|  |  |
| --- | --- |
| 년도-학기 | 2021년 2학기 |
| 과목명 | 자동화프로그래밍 |

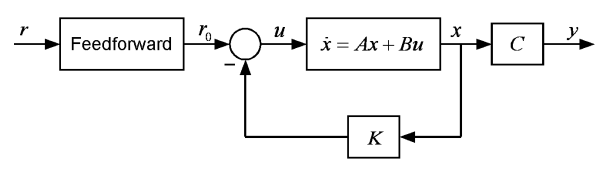
|  |  |
| --- | --- |
| **번호** | **실험 제목** |
| LAB11 | Discrete time estimator based controller |

|  |  |
| --- | --- |
| 실험 일자 | 2021년 11월 30일 |
| 제출자 이름 | 강\*\* |
| 제출자 학번 | 201803\*\*\*\* |
| 팀원 이름 |  |
|  |  |

**Chapter 1. 관련 이론(Theoretical Background)**

Discrete-Time 의 state space State-Space model은 continuous time 일 때와 크게 다르지 않다.

<Full-state Feedback>



Full-state Feedback controller는 시스템의 상태방정식이 일 때 상태변수인 모든 x를 제어입력 u에 반영하여 피드백 하는 것을 의미한다.

State Feedback Controller 식은 다음과 같이 나타낼 수 있다.

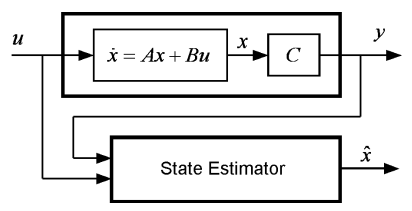
x(k+1)=Ax(k)+Bu(k)

u=-Kx(k)

x(k+1)=Ax(k)-BKx(k)=(A-BK)x(k)

<State Estimator>

State Estimator는 입력과 출력을 통해 을 추정하여 결국엔 x와 같은 을 얻어내는 것이다.



으로 설정하여 가 0이 되도록 한다.



텍스트, 손목시계이(가) 표시된 사진

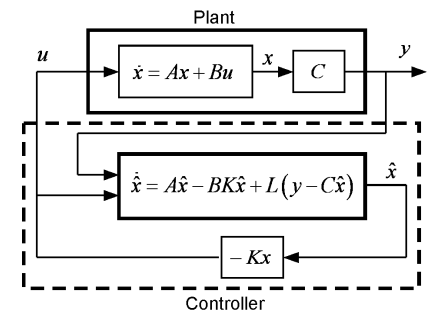
자동 생성된 설명

이므로 이고 일 때 가 된다.

이에 따라 estimator의 상태변수 방정식은 다음과 같아진다.



이를 이용한 estimator based controller는 다음과 같은 모습을 보인다.





이에 따라 estimator-based controller의 상태변수 방정식은 다음과 같아진다.





Non-zero reference를 가질 경우 식은 다음과 같다.

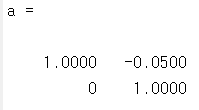
텍스트이(가) 표시된 사진

자동 생성된 설명

**Chapter 2. 실험 결과(Experimental Results)**

[200Hz]

State feedback 제어기를 구하기 위한 pole과 Estimator의 pole을 설정하고, Discrete State equation의 ***A, B*** 와 State estimator의 *K* = []와 L= []을 구한 결과는 다음과 같다.

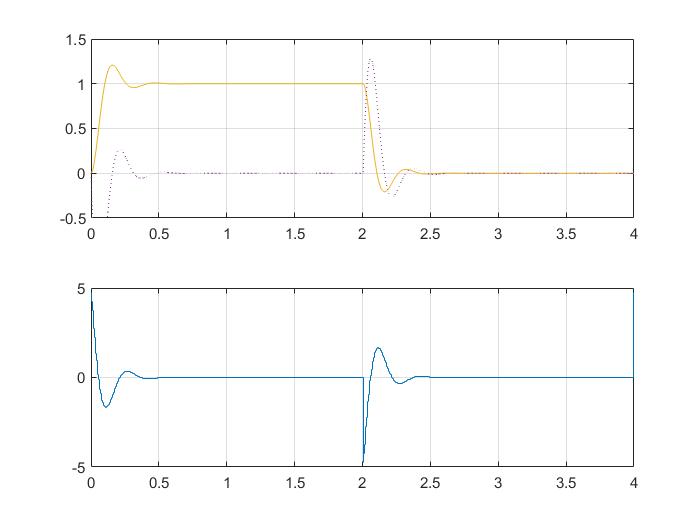
 테이블이(가) 표시된 사진

자동 생성된 설명 텍스트이(가) 표시된 사진

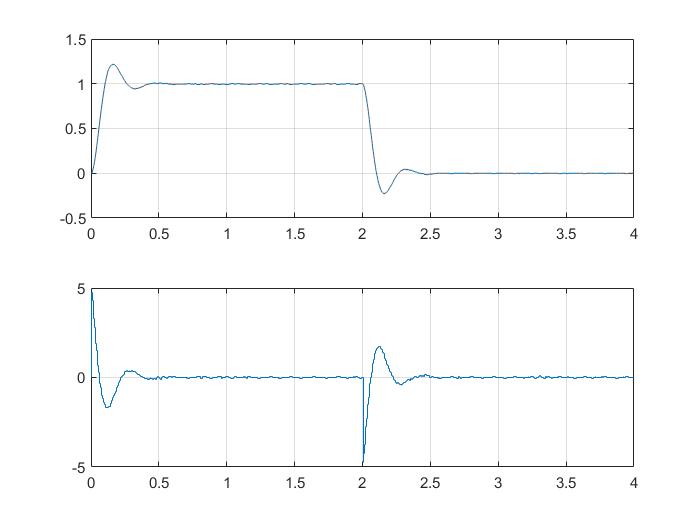
자동 생성된 설명 텍스트이(가) 표시된 사진

자동 생성된 설명

다음은 Matlab 시뮬레이션을 실행한 결과이다.

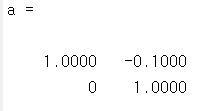


다음은 제어기를 구현하여 실험한 결과이다. 시뮬레이션 결과와 비슷한 것을 볼 수 있다.



[100Hz]

이전과 동일하게 pole 위치를 설정하고, Discrete State equation의 ***A, B*** 와 State estimator의 *K* 와 L을 구한 결과는 다음과 같다.

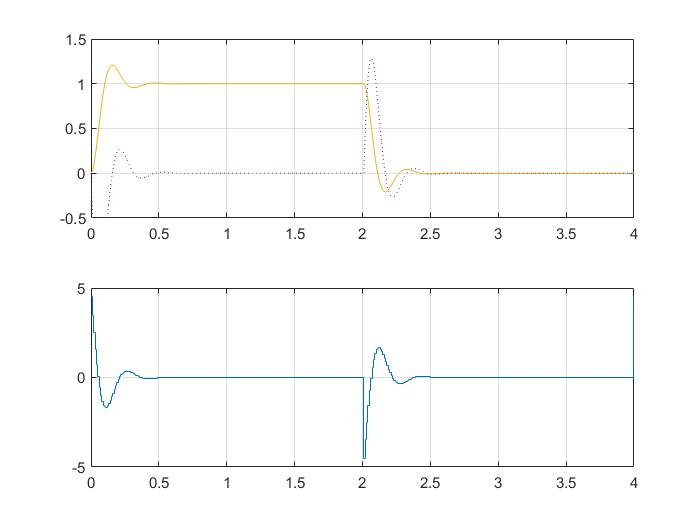
 테이블이(가) 표시된 사진

자동 생성된 설명 텍스트이(가) 표시된 사진

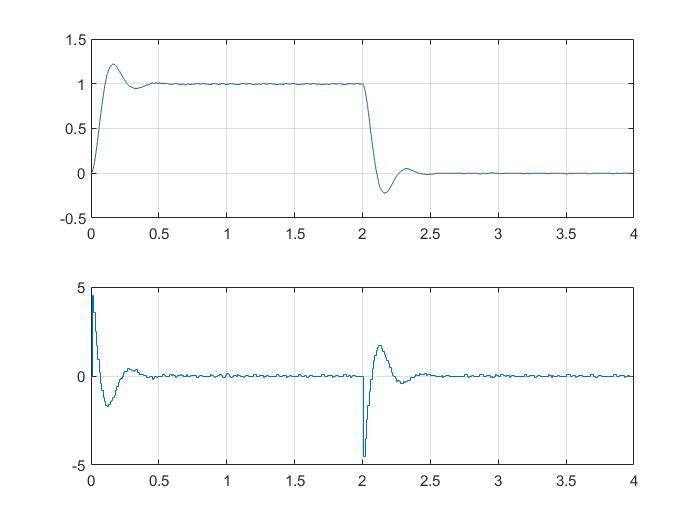
자동 생성된 설명 텍스트이(가) 표시된 사진

자동 생성된 설명

다음은 Matlab 시뮬레이션을 실행한 결과이다.

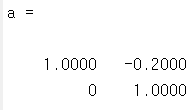


다음은 제어기를 구현하여 실험한 결과이다. steady state 부분에 노이즈와 같은 값들이 보이긴 하지만 시뮬레이션 결과와 비슷한 것을 볼 수 있다.



[50Hz]

이전과 동일하게 pole 위치를 설정하고, Discrete State equation의 ***A, B*** 와 State estimator의 *K*]와 L을 구한 결과는 다음과 같다.

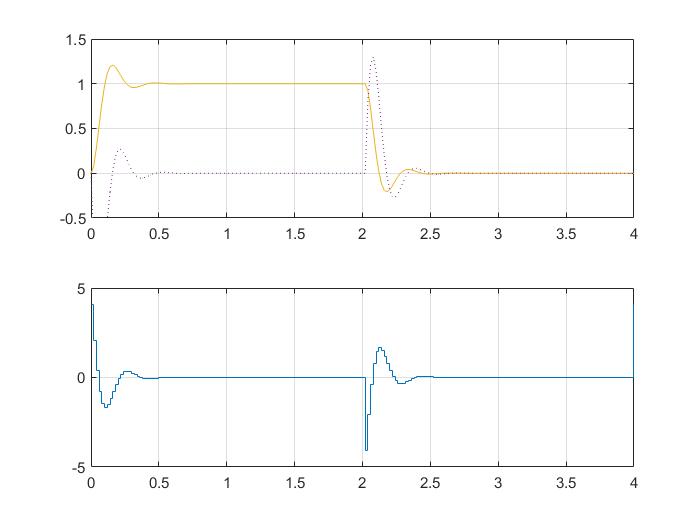
 테이블이(가) 표시된 사진

자동 생성된 설명 텍스트이(가) 표시된 사진

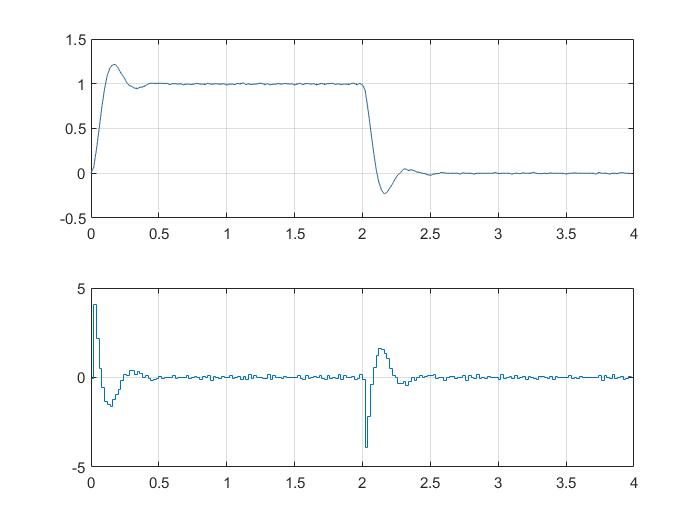
자동 생성된 설명 텍스트이(가) 표시된 사진

자동 생성된 설명

다음은 Matlab 시뮬레이션을 실행한 결과이다.

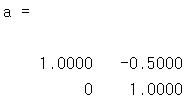


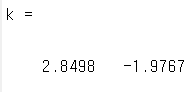
다음은 제어기를 구현하여 실험한 결과이다. 50Hz에서도 steady state 부분에 노이즈와 같은 값들이 보이긴 하지만 시뮬레이션 결과와 비슷한 것을 볼 수 있다.



[20Hz]

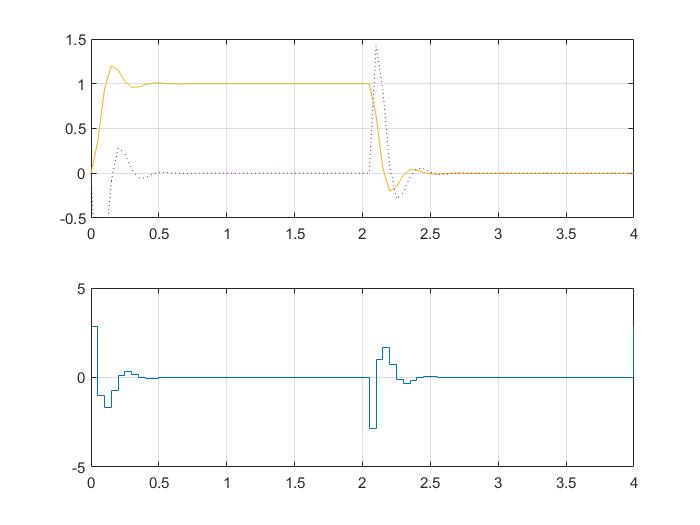
이전과 동일하게 pole 위치를 설정하고, Discrete State equation의 ***A, B*** 와 State estimator의 *K* 와 L을 구한 결과는 다음과 같다.

 테이블이(가) 표시된 사진

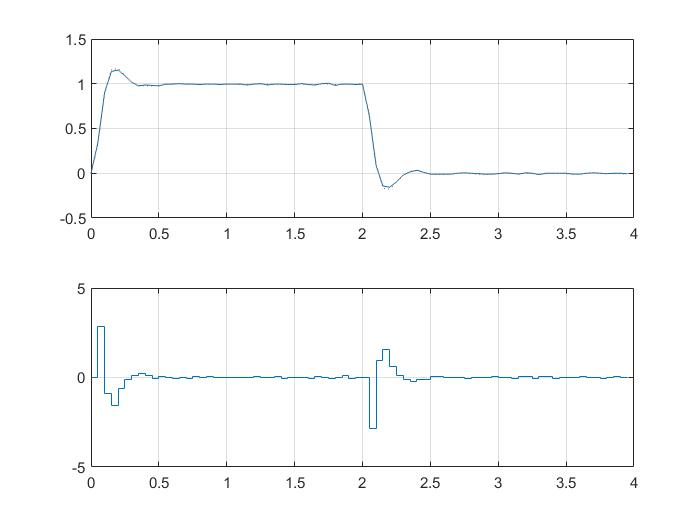
자동 생성된 설명  텍스트이(가) 표시된 사진

자동 생성된 설명

다음은 Matlab 시뮬레이션을 실행한 결과이다.



다음은 제어기를 구현하여 실험한 결과이다. 20Hz에서도 steady state 부분에 노이즈와 같은 값들이 보이긴 하지만 sampling frequency가 작아서 50Hz에 비해 비교적 노이즈가 적어 보인다. 전체적으로는 시뮬레이션 결과와 비슷한 것을 볼 수 있다.



**Chapter 3. 결론 및 Discussion**

과제를 수행하여 시뮬레이션 결과와 실험결과가 동일함을 확인하였고, state feedback의 pole이 -10+20j -10-20j이었기 때문에 출력에서 overshoot이 나타났음 또한 확인하였다.

이전 lab6과 7에서 했던 continuous time에서의 state feedback, estimator과 이론적 배경과 구현 과정이 비슷하다는 것을 실험을 통해서 알 수 있었다. 이전 과제에서는 코드를 직접 구현해야 했지만 이번에는 완성된 코드를 이용했기 때문에 어려움 없이 과제를 할 수 있었다. control 식과 추정기의 상태변수를 구하는 코드를 확인해보고, x\_hat을 구할 때 A matrix와 B matrix 값을 사용하여 구한다는 점이 continuous에서의 과제와 가장 다른 점이라고 생각했다.

**Appendix:**

**실험 조건을 변경하여 실행해보는 과제였기 때문에 소스코드는 생략하였습니다.**