_k\}_{k=0}^{\infty}$ 구하여 적용하면 된다. 여기서, $\Delta u_i = u_i - u_{i-1}$ ($u_{-1} = 0$)이다. 제약조건이 없는 선형 시스템 문제의 경우는 쉽게 이 문제를 풀 수 있지만 제약조건이 있거나 식 (1)이 비선형인 경우에는 최적화 문제를 쉽게 풀 수가 없다. 따라서 평가함수 J 를 다음과 같이 바꾼다.

$$J'_k = \sum_{i=1}^{n_y} \|r_{k+i} - \widehat{y}_{k+i}\|^2 + \sum_{i=0}^{n_u-1} \|\Delta u_{k+i}\|^2 \quad (3)$$

여기서 n_y 는 출력 예측 구간(prediction horizon)이라고 하고 n_u 는 제어 입력 구간(control horizon)이라고 한다. ($n_u \leq n_y$) 독립변수는 u_k, \dots, u_{k+n_u-1} 의 n_u 개다. $n_u < n_y$ 일 때 출력 $y_{k+n_u+1}, \dots, y_{k+n_y}$ 를 구할 수 없으므로

$$u_{k+i} = u_{k+n_u-1}, \quad n_u \leq i \leq n_y - 1 \quad (4)$$

을 가정하여 예측된 출력 \widehat{y}_{k+i} 를 실제 출력 대신에 적용한다. (이것이 예측제어라는 용어가 사용되는 이유이다). J 를 최적화하는 문제보다는 J'_k 를 최적화하는 문제가 독립 변수의 수가 적으므로 훨씬 간단하다. 이동 구간 제어 방식은 시간 k 인 순간에 J'_k 를 최소화하는 입력 u_k, \dots, u_{k+n_u-1} 을 구하여 u_k 만 쓰고 $u_{k+1}, \dots, u_{k+n_u-1}$ 은 무시하고 시간 $k+1$ 인 순간에는 J'_{k+1} 을 최소화하는 입력을 구하여 같은 방법으로 계속해 나간다. 이런 이동 구간 제어 방식을 사용했을 때 J 를 고려했을 때보다 성능은 다소 떨어지지만 제어가 훨씬 쉬워지는 장점이 있다. n_y, n_u 값은 모델 예측 제어기에서 매우 중요한 변수이다. n_u 가 작으면 식 (3)을 최소화하는 입력은 쉽게 구할 수 있으나 시스템 성능이 나빠질 수 있다. 이에 관해서는 참고문헌[4] 등에서 발견할 수 있다.

마지막으로 제약조건에 대해 설명한다. 가장 간

단하면서 흔히 접하는 제약조건은 다음과 같다.

$$\underline{u}_i \leq u_i \leq \overline{u}_i \quad (5)$$

3. 실수코딩 유전알고리즘을 이용한 MPC

전 절에서 설명한 MPC 알고리즘은 다음과 같이 요약된다. 우선, 적절한 n_y, n_u 값을 선택한 후 다음의 과정을 반복한다.

Step1. 제약조건 (5)와 식 (4)를 만족하면서 식 (3)의 평가함수 J'_k 을 최소화하는 입력 u_k, \dots, u_{k+n_u-1} 를 구한다.

Step2. 입력 $u_{k+1}, \dots, u_{k+n_u-1}$ 는 무시하고 입력 u_k 를 적용한다.

위 과정을 매 순간 반복한다.

본 논문에서는 위 Step 1의 최적화 문제를 실수코딩 유전알고리즘을 이용하여 해결하는 방법을 제시한다.

염색체 $s_i (\in R^{n_u})$ 는 다음과 같이 정의한다.

$$s_i = \begin{bmatrix} s_{i1} \\ \vdots \\ s_{in_u} \end{bmatrix} = \begin{bmatrix} u_k \\ \vdots \\ u_{k+n_u-1} \end{bmatrix} \quad (6)$$

위의 염색체에 대해 교배는 볼록교배(convex crossover)를 사용하고 돌연변이는 실수크리프돌연변이(real number creep mutation) 연산자를 적용한다. 두 부모 염색체 s_i 와 s_j 에 대해 볼록교배에 의해 하나의 자손 s_c 가 다음의 규칙으로 생성된다.

$$s_c = \lambda s_i + (1 - \lambda) s_j \quad (7)$$

여기서 λ 는 $[0,1]$ 에서 발생하는 난수이다. 부모 염색체들이 식 (5)와 같은 볼록한(convex) 제약조건을 만족하면 식 (7)에 의한 자손 염색체도 식 (5)의 제약조건을 만족한다. 만일 제약조건이 복잡해서 볼록하지 않으면 s_c 가 제약조건을 만족할 때까지 식 (7)의 작업을 계속한다.

또한, 돌연변이는 하나의 염색체 s_i 에 대해 하나의 자손 s_c 가 다음의 규칙으로 생성된다.

$$s_{cj} = s_{ij} + 2(1 - \lambda_j)\epsilon(\overline{u}_j - \underline{u}_j), \quad 1 \leq j \leq n_u \quad (8)$$

여기서 λ_j 는 $[0,1]$ 에서 발생하는 난수이고 $\epsilon = 0.01$ 로 한다. 즉 s_c 는 s_i 근처에 있는 점이고 정확한 해를 구하는데 유리하다. 만일 s_c 가 식 (5)의 제약조건을 만족하지 않으면 만족할 때까지 식 (8)의 작업을 계속한다.

유전알고리즘의 초기 집단은 $1 \leq j \leq n_u$ 인 임

의의 j 에 대해

$$s_{ij} = \underline{u}_j \text{ or } \overline{u}_j \quad (9)$$

$$s_{il} = \underline{u}_l + \lambda_l (\overline{u}_l - \underline{u}_l), \quad l \neq j$$

을 만족하는 임의의 염색체 N 개로 구성한다. 여기서 λ_l 은 0과 1사이의 임의의 실수다. MPC 문제의 답이 종종 제약조건의 경계점에 있으므로 초기 집단을 위와 같이 정하는 게 좋을 뿐 아니라 다음 세대들에서도 경계에 있는 염색체들을 일정 수만큼 계속 공급해 주는 것이 해를 구하는데 유리할 것이다.

다음 세대를 정하는 재생산은 어떤 방식을 적용하여도 되지만 N 개 중 예를 들어 75퍼센트만 선택하고 나머지 25퍼센트는 전역탐색과 위에 설명하였듯이 경계부분 해를 고려하여 초기 집단을 선택하는 방식으로 공급한다. 이로 인하여 지역 해(local solution)에 빠지지 않고 정확한 해를 효율적으로 구할 수 있는 알고리즘이 된다.

이상의 규칙들을 이용하여 여러 가지 유전알고리즘을 고려할 수 있다. 다음 절에서 우리의 유전알고리즘의 유용성을 모의실험을 통하여 보인다. 다음 절에서 사용한 알고리즘은 구체적으로 다음의 3단계로 구분된다.

1. **블록교배:** N 개의 염색체들을 임의로 일렬로 나열하여 차례로 인접한 두 염색체들을 부모로 하고 식 (7)의 블록교배에 의해서 $N/2$ 개의 자손 염색체들을 얻는다. $3N/2$ 개의 부모와 자손 염색체들 중 적합도 순위(rank)에 의해 $N/2$ 개의 염색체들을 선택하고 나머지는 버린다.

2. **국소돌연변이:** 1에서 얻은 $N/2$ 개의 염색체들 중 임의로 $N/4$ 개의 염색체들을 선택하여 식 (8)의 돌연변이에 의해 $N/4$ 개의 자손을 얻는다.

3. **전역탐색:** 1과 2에서 $3N/4$ 개의 다음 세대의 염색체들을 얻었는데, 나머지 $N/4$ 개의 염색체들은 식 (9)에 의해 초기 집단과 같은 방법으로 구성한다.

이상의 세 과정들을 정해진 세대만큼 반복하여 최종적으로 얻은 집단에서 적합도가 가장 높은 염색체가 Step1에 정의된 최적화 문제의 답 $[u_k, \dots, u_{k+n_u-1}]^T$ 이다. 과정 3과 과정 1이 결합하여 계속 전역탐색이 가능하게 되고, 과정 2와 과정 1이 결합하여 보다 더 정확한 해를 구할 수 있다.

4. 예 제

이 절에서는 비선형 시스템

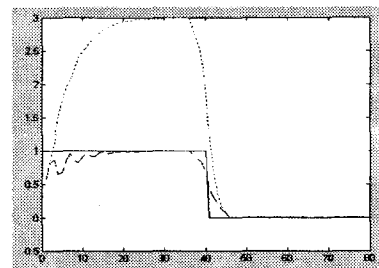
$$y_{k+1} = \frac{y_k y_{k-1} y_{k-2} u_{k-1} (y_{k-2} - 1) + u_k}{1 + y_{k-1}^2 + y_{k-2}^2} \quad (10)$$

에 대하여 앞에서 제시된 유전알고리즘을 이용한 모델 예측 제어기를 설계하고 컴퓨터 시뮬레이션을 통해 기존의 연구 결과인 이진코딩 유전알고리즘을 이용한 방법[5,6]과 비교한다. 출력 예측 구간과 제어 입력 구간을 $n_y = 5$ 와 $n_u = 3$ 로 하고 집단크기 $N = 80$ 이고 100세대까지 진화한다고 가정한다. 간편한 시뮬레이션을 위하여 MATLAB을 사용한다.

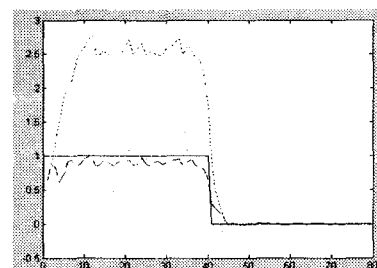
Table 1 그림에서 사용된 표기

입력	추종 출력	출력
Dot	Solid	Dash

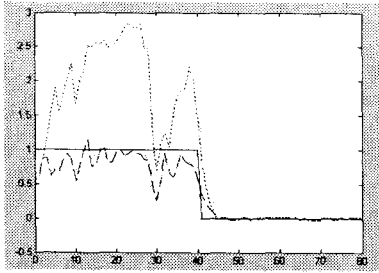
돌연변이 확률 p_m 을 0.2로 하고 이진 코딩의 길이는 각 입력마다 10으로 하여 ($n_u = 3$ 이므로 유전자의 전체 길이는 30으로 하여) 여러 입력 범위에 대한 이진코딩 유전알고리즘 MPC 실험결과를 그림 1에 보였다. 그림 2는 같은 조건하의 실수코딩 유전알고리즘 MPC 실험결과이다. 그림 1에서는 입력 범위에 따라 성능이 많이 변하지만 그림 2에서는 성능이 비슷함을 알 수 있다. 또한 유전알고리즘에서는 실험 시간도 중요한데 그림 1에서 약 240분이 걸리고 그림 2에서 약 75분이 걸렸으므로 3분의 1 수준으로 줄어들었다. 이는 코딩과 디코딩 과정에 드는 시간과 돌연 변이를 각각의 디지털에 적용하는데 걸리는 시간에 기인한다.



(a) $|u| \leq 3.0$

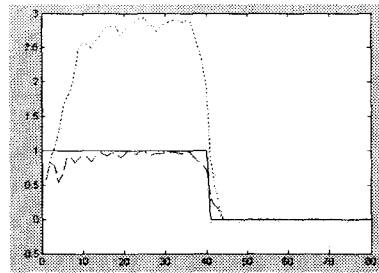


(b) $|u| \leq 10.0$



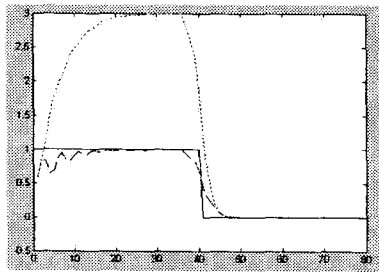
(c) $|u| \leq 20.0$

그림 1. 이진코딩 유전알고리즘 MPC
Fig 1. GA based MPC with binary coding

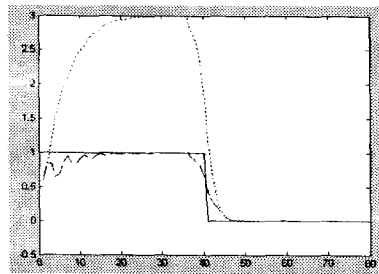


$|u| \leq 10.0$, 코드길이=12

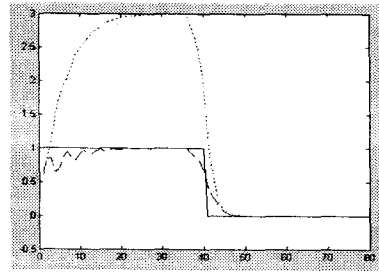
그림 3. 이진코딩 유전알고리즘 MPC
Fig 3. GA based MPC with binary coding



(a) $|u| \leq 3.0$



(b) $|u| \leq 10.0$



(c) $|u| \leq 20.0$

그림 2. 실수코딩 유전알고리즘 MPC
Fig 2. GA based MPC with real coding

그림 3에는 $|u| \leq 10.0$ 로 하고 코드 길이를 달리 한 결과인데 그림 1(b)의 결과와 큰 차이가 없다. 코드 길이를 길게 하면 계산 시간은 그 만큼 더 걸린다. 물론 코드 길이를 길게 하고 더 많은 세대를 진화시키면 성능이 개선될 것이다.

5. 결론

본 논문은 실수코딩 유전알고리즘을 이용하여 모델 예측 제어를 설계하고 시스템의 성능을 개선하는 방법에 대한 연구이다. 예제에서 보듯이 MPC와 같은 실수 값을 다루는 문제에서는 이진 코딩보다 실수코딩을 사용한 유전알고리즘이 여러 가지 면에서 더 유리하다. 그 중 가장 중요한 것은 시간문제이다. 코딩과 디코딩 과정과 돌연 변이 과정에 드는 시간을 절약하면 더 많은 집단 또는 세대를 진화할 수 있다. 또한 실수코딩을 이용하면 코드의 길이에 대해 신경 쓸 필요가 없고 예제에서 보듯이 해의 범위(이 경우, 입력의 범위)에도 신경을 덜 써도 된다.

유전알고리즘이 시간이 많이 걸리므로 비교적 여유가 있는 화학공정 등을 제외한 로봇 같은 빠른 제어를 요구하는 경우 더 많은 연구가 필요하다.

참 고 문 헌

- [1] 강훈, 심귀보, "지능정보시스템", 브레인코리아, 2003.
- [2] 진강규, "유전알고리즘과 그 응용", 교우사, 2000.
- [3] E.F. Camacho and C. Bordons, "Model Predictive Control", Springer, 2004
- [4] J.A. Rossiter, "Model-Based Predictive Control", CRC Press, 2003
- [5] S.C. Shin and Z. Bien, "Constrained GA-based Predictive Control," 대한전자공학회, 추계종합학술대회논문집, 22권 2호, pp.732-735, 1999.
- [6] S.C. Shin and S.B. Park, "GA-based predictive control for nonlinear processes," Electronics Letters, Vol.34. pp.1980-1981, 1998.