(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2016년02월12일
(11) 등록번호 10-1593038
(24) 등록일자 2016년02월02일

(51) 국제특허분류(Int. Cl.)
G05B 13/00 (2006.01)
(21) 출원번호 10-2014-0159847
(22) 출원일자 2014년11월17일
심사청구일자 2014년11월17일
(56) 선행기술조사문헌
WO2007139088 A1
JP2006525447 A
KR100221231 B1

(73) 특허권자
인하대학교 산학협력단
인천광역시 남구 인하로 100, 인하대학교 (용현동)
(72) 발명자
이영삼
인천광역시 연수구 용담로 14 (청학동, 하나아파트) 107동 1503호
경기영
경기 이천시 경충대로2586번길 31, 4층 (중리동, 남천빌딩)
(뒷면에 계속)
(74) 대리인
이원희

전체 청구항 수 : 총 9 항 심사관 : 강석제

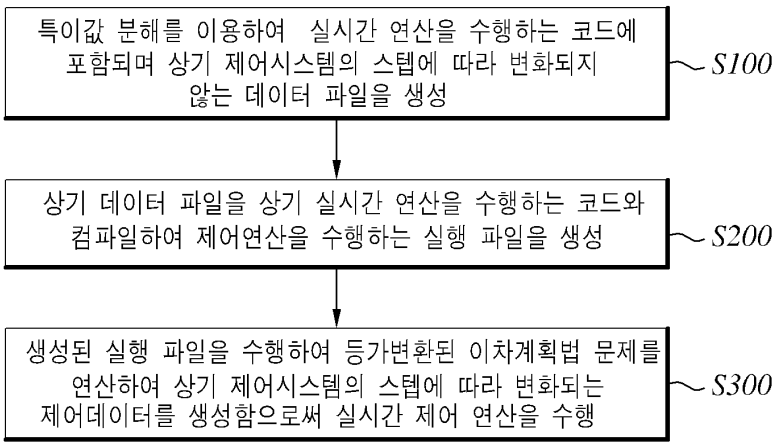
(54) 발명의 명칭 **이차계획법을 이용한 예측제어 방법**

(57) 요 약

본 발명은 이차계획법을 이용한 예측제어 방법에 관한 것으로,

제어시스템의 제약조건을 만족하면서 주어진 비용함수를 최소화하기 위하여 특이값 분해를 이용하여 실시간 연산을 수행하는 코드에 포함되며 상기 제어시스템의 스텝에 따라 변화되지 않는 데이터 파일을 생성하는 단계; 상기 데이터 파일을 상기 실시간 연산을 수행하는 코드와 컴파일하여 제어연산을 수행하는 실행 파일을 생성하는 단계; 상기 생성된 실행 파일을 수행하여 등가변환된 이차계획법 문제를 연산하며, 상기 제어시스템의 스텝에 따라 변화되는 제어 데이터를 생성함으로써 실시간 제어 연산을 수행하는 단계를 포함하는 것을 특징으로 한다.

대 표 도 - 도1



(72) 발명자

박재헌

인천 남구 인하로 292, 1동 1201호 (주안동, 쌍용아파트)

김진걸

경기도 과천시 별양로 12 래미안슈르 아파트 335-1704

이 발명을 지원한 국가연구개발사업

과제고유번호 NIPA-2014-H0401-14-1003

부처명 미래창조과학부

연구관리전문기관 정보통신산업진흥원

연구사업명 IT 융합 고급인력과정 지원사업

연구과제명 IT 융합 재활의료기기 센터(생체신호를 이용한 IT기반 재활의료기기 개발)

기 여 율 1/1

주관기관 한국산업기술대학교

연구기간 2011.06.01 ~ 2014.12.31

특허청구의 범위

청구항 1

제1 컴퓨팅 수단이 제어시스템의 제약조건을 만족하면서 주어진 비용함수를 최소화하기 위하여 특이값 분해를 이용하여 실시간 연산을 수행하는 코드에 포함되며 상기 제어시스템의 스텝에 따라 변화되지 않는 데이터 파일을 생성하는 단계;

상기 제1 컴퓨팅 수단이 상기 데이터 파일을 상기 실시간 연산을 수행하는 코드와 컴파일하여 제어연산을 수행하는 실행 파일을 생성하는 단계; 및,

제2 컴퓨팅 수단이 상기 생성된 실행 파일을 수행하여 등가변환된 이차계획법 문제를 연산하여 상기 제어시스템의 스텝에 따라 변화되는 제어데이터를 생성함으로써 실시간 제어 연산을 수행하는 단계

를 포함하는 이차 계획법을 이용한 예측제어 방법.

청구항 2

청구항 1에 있어서,

상기 비용함수(J)는 하기 수학식 (1)로부터 연산되고, 상기 제약조건은 하기 수학식 (2)로부터 연산되는 것을 특징으로 하는 이차 계획법을 이용한 예측제어 방법:

수학식(1); $$Minimize\ J=\sum_{j=1}^{N_p}\| w_{k+j}-\widehat{y_{k+j}}\|^2+\sum_{j=0}^{N_c-1}\| \Delta u_{k+j}\|_{\Lambda}^2$$,

여기서, w는 기준입력, $\hat{y}$는 예측 출력, Δu는 제어증분(control increment)를 나타내며, N_p는 예측구간(prediction horizon), N_c는 제어구간(control horizon), Λ는 제어증분에 대한 가중치를 나타내는 대각행렬이고,

수학식(2); $$u_{\min}\le u_{k+j}\le u_{\max},\ \Delta u_{\min}\le \Delta u_{k+j}\le \Delta u_{\max}$$,

여기서, 아래첨자는 시점을 나타내는 것으로, k는 현재 시점, u는 입력, u_{k+j}는 현재 시점으로부터 j스텝 이후의 입력을 의미함.

청구항 3

청구항 1에 있어서,

상기 이차계획법 문제는 하기 수학식 (4)의 조건에 따른 하기 수학식 (3)으로 표현되는 것을 특징으로 하는 이차 계획법을 이용한 예측제어 방법:

수학식(3); $$Minimize\, J(x)=\frac{1}{2}x^TPx+q^Tx$$,

수학식(4); $$Ax\le b$$

여기서, J는 비용함수, x는 최적화 변수, P 및 A는 행렬, q 및 b는 소정의 벡터임.

청구항 4

청구항 3에 있어서,

상기 데이터 파일은 상기 P 및 상기 A와 관련된 데이터를 포함하고,

상기 제어 데이터는 상기 q 및 상기 b와 관련된 데이터를 포함하는 것을 특징으로 하는 이차 계획법을 이용한 예측제어 방법.

청구항 5

청구항 3에 있어서,

상기 P는 하기 수학식 (5)로부터 산출되고, 상기 A는 하기 수학식 (6)으로 표현되는 것을 특징으로 하는 이차 계획법을 이용한 예측제어 방법:

수학식 (5): $$P = G^T G + \Xi$$,

여기서, $\Xi = diag[\Lambda, \Lambda, \cdots, \Lambda]$ 이고, Λ는 제어증분에 대한 가중치를 나타내는 대각행렬이고, G는 시스템의 모델정보를 이용하여 계산한 예측행렬이고,

수학식 (6): $$A = \begin{bmatrix} I_D \\ -I_D \\ I_L \\ -I_L \end{bmatrix}$$,

여기서, $$I_D = \begin{bmatrix} I_m & 0 & \cdots & 0 \\ 0 & I_m & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & I_m \end{bmatrix}, \; I_L = \begin{bmatrix} I_m & 0 & \cdots & 0 \\ I_m & I_m & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ I_m & I_m & \cdots & I_m \end{bmatrix}$$ 임.

청구항 6

청구항 3에 있어서,

상기 q는 하기 수학식 (7)로부터 산출되고, 상기 b는 하기 수학식 (8)로부터 산출되는 것을 특징으로 하는 이차 계획법을 이용한 예측제어 방법:

수학식 (7): $$q = -2(W_k - f_k)^T G$$

여기서, W는 기준입력, f_k는 자유응답(free response)으로써 과거의 데이터가 미래의 출력에 기여하는 양, G는 시스템의 모델정보를 이용하여 계산한 예측행렬이고,

수학식 (8): $$b = \begin{bmatrix} I_V u_{\max} \\ -I_V u_{\min} \\ I_V (u_{\max} - u_{k-1}) \\ -I_V (u_{\min} - u_{k-1}) \end{bmatrix}$$

$I_V = \begin{bmatrix} I_m \\ I_m \\ \vdots \\ I_m \end{bmatrix}$, k는 현재 시점, u는 입력임.

여기서,

청구항 7

청구항 1에 있어서,

상기 이차계획법 문제의 연산은,

상기 이차계획법 문제를 부등식 조건을 갖는 최소제곱 문제(LSI, Least squares with inequality)로 등가변형하는 단계;

상기 부등식 조건을 갖는 최소제곱 문제를 최소거리계획법 문제(LDP, Least distance programming)로 등가변형하는 단계; 및,

상기 최소거리 계획법 문제를 음이 아닌 최소제곱 문제(NNLS, Nonnegative least squares)로 등가변형하는 단계를 포함하는 것을 특징으로 하는 이차 계획법을 이용한 예측제어 방법.

청구항 8

청구항 1에 있어서,

상기 데이터 파일은 상기 제어시스템과 독립적으로 마련되는 제1 컴퓨팅 수단에 의해 생성되고,

상기 제어 데이터는 상기 제어시스템에 장착된 제2 컴퓨팅 수단에 의해 상기 제어시스템의 매 스텝마다 실시간으로 연산되어 생성되는 것을 특징으로 하는 이차 계획법을 이용한 예측제어 방법.

청구항 9

PC가 제어시스템의 제약조건을 만족하면서 주어진 비용함수를 최소화하기 위하여 특이값 분해를 이용하여 실시간 연산을 수행하는 코드에 포함되며 상기 제어시스템의 스텝에 따라 변화되지 않는 데이터 파일을 생성하는 단계;

상기 PC가 상기 데이터 파일을 상기 실시간 연산을 수행하는 코드와 컴파일하여 제어연산을 수행하는 실행 파일을 생성하는 단계; 및,

마이크로콘트롤러가 상기 생성된 실행 파일을 수행하여 등가변환된 이차계획법 문제를 연산하여 상기 제어시스템의 스텝에 따라 변화되는 제어데이터를 생성함으로써 실시간 제어 연산을 수행하는 단계

를 포함하는 이차 계획법을 이용한 예측제어 방법.

명 세 서

기 술 분 야

[0001] 본 발명은 이차계획법을 이용한 예측제어 방법에 관한 것으로, 더욱 상세하게는 실제 마이크로 컨트롤러 시스템에 장착 가능하여 연산속도가 향상되며 실시시간 계산 성능이 향상된 이차계획법을 이용한 예측제어 방법에 관한 것이다.

배 경 기 술

[0002] 예측 제어는 산업계 및 학계에서 많은 관심을 받아왔고 특히 응답이 비교적 느린 공정 시스템의 제어에 적합하다고 알려져 있다. 예측 제어는 모델 정보를 이용하여 시스템의 미래 출력을 예측하고 예측된 출력에 의해 발생

한 비용함수가 최소가 되게끔 시스템의 제어입력을 최적화 연산을 통해 계산하는 제어방식이다.

[0003] 예측제어는 가정하고 있는 모델식의 종류에 따라 다양한 형태로 연구되었다. 그중 대표적인 기법들로는 ARMA 모델을 가정하는 GPC, 스텝응답 모델을 사용하는 DMC(Dynamic matrix control), 상태공간 모델을 사용하는 MPC(Model predictive control) 등이 있다. 예측제어는 비용함수를 최소로 만든다는 최적제어의 개념과 더불어 입력과 출력에 나타나는 제약조건을 조직적으로 고려할 수 있는 장점을 가진다는 특징이 있다.

[0004] 예측제어의 해를 구하는 문제는 선형행렬부등식(LMI, Linear matrix inequality) 문제로 귀결되는 일부 예측제어기법을 제외하고는 대부분 QP(이차계획법) 문제로 귀결된다. 따라서 QP(이차계획법) 문제의 해법은 예측제어기의 구현에 필수적인 요소이다.

[0005] 다수의 발명들이 예측제어의 이론 또는 응용문제를 다루었으며, 모의실험을 통해 효용성을 검증하는 접근법을 쓰고 있다. 하지만 예측제어를 C-code 등으로 구현하여 실제 마이크로 컨트롤러에 탑재하는 것과 관련된 연구는 상대적으로 관심을 많이 받지 못했다. 한가지 이유로는 예측제어 알고리즘을 구현하기 위해 필요한 QP(이차계획법)의 해법에 대한 C-code 알고리즘을 마이크로 컨트롤러에 탑재할 만큼 간결한 것을 구하기 힘들고 QP(이차계획법) 문제의 풀이에 소요되는 시간이 길어서 비교적 빠른 시스템에는 실시간 적용가능성이 떨어지기 때문이다. 그 결과 예측제어를 실시간으로 적용하는 데 한계가 있었다. 본 발명에서는 상기한 문제점을 해결할 수 있는 방안을 제안한다.

발명의 내용

해결하려는 과제

[0006] 본 발명이 해결하고자 하는 기술적 과제는 예측제어 시스템의 계산 효율을 크게 향상시켜서 예측제어를 실시간으로 적용할 수 있는 이차계획법을 이용한 예측제어 방법을 제공하는데 목적이 있다.

과제의 해결 수단

[0007] 상기한 과제를 해결하기 위한 본 발명에 따른 이차계획법을 이용한 예측제어 방법은,

[0008] 제어시스템의 제약조건을 만족하면서 주어진 비용함수를 최소화하기 위하여 특이값 분해를 이용하여 실시간 연산을 수행하는 코드에 포함되며 상기 제어시스템의 스텝에 따라 변화되지 않는 데이터 파일을 생성하는 단계; 상기 데이터 파일을 상기 실시간 연산을 수행하는 코드와 컴파일하여 제어연산을 수행하는 실행 파일을 생성하는 단계; 상기 생성된 실행 파일을 수행하여 등가변환된 이차계획법 문제를 연산하며, 상기 제어시스템의 스텝에 따라 변화되는 제어 데이터를 생성함으로써 실시간 제어 연산을 수행하는 단계를 포함하는 것을 특징으로 한다.

[0009] 본 발명의 일 양상에 따르면, 상기 비용함수(J)는 하기 수학식 (1)로부터 연산되고, 상기 제약조건은 하기 수학식 (2)로부터 연산될 수 있다.

[0010] 수학식(1);

$$Minimize\ J=\sum_{j=1}^{N_p}\| w_{k+j}-\widehat{y_{k+j}}\|^2+\sum_{j=0}^{N_c-1}\| \triangle u_{k+j}\|_{\Lambda}^2,$$

[0011] 여기서, w는 기준입력, $\hat{y}$는 예측 출력, $\triangle u$는 제어증분(control increment)를 나타내며, N_p는 예측구간(prediction horizon), N_c는 제어구간(control horizon), Λ는 제어증분에 대한 가중치를 나타내는 대각행렬이

고,

[0012] 수학식(2); $u_{\min} \le u_{k+j} \le u_{\max},\ \Delta u_{\min} \le \Delta u_{k+j} \le \Delta u_{\max}$,

[0013] 여기서, 아래첨자는 시점을 나타내는 것으로, k는 현재 시점, u는 입력, u_{k+j}는 현재 시점으로부터 j스텝 이후의 입력을 의미함.

[0014] 본 발명의 일 양상에 따르면, 상기 이차계획법 문제는 하기 수학식 (4)의 조건에 따른 하기 수학식 (3)으로 표현될 수 있다.

[0015] 수학식(3); $$Minimize\, J(x) = \frac{1}{2}x^T Px + q^T x$$,

[0016] 수학식(4); $Ax \le b$

[0017] 여기서, J는 비용함수, x는 최적화 변수, P 및 A는 행렬, q 및 b는 소정의 벡터이다.

[0018] 본 발명의 일 양상에 따르면, 상기 데이터 파일은 상기 P 및 상기 A와 관련된 데이터를 포함하고, 상기 제어 데이터는 상기 q 및 상기 b와 관련된 데이터를 포함한다.

[0019] 상기 P는 하기 수학식 (5)로부터 산출되고, 상기 A는 하기 수학식 (6)으로 표현될 수 있다.

[0020] 수학식 (5); $P = G^T G + \Xi$,

[0021] 여기서, $\Xi = diag[\Lambda, \Lambda, \cdots, \Lambda]$ 이고, Λ는 제어증분에 대한 가중치를 나타내는 대각행렬이고, G는 시스템의 모델정보를 이용하여 계산한 예측행렬이고,

[0022] 수학식 (6); $$A = \begin{bmatrix} I_D \\ -I_D \\ I_L \\ -I_L \end{bmatrix}$$,

[0023] 여기서, $$I_D = \begin{bmatrix} I_m & 0 & \cdots & 0 \\ 0 & I_m & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & I_m \end{bmatrix},\ I_L = \begin{bmatrix} I_m & 0 & \cdots & 0 \\ I_m & I_m & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ I_m & I_m & \cdots & I_m \end{bmatrix}$$ 임.

[0024] 한편, 상기 q는 하기 수학식 (7)로부터 산출되고, 상기 b는 하기 수학식 (8)로부터 산출될 수 있다.

[0025] 수학식 (7): $q = -2(W_k - f_k)^T G$

[0026] 여기서, w는 기준입력, f_k는 자유응답(free response)으로써 과거의 데이터가 미래의 출력에 기여하는 양, G는 시스템의 모델정보를 이용하여 계산한 예측행렬이고,

[0027] 수학식 (8): $$b = \begin{bmatrix} I_V u_{\max} \\ -I_V u_{\min} \\ I_V(u_{\max} - u_{k-1}) \\ -I_V(u_{\min} - u_{k-1}) \end{bmatrix}$$

[0028] 여기서, $I_V = \begin{bmatrix} I_m \\ I_m \\ \vdots \\ I_m \end{bmatrix}$, k는 현재 시점, u는 입력임.

[0029] 본 발명의 일 양상에 따르면, 상기 이차계획법 문제의 연산은,

[0030] 상기 이차계획법 문제를 부등식 조건을 갖는 최소제곱 문제(LSI, Least squares with inequality)로 등가변형하는 단계; 상기 부등식 조건을 갖는 최소제곱 문제를 최소거리계획법 문제(LDP, Least distance programming)로 등가변형하는 단계; 및, 상기 최소거리 계획법 문제를 음이 아닌 최소제곱 문제(NNLS, Nonnegative least squares)로 등가변형하는 단계를 포함한다.

[0031] 본 발명의 일 양상에 따르면, 상기 데이터 파일은 상기 제어시스템과 독립적으로 마련되는 제1 컴퓨팅 수단에 의해 생성되고, 상기 제어 데이터는 상기 제어시스템에 장착된 제2 컴퓨팅 수단에 의해 상기 제어시스템의 매 스텝마다 실시간으로 연산되어 생성되는 것을 특징으로 한다.

발명의 효과

[0032] 종래에는 이차계획법 문제의 해를 구하기 위해서는 특이값 분해와 같은 매우 복잡한 계산이 필요하게 되고, 그 결과 예측제어를 실시간으로 적용하는 데 한계가 있었다. 그러나, 본 발명의 이차계획법을 이용한 예측제어 방법에 따르면, 복잡한 계산이 필요하지만 제어시스템의 스텝에 따라 변화되지 않는 데이터는 별도의 컴퓨팅 장치를 통해 미리 생성하고, 비교적 계산이 간단하고 제어시스템의 스텝에 따라 변화되는 데이터는 제어시스템에 장착된 컴퓨팅 장치를 통해 생성하고 이를 컴파일하여 실시간 제어 연산을 수행할 수 있게 되어, 제어시스템의 계산 효율을 크게 향상시킬 수 있고, 그 결과 예측제어를 실시간으로 적용할 수 있게 된다.

도면의 간단한 설명

[0033] 도 1은 본 발명의 일 실시예에 따른 이차계획법을 이용한 예측제어 방법을 도시한 순서도이다.

도 2는 본 발명의 일 실시예에 따른 이차계획법을 이용한 예측제어 방법을 구현하기 위한 알고리즘 개요도이다.

도 3은 본 발명의 일 실시예에 따른 이차계획법을 이용한 예측제어 방법을 도시한 개념도이다.

발명을 실시하기 위한 구체적인 내용

[0034] 이하에서 첨부된 도면들을 참조하여 본 발명에 따른 이차계획법을 이용한 예측제어 방법에 대해 상세하게 설명한다. 이때 도면에 도시되고 또 이것에 의해서 설명되는 본 발명의 구성과 작용은 적어도 하나의 실시예로서 설명되는 것이며, 이것에 의해서 본 발명의 기술적 사상과 그 핵심 구성 및 작용이 제한되지는 않는다.

[0035] 본 발명에서 사용되는 용어는 본 발명에서의 기능을 고려하면서 가능한 현재 널리 사용되는 일반적인 용어를 선택하였으나, 이는 당해 기술분야에 종사하는 기술자의 의도 또는 관례 또는 새로운 기술의 출현 등에 따라 달라질 수 있다. 또한, 특정한 경우는 출원인이 임의로 선정한 용어도 있으며, 이 경우 해당되는 발명의 설명 부분에서 상세히 그 의미를 기재할 것이다. 따라서 본 발명에서 사용되는 용어는 단순한 용어의 명칭이 아닌 그 용어가 가지는 의미와 본 발명의 전반에 걸친 내용을 토대로 정의되어야 함을 밝혀두고자 한다.

[0036] 이하, 도면을 참조하여 본 발명에 따른 이차계획법을 이용한 예측제어 방법을 더욱 상세하게 설명한다.

[0037] 본 발명의 이차계획법을 이용한 예측제어 방법은, 복잡한 계산이 필요하지만 제어시스템의 스텝에 따라 변화되지 않는 데이터는 별도의 컴퓨팅 장치를 통해 미리 생성하고, 비교적 계산이 간단하고 제어시스템의 스텝에 따라 변화되는 데이터는 제어시스템에 장착된 컴퓨팅 장치를 통해 생성하고 이를 컴파일하여 실시간 제어 연산을 수행하는 것을 주요 특징으로 한다.

[0038] 이를 위해, 본 발명의 일실시예에 따른 이차계획법을 이용한 예측제어 방법은, 먼저 제어시스템의 제약조건을 만족하면서 주어진 비용함수를 최소화하기 위하여 특이값 분해를 이용하여 실시간 연산을 수행하는 코드에 포함되며 상기 제어시스템의 스텝에 따라 변화되지 않는 데이터 파일을 생성한다. (S100)

[0039] 제어시스템의 제약조건을 만족하면서 주어진 비용함수를 최소화하기 위한 비용함수는 아래 식 (1)과 같이 표현되고, 제약조건은 아래 식 (2)와 같이 표현된다.

[0040] 식 (1): $$Minimize\ J=\sum_{j=1}^{N_p}\| w_{k+j}-\hat{y}_{k+j}\|^2+\sum_{j=0}^{N_c-1}\| \triangle u_{k+j}\|_{\Lambda}^2$$

[0041] 식 (2): $$u_{\min}\le u_{k+j}\le u_{\max},\ \triangle u_{\min}\le \triangle u_{k+j}\le \triangle u_{\max}$$

[0042] 여기서, 변수에 사용된 아래첨자는 시점을 나타내는 것으로 가정하고 현재 시점을 k라고 가정하자. 입력을 u라고 한다면 u_{k+j}는 현재 시점으로부터 j스텝 이후의 입력을 의미한다. 예측제어에서는 매 스텝마다 상기 식 (2)와 같은 제약조건을 만족하면서 주어진 비용함수를 최소화하는 제어량을 계산한다.

[0043] 상기 식 (1) 및 식 (2)에서 W는 기준입력, $\hat{y}$는 예측 출력, $\triangle u$는 제어증분(control increment)를 나타내며 N_p는 예측구간(prediction horizon), N_c는 제어구간(control horizon)이다. 일반적으로 $N_p \ge N_c$이고 Λ는 제어증분에 대한 가중치를 나타내는 대각행렬이다.

[0044] 예측제어에서는 매 스텝마다 $k+N_c-1$까지 제어증분량을 계산하지만 사용되는 것은 $\triangle u_k$로써 실제 시스템에 사용되는 제어입력과는 $u_k = u_{k-1}+\triangle u_k$의 관계가 성립한다. 현재로부터 N_p스텝 이후까지의 예측 출력을 행렬과 벡터를 이용하여 나타내면 아래 식 (3)과 같이 나타낼 수 있다.

[0045] 식 (3) : $$\hat{Y}_k = G\triangle U_k + f_k$$

[0046] 상기 식 (3)은 다시 하기 식 (4) 및 식 (5)와 같이 나타낼 수 있다.

[0047] 식 (4) : $$\hat{Y}_k = \begin{bmatrix} \hat{y}_{k+1}^T & \hat{y}_{k+2}^T & \cdots & \hat{y}_{k+N_p}^T \end{bmatrix}^T$$

[0048] 식 (5) : $$\triangle U_k = \begin{bmatrix} \triangle U_k^T & \triangle U_{k+1}^T & \cdots & \triangle U_{k+N_c-1}^T \end{bmatrix}$$

[0049] 상기 식 (3)에서의 G는 시스템의 모델정보를 이용하여 계산한 예측행렬이고, f_k는 free response로써 과거의

data가 미래의 출력에 기여하는 양을 나타낸다. 여기서 G에는 아래 첨자가 붙지 않았는데 이것은 매 스텝마다 G의 값이 변하지 않기 때문이다. G와 f_k는 DMC, GPC, MPC 등 예측제어의 종류에 따라 각각의 계산 방식이 존재한다. 미래의 기준입력 벡터를 아래 식 (6)과 같이 정의하면,

[0050] 식 (6) : $$W_k = \left[w_{k+1}^T \; w_{k+2}^T \; \cdots \; w_{k+N_p}^T \right]^T$$

[0051] 예측제어에서 최소화하고자 하는 비용함수는 $\triangle U_k$의 함수가 되어 하기 식 (7)과 같이 나타낼 수 있다.

[0052] 식 (7) : $$J(\triangle U_k) = (W_k - \hat{Y}_k)^T (W_k - \hat{Y}_k) + \triangle U_k^T \Xi \triangle U$$

[0053] 이때, $\Xi = diag[\Lambda, \Lambda, \cdots, \Lambda]$ 로 정의된다.

[0054] 비용함수인 식 (1)을 풀어서 $\triangle U_k$에 대한 2차형으로 정리하면 식 (8)과 같다.

[0055] 식 (8) : $$J(\triangle U_k) = \triangle U_k^T [G^T G + \Xi \triangle U_k - 2(W_k - f_k)^T G \triangle U_k + (W_k - f_K)^T (W_k - f_k)$$

[0056] 시스템이 갖는 입력 제약조건을 다시 나타내기 위하여, 식 (9)와 같이 I_D, I_L, I_V를 정의한다.

[0057] 식 (9) : $$I_D = \begin{bmatrix} I_m & 0 & \cdots & 0 \\ 0 & I_m & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & I_m \end{bmatrix}, \; I_L = \begin{bmatrix} I_m & 0 & \cdots & 0 \\ I_m & I_m & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ I_m & I_m & \cdots & I_m \end{bmatrix}, \; I_V = \begin{bmatrix} I_m \\ I_m \\ \vdots \\ I_m \end{bmatrix}$$

[0058] 이때, 식 (9)는 예측구간 및 제어구간의 길이에 따라 적절한 차원을 가진다고 가정하자. 식 (2)에 주어진 입력에 대한 제약조건을 행렬과 벡터를 이용하여 식 (10)과 같이 나타낼 수 있다.

[0059] 식 (10) : $$\begin{aligned} I_D \triangle U_k &\le I_V \triangle U_{\max}, \;\; -I_D \triangle U_k \le -I_V \triangle U_{\min} \\ I_L \triangle U_k &\le I_V (u_{\max} - u_{k-1}), \;\; -I_L \triangle U_k \le -I_V (u_{\min} - u_{k-1}) \end{aligned}$$

[0060] 상기 식 (10)을 $Ax \le b$의 형태로 나타내면 하기 식 (11)과 같이 나타낼 수 있다.

[0061] 식 (11) : $$\begin{bmatrix} I_D \\ -I_D \\ I_L \\ -I_L \end{bmatrix} \triangle U_k \le \begin{bmatrix} I_V u_{\max} \\ -I_V u_{\min} \\ I_V (u_{\max} - u_{k-1}) \\ -I_V (u_{\min} - u_{k-1}) \end{bmatrix}$$

[0062] 상기 식 (7)에서 마지막 항인 $(W_k - f_k)^T (W_k - f_k)$는 $\triangle U_k$의 함수가 아니므로 최적화와는 무관하다. 따라서 상기

식 (2)를 만족시키며 상기 식 (8)을 최소화시키는 해 $\triangle U_k^*$는 아래 식(12), 식 (13)의 P, A, q, b의 변수로 이루어진 이차계획법(Qp, Quadratic programming) 문제의 해가 된다. 이 때, 본 단계는 하기 식(12), 식(13)의 P, A에 대한 행렬 데이터를 연산하여 데이터 파일을 생성할 수 있다.

[0063] 식 (12) : $$Minimize\, J(x) = \frac{1}{2}x^T Px + q^T x$$

[0064] 식 (13) : $$Ax \leq b$$

[0065] 여기서, J는 비용함수, x는 최적화 변수, P 및 A는 행렬, q 및 b는 소정의 벡터이다.

[0066] 식 (14) : $$P = G^T G + \Xi$$

[0067] 식 (15) : $$A = \begin{bmatrix} I_D \\ -I_D \\ I_L \\ -I_L \end{bmatrix}$$

[0068] 상기 식 (14) 및 식 (15)를 참조하면, P와 A는 제어 시스템의 모델에 따라 변하지만 예측제어 시스템의 스텝에 따라서는 변하지 않는다는 것을 알 수 있다. 따라서, 상기 P와 A는 실시간으로 연산되지 않고 컴퓨팅 장치의 연산 프로그램 (예를 들어, Matlab)을 통해 미리 연산될 수 있다.

[0069] 본 발명의 제 1 데이터 파일은 상기 P와 A의 행렬 데이터일 수 있다. P와 A의 행렬 데이터는 제어시스템과 독립적으로 마련되는 제1 컴퓨팅 수단에 의해 생성될 수 있다.

[0070] 다음으로, 상기 데이터 파일을 상기 실시간 연산을 수행하는 코드와 컴파일하여 제어연산을 수행하는 실행 파일을 생성한다. (S200)

[0071] 다음으로,생성된 실행 파일을 수행하여 등가변환된 이차계획법 문제를 연산하여 상기 제어시스템의 스텝에 따라 변화되는 제어데이터를 생성함으로써 실시간 제어 연산을 수행한다. (S300)

[0072] 상기 제어 데이터는 하기 수학식 16 및 수학식 17의 q, b에 대한 데이터로 상기 제어 데이터는 제어시스템의 매 스텝에 따라 변하는 값이다. 따라서, 상기 제어 데이터는 실시간으로 자동제어 시스템의 스텝에 따라 생성된다.

[0073] 식 (16) : $$q = -2(W_k - f_k)^T G$$

[0074] 식 (17) : $$b = \begin{bmatrix} I_V u_{\max} \\ -I_V u_{\min} \\ I_V(u_{\max} - u_{k-1}) \\ -I_V(u_{\min} - u_{k-1}) \end{bmatrix}$$

[0075] 상기 식 (16) 및 식 (17)을 참조하면, q와 b는 예측제어 시스템의 스텝에 따라서는 변한다는 것을 알 수 있다. 따라서, 상기 q와 b는 실시간으로 연산되며, 제어시스템에 장착된 제2 컴퓨팅 수단(예를 들어, 마이크로 컨트롤러(MCU))에 의해 매 스텝마다 실시간으로 연산되어 제어 데이터를 실시간으로 생성한다.

[0076] 한편, 이차계획법(Qp, Quadratic programming) 문제는 하기 식 (18)과 이에 대한 제약조건인 식 (19)를 통해 정의될 수 있다.

[0077] 식 (18) : $$Minimize\, J(x) = \frac{1}{2}x^T Px + q^T x$$

[0078] 식 (19) : $$Ax \leq b$$

[0079] 이때, $P \geq 0$이면 이차계획법(Qp, Quadratic programming) 문제는 전역해(global solution)가 보장되고, $P > \infty$이면 유일해가 존재한다. 예측제어에서의 제어입력 계산 문제는 $P > 0$인 QP 문제로 귀결된다. 이 발명에서는 $P > 0$인 경우에 대한 이차계획법(Qp, Quadratic programming) 문제의 해를 제시하고 이를 예측제어 알고리즘에 적용할 때 계산효율성을 향상시키는 방법에 관한 것이다.

[0080] 먼저, $P > 0$인 경우의 이차계획법(Qp, Quadratic programming) 문제에 대한 해를 유도하기 위해 식 (20) 및 제약조건인 식 (21)과 같은 부등식 조건을 갖는 최소제곱 문제(LSI, Least squares with inequality)로 등가변형한다. (S111)

[0081] 식 (20) : $$Minimize\, \hat{J}(x) = \| \sqrt{P}x + \sqrt{P^{-1}}q \|$$

[0082] 식 (21) : $$Ax \leq b$$

[0083] 여기서 비용함수 $\hat{J}(x)$에 대해 식 (22)가 성립하므로 QP 문제의 비용함수 $J(x)$와 LSI 문제의 비용함수 $\hat{J}(x)$는 식 (23)과 같은 관계가 성립한다.

[0084] 식 (22) : $$\hat{J}^2(x) = x^T Px + 2q^T x + q^T P^{-1} q$$

[0085] 식 (23) : $$J(x) = \frac{1}{2}[\hat{J}^2(x) - q^T P^{-1} q]$$

[0086] 이때, $\hat{J}(x) \geq 0$이고, J는 $\hat{J}$에 대해 단조 증가함수이므로 결국 $\hat{J}$가 최소일 때 J도 최소값을 갖게 된다. 즉, 이차계획법(Qp, Quadratic programming) 문제는 부등식 조건을 갖는 최소제곱 문제(LSI, Least squares with inequality)로 등가변형 될 수 있다.

[0087] 다음으로, 상기 등가변형된 부등식 조건을 갖는 최소제곱 문제(LSI, Least squares with inequality)을 최소거리계획법 문제(LDP, Least distance programming)로 등가변형한다. (S112)

[0088] $\bar{q} = P^{-1}q$라고 하면 $\hat{J} = \| \sqrt{P}(x + \bar{q}) \|$ 로 나타낼 수 있다. 특이값 분해를 이용하여 $\sqrt{P} = USV^T$로 나타낼 수 있다고 가정해보자. P>0이므로 S는 항상 역행렬이 존재한다. U는 단위직교행렬이므로 이로부터 식 (24)가 성립함을 알 수 있다.

[0089] 식 (24) : $\hat{J}(x) = \| USV^T(x+\bar{q}) \| = \| SV^T(x+\bar{q}) \|$

[0090] 여기서 $z = SV^T(x+\bar{q})$, $\bar{A} = -AVS^{-1}$, $\bar{b} = -A\bar{q} - b$ 라고 하면 최적화 문제는 식 (25) 및 제약조건인 식 (26)과 같은 최소거리계획법 문제(LDP, Least distance programming)로 변형될 수 있다.

[0091] 식 (25) : $Minimize \quad \| z \|$

[0092] 식 (26) : $Ax \leq b$

[0093] 최소거리계획법 문제(LDP, Least distance programming)의 해를 z^*라고 했을 때 z^*는 음이 아닌 최소제곱 문제(NNLS, Nonnegative least squares)를 풀어 그 해를 통해 구할 수 있다.

[0094] 다음으로, 음이 아닌 최소제곱 문제(NNLS, Nonnegative least squares)의 해를 통해 상기 최소거리계획법 문제(LDP, Least distance programming)의 해를 구한다. (S113)

[0095] 음이 아닌 최소제곱 문제(NNLS, Nonnegative least squares)는 식 (27)과 이의 제약조건인 식 (28)로 등가변형된다.

[0096] 식 (27) : $Minimize \quad \| Qd - r \|$

[0097] 식 (28) : $d \geq 0$

[0098] 이때, 상기 식 (27) 및 식 (28)에서 $Q = \begin{bmatrix} \bar{A}^T \\ \bar{b}^T \end{bmatrix}$, $r = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}$ 이다.

[0099] 상기 식 (27) 및 식 (28)로 표현된 음이 아닌 최소제곱 문제(NNLS, Nonnegative least squares)의 해를 d^*라고 하면 최소거리계획법 문제(LDP, Least distance programming) 의 해는 식 (29)로 구할 수 있다.

[0100] 식 (29) : $z^* = \dfrac{\bar{A}^T d}{(1 - b^T d)}$

[0101] 상기 식 (29)의 z^*로부터 원래의 이차계획법 문제(Qp, Quadratic programming)의 해 x^*를 식 (30)과 같이 구할 수 있다.

[0102] 식 (30) : $x^* = VS^{-1}z^* - \bar{q} = VS^{-1}z^* - P^{-1}q$

[0103] P, q, A, b가 주어졌을 때 QP의 해를 얻기 위해 필요한 계산들을 열거하면 다음과 같다.

[0104] (a) 행렬간의 곱, 행렬과 벡터의 곱, 벡터의 덧셈 및 뺄셈

[0105] (b) $\sqrt{P}$ 에 대한 특이값 분해

[0106] (c) P^{-1} 를 계산하기 위한 역행렬 계산

[0107] (d) NNLS의 풀이

[0108] 이 중에서 (a)에 대한 계산은, 예를 들어, C-code로 쉽게 구현 가능하다. 그러나, (b), (c), (d)에 대한 계산을 C-code로 구현하기 위해서는 복잡하고 긴 알고리즘이 필요하다. (b)와 (c)에 해당하는 계산은 둘다 행렬의 특이값 분해를 통해 계산할 수 있고 NNLS의 해는 "C. L. Lawson and R. J. Hanson, Solving least squares problems, Prentice-Hall, Englewood Cliffs, New Jersey, 1974."에서 제시된 알고리즘을 사용하여 계산할 수 있다.

[0109] (b)와 (c)의 계산이 특이값 분해를 통해 계산 가능한 것을 살펴 본다. P가 식 (31)과 같이 특이값 분해된다고 가정하면,

[0110] 식 (31) : $P = U\Sigma V^T$

[0111] P>0이므로, Σ>0이고 U=V가 성립한다. U는 단위직교행렬이므로, $UU^T = I$ 이다. 하기 식 (32)로부터 $\sqrt{P}$ 는 식 (33)과 같이 나타낼 수 있다.

[0112] 식 (32) : $U\sqrt{\Sigma}V^T(U\sqrt{\Sigma}V^T) = U\Sigma V^T = P$

[0113] 식 (33) : $\sqrt{P} = USV^T,\ S = \sqrt{\Sigma}$

[0114] 여기서, Σ는 대각행렬이므로, S역시 대각행렬이고 주대각선 성분은 Σ의 주대각선 성분에 대한 제곱근의 값을 갖기 때문에 쉽게 계산할 수 있다. P^{-1} 를 계산하기 위해서는 LU 분해에 이은 back substitution 방법이나 Cramer rule과 같은 역행렬 계산 방법을 사용할 수 있지만, 여기서는 P의 특이값 분해를 계속 사용하여 계산할 수 있다. 다음과 같이, $U\Sigma V^T(V\Sigma^{-1}U^T) = I$ 이 성립하므로 $P^{-1} = V\Sigma^{-1}U^T$ 임을 알 수 있다. Σ^{-1} 역시 주대각선 성분에 대한 역수만을 계산하면 되므로 쉽게 계산이 가능하다.

[0115] 전술한 QP 문제의 해는 (d) NNLS의 풀이 뿐만 아니라, (b), (c)의 특이값 분해와 같은 복잡한 계산이 필요하다. 하지만 선형시불변 시스템에 대한 예측제어를 구현하기 위해 매 스텝마다 풀이되어야 하는 QP 문제에서는 P와 A가 고정되어 있고 q와 b만 매 스텝마다 변하는 특징을 가진다. 따라서 이러한 특징을 활용하면 계산 효율을 크게 상승시킬 수 있는 알고리즘 구현이 가능하다. 매 스텝마다 값이 바뀌는 변수들을 나타내기 위해 아래첨자를 사용하기로 한다.

[0116] P가 고정되기 때문에 P에 대한 특이값 분해, 그리고 $\sqrt{P}$, P^{-1} 등의 계산은 예측제어시스템과 독립적으로 마련되는 제1 컴퓨팅 수단(offline으로 표시됨)으로 계산해도 된다. 예측제어 알고리즘에서 기본적인 사칙연산, 행렬간의 곱, 행렬과 벡터의 곱, 벡터의 덧셈 및 뺄셈, 그리고 NNLS등의 계산만이 제어시스템에 장착된 제2 컴퓨팅 수단(online으로 표시됨)으로 계산되어져야 한다. 도 2는 이러한 내용을 반영하여 제안한 예측제어 구현 알고리즘의 흐름도를 나타낸다.

[0117] 도 2에서 offline 계산 부분을 마이크로콘트롤러에서 수행한다면 특이값 분해에 대한 C-code 기반의 풀이법이 필요하다. 행렬에 대한 특이값 분해를 위한 C-code 알고리즘은 "W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Plannery, Numerical recipies in C, Cambridge University Press, Cambridge, 1988"을 참조할 수 있다.

[0118] NNLS의 풀이 알고리즘은 "C. L. Lawson and R. J. Hanson, Solving least squares problems, Prentice-Hall, Englewood Cliffs, New Jersey, 1974."에 제시되고 있으며 이에 대한 C-code는, 예를 들어, "http://hesperia.gsfc.nasa.gov/~schmahl/nnls/nnls.c"와 같이 인터넷을 통해 구할 수 있다.

[0119] 만약, offline 계산 부분을 마이크로콘트롤러에서 수행하는 것이 아니라 Matlab이나 CEMTool과 같은 소프트웨어를 이용하여 처리할 경우에는 역행렬, 특이값 분해 등의 계산이 지원되는 함수를 통해 손쉽게 계산된다. 따라서 offline 계산을 모두 수행한 후에 online 계산에서 활용되는 data들을 fprintf 함수를 활용하여 C-code header file에 출력하고 이것을 마이크로콘트롤러에서 수행될 application에서 사용하는 방식을 사용한다면 NNLS의 풀이에 대한 해만 C-code로 구현하면 예측제어 알고리즘을 마이크로콘트롤러에서 수행될 수 있도록 구현이 가능하다.

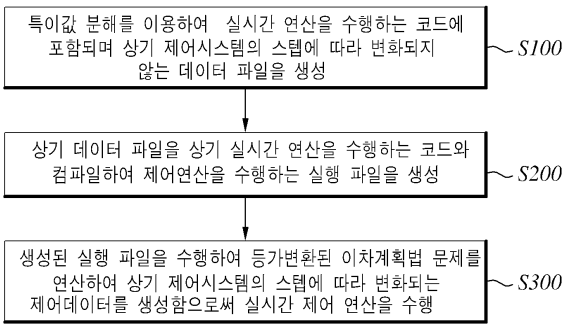
[0120] 도 3은 이러한 내용을 정리하여 담고 있는 개념도이다. offline 계산에 대한 부분은 제1 컴퓨팅 수단(예를 들어, PC)에서 수행되고 online 계산 부분은 제2 컴퓨팅 수단(예를 들어, 마이크로콘트롤러)에서 수행된다.

[0121] 종래에는 이차계획법 문제의 해를 구하기 위해서는 특이값 분해와 같은 매우 복잡한 계산이 필요하게 되고, 그 결과 예측제어를 실시간으로 적용하는 데 한계가 있었다. 그러나, 본 발명의 이차계획법을 이용한 예측제어 방법에 따르면, 복잡한 계산이 필요하지만 제어시스템의 스텝에 따라 변화되지 않는 데이터는 별도의 컴퓨팅 장치를 통해 미리 생성하고, 비교적 계산이 간단하고 제어시스템의 스텝에 따라 변화되는 데이터는 제어시스템에 장착된 컴퓨팅 장치를 통해 생성하고 이를 컴파일하여 실시간 제어 연산을 수행할 수 있게 되어, 제어시스템의 계산 효율을 크게 향상시킬 수 있고, 그 결과 예측제어를 실시간으로 적용할 수 있게 된다.

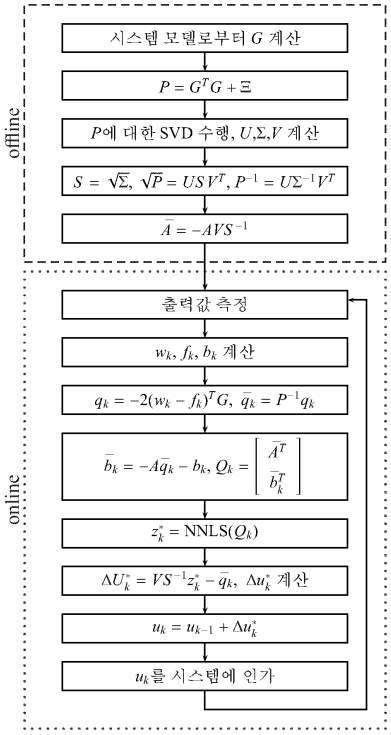
[0122] 이상에서 본 발명의 바람직한 실시예에 대해 도시하고 설명하였으나, 본 발명은 상술한 특정의 바람직한 실시예에 한정되지 아니하며, 청구범위에서 청구하는 본 발명의 요지를 벗어남이 없이 당해 발명이 속하는 기술분야에서 통상의 지식을 가진 자라면 누구든지 다양한 변형 실시가 가능한 것은 물론이고, 그와 같은 변경은 청구범위 기재의 범위 내에 있게 된다.

도면

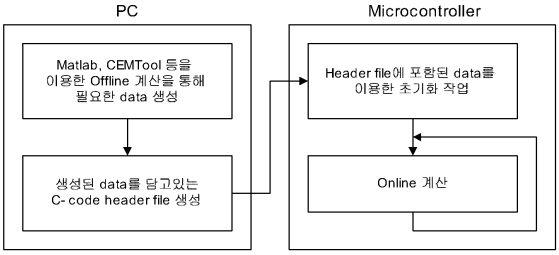
도면1



도면2


offline
시스템 모델로부터 G 계산
$P = G^T G + \Xi$
P에 대한 SVD 수행, U,Σ,V 계산
$S = \sqrt{\Sigma}$, $\sqrt{P} = USV^T$, $P^{-1} = U\Sigma^{-1}V^T$
$\bar{A} = -AVS^{-1}$
online
출력값 측정
w_k, f_k, b_k 계산
$q_k = -2(w_k - f_k)^T G$, $\bar{q}_k = P^{-1}q_k$
$\bar{b}_k = -A\bar{q}_k - b_k$, $Q_k = \begin{bmatrix} \bar{A}^T \\ \bar{b}_k^T \end{bmatrix}$
$z_k^* = \mathrm{NNLS}(Q_k)$
$\Delta U_k^* = VS^{-1}z_k^* - \bar{q}_k$, Δu_k^* 계산
$u_k = u_{k-1} + \Delta u_k^*$
u_k를 시스템에 인가

도면3


PC
Microcontroller
Matlab, CEMTool 등을
이용한 Offline 계산을 통해
필요한 data 생성
생성된 data를 담고있는
C-code header file 생성
Header file에 포함된 data를
이용한 초기화 작업
Online 계산