|  |  |  |
| --- | --- | --- |
| **[EEC3600-001] 수치해석** | | |
| 소속: 전기전자공학부 | 학번: 12191529 | 이름: 장준영 |
| **Term Project** | | **Prob #2** |

1. **Problem**
2. 문제

|  |
| --- |
|  |
|  |

1. **Solution (a)**

|  |
| --- |
| 이 코드는 단순 vehicle 모델 기반 차량의 선형화된 조향 시스템을 시뮬레이션하기 위해, Euler-forward 방식으로 선형 동역학을 이산화하고, 시간 구간 전체에 걸쳐 제어 시퀀스가 상태에 미치는 영향을 행렬 형태로 구성한 것이다. 최종적으로 아래 수식을 만족하는 행렬 를 구성한다:  ■ **기본 파라미터 및 초기 상태 설정**    • 본 실험에서는 차량의 선형화된 조향 모델을 시뮬레이션하기 위해 10초의 시계열 구간을 설정하였다. 이산화 간격은 1초로 설정하였으며, 차량의 휠베이스 길이는 2.0m로 가정하였다. 시스템의 초기 상태는 로, 초기 위치는 원점의 하단에 위치하며 차량은 수평방향()을 향하고 있다. 제어 입력의 기준값은 로, 차량은 10m/s의 직진 속도를 유지하며 조향각은 0이다. 목표 상태는 로 설정하였으며, 이는 차량이 x축을 따라 100m 전진하고 y축으로는 2m 옮겨지도록 유도하는 것이다.  ■ **선형화 (Linearization)**    • 행렬 는 상태벡터 에 대한 편미분 Jacobian이다.  • 조향각이 0이므로 로 단순화된다.    • 행렬 는 입력 벡터 에 대한 편미분 Jacobian이다.  • 항이 이므로, delta에 대한 도함수가 포함된다.    • 값 자체, 즉 선형화 기준점에서의 시스템 응답이다.  • 주어진 vehicle 모델은 비선형 시스템이므로, 시스템을 선형 형태로 다루기 위해 지정된 평형점 ()에서 선형화를 수행하였다. 이를 통해 상태 변수에 대한 Jacobian 행렬 , 그리고 상수 항 를 각각 정의하였다. 이러한 선형화는 차량의 주행 경로가 평형점 부근에서 크게 벗어나지 않는다는 가정 하에서 유효하며, 선형 시스템 제어 이론을 적용할 수 있도록 한다.  ■ **Euler-forward 이산화 (Discretization)**    • 오일러 방식에 따라 등으로 구성된다.  • 선형화된 연속 시간 시스템을 실제 구현 가능한 이산 시간 시스템으로 변환하기 위해 Euler-forward 방식으로 이산화를 수행하였다. 시간 간격 을 기준으로, 이산화된 시스템은 다음과 같은 행렬로 표현된다:  이 과정을 통해 시간 이산 시스템 형태인 를 구성할 수 있게 된다.    ■ **행렬 F, G, H 구성**    • 시간 축을 따라 총 개의 상태벡터(각 3차원)를 수직으로 쌓은 형태. 총 행렬들이 구성된다.  • 시간 구간 전체(T=10)에 대해, 전체 상태 벡터를 수직으로 쌓은 형태로 표현하면 다음과 같다:  이를 위해 다음과 같은 3개의 블록 형태 행렬을 정의하였다:  • : 각 제어 입력 이 전체 상태에 미치는 영향.  • : 초기 상태 의 영향을 누적해서 표현.  • : 상수 항 의 시간 누적 영향을 표현.  이렇게 정의된 구조는 이후 최소제곱 기반 최적화를 통해 들을 결정하는 데 필요한 기반을 제공한다.  ■ **시간 루프를 통한 누적 행렬 계산**    • 이 루프는 시간 에 대해 반복하면서:  • 각 시간의 제어 입력 가 얼마나 영향을 미치는지를 에 누적.  • 초기 상태 의 영향을 에 반영.  • Offset 상수항 는 누적합으로 에 반영.  • 시간 구간 t=0부터 T-1까지 반복하면서, 각 time step에서 제어 입력이 전체 상태 벡터에 미치는 누적 효과를 행렬에 차곡차곡 더해가는 방식으로 구성하였다. 또한 각 시점에서 초기 상태 가 선형 시스템을 통해 어떻게 전달되는지를 행렬로 구성하였고, 상수항 는 행렬 거듭제곱과 누적합을 통해 에 반영하였다.  ■ **예시 실행 및 결과 확인**    • 는 전체 시간 동안의 상태들을 쌓은 벡터이며, 각 구간마다 형태로 총 벡터이다.  • 이 궤적은 다음 (b)와 (c) 단계에서 최적화를 통해 목표 상태에 접근시키는 데 사용된다.​  • 작성된 행렬을 바탕으로, 초기 상태 와 제어 입력 를 이용하여 전체 상태 궤적 을 계산하였다. 이 시뮬레이션 결과는 단순한 초기 조건에서의 시스템 응답을 확인하는 용도로 사용되며, 이후 (b), (c) 항목에서 제어 입력을 최적화하여 목표 상태에 도달하도록 조정될 예정이다. 최종적으로 출력된 행렬의 크기를 통해 구성의 정확성을 검증할 수 있다. |

1. **Solution (b)**
2. **문제 분석**

차량 조향 모델에서, 초기 상태 에서 시작하여 목표 상태 에 도달하도록 하는 제어 입력 시퀀스를 설계하고자 한다. 이를 위해, 아래 목적 함수를 최소화하는 최적 제어 문제를 최소제곱 문제 형태로 정식화한다:

여기서 는 제어 입력 크기를 얼마나 패널티 줄 것인지 결정하는 가중치이다.

1. **상태 전개 수식 기반 행렬 표현**

앞서 (a)에서 다음과 같은 선형 시스템 전개식을 유도하였다:

여기서:

* : 모든 시점의 상태를 수직으로 쌓은 벡터.
* : 모든 시점의 제어 입력을 수직으로 쌓은 벡터.
* , ,

1. **상태 Least-Square Problem 정식화**

목적 함수는 전체 시간 구간 동안의 상태 오차와 제어 입력 크기의 가중합으로 구성된다. 이를 행렬 형태로 다시 정리하면 다음과 같다:

여기서:

* : 목표 상태를 T회 반복하여 쌓은 벡터.

즉, 최종적으로 우리는 다음과 같은 표준 선형 최소제곱 문제를 얻게 된다:

이는 형태의 표준적인 선형 최소제곱 문제이며, 해는 다음과 같이 계산할 수 있다:

1. **Solution (c)**

아래는 문제 (c)에서 요구한 내용을 만족하는 Python 전체 코드이다. 이 코드는 다음의 동작을 수행한다:

1. (a)에서 구성한 선형 시스템 행렬 를 사용한다.
2. (b)에서 수식화한 목적 함수

를 최소화하는 least-square 문제를 풀고

1. 값들을 바꿔가며 해를 시각화한다.

|  |
| --- |
| ■ **파라미터 및 초기 상태 설정**    • T: 제어 시퀀스의 길이 (10초 동안 제어).  • dt: 시간 간격 (dt).  • l: 차량의 휠베이스 길이 (2m).  • x\_bar: 선형화 기준점인 초기 상태.  • u\_bar: 기준 제어 입력.  • x\_goal: 차량이 도달해야 할 목표 상태.  • x\_goal\_stack: 각 시간 스텝마다 동일한 목표 상태를 가진다고 가정하여 30x1 형태로 반복.  ■ **선형화 (Linearization)**    • 초기 상태 및 입력에서 파생된 파라미터들을 추출.    • 선형 시스템의 상태 행렬 A, 입력 행렬 B, 상수항 c를 구성.  • Vehicle Model 기반 동역학 행렬을 미분하여 얻은 Jacobian.  ■ **Euler-forward 이산화**    • Euler Discretization을 적용하여 연속 시스템을 이산 시스템으로 변환:  ■ **상태 전개 행렬 F, G, H 생성**    • 시간 축을 따라 상태들을 쌓은 벡터 를 만들기 위한 전개 행렬:    • 각 시간 스텝마다:  • : 입력 벡터 가 전체 상태에 미치는 누적 영향.  • : 초기 상태 가 누적 영향을 주는 방식.  • : Offset 의 누적 반영.  ■ **Least-Square 문제 풀이 ( 변화에 따라)**    • 다양한 값을 실험:  • 작은 : 빠른 수렴 속도  • 큰 : 작은 제어 입력    • 표준 선형 최소제곱 문제 형태로 푼다.  • 각 값에 따른 최적 입력 와 그에 따른 상태 궤적 를 저장한다.  ■ **시각화 (결과 그래프 출력)**    • 2개의 subplot: 위치 궤적 & 제어 입력 크기    • 평면에서의 이동 궤적 시각화.  • 목표점 표시    • 시간에 따른 제어 입력의 크기 시각화.  • 값이 커질수록 전체 입력이 줄어듦을 확인할 수 있다. |
|  |

**[결과 분석]**

1. **위치 궤적 (Position Trajectory)**

* 가 작을수록 (예: 0.01):

• 장점: 목표 위치에 더 가까이, 더 빠르게 도달한다.

• 단점: 그 과정에서 매우 큰 제어 입력을 사용한다.

* 가 클수록 (예: 100):

• 단점: 제어 입력을 작게 유지하려다 보니 도달 시간이 길어진다.

* 정도에서는:

• 장점: 제어 입력과 위치 정확도 간에 좋은 균형을 보인다.

1. 제어 입력 크기 (Control Magnitude per Step)

* 가 작을수록:

• 초반에 매우 큰 를 사용해 목표에 강하게 접근한다.

* 가 클수록:

• 전체적으로 제어 입력의 크기가 작고, 변화 폭도 적다.

• 이는 목적함수의 두 항,

사이의 trade-off가 잘 반영된 결과다.

본 실험에서는 목적함수에 포함된 제어 입력 항의 가중치 값을 변화시키며 시스템의 응답 특성을 분석하였다. 가 작을수록 시스템은 더 정확히 목표 상태에 도달하지만, 그 대가로 매우 큰 제어 입력이 요구되며 이는 실제 시스템에서 과도한 하드웨어 부담이나 안전 문제로 이어질 수 있다. 반면, 가 클수록 제어 입력은 작아지지만 목표 상태에서의 오차가 커진다. 이 결과는 정확도와 제어 노력 간의 명확한 trade-off를 잘 보여주며, 값의 선택이 제어 시스템 설계에서 중요한 조율 변수임을 나타낸다.