

Inverted Pendulum

Innova Lee(이상훈)
gcccompil3r@gmail.com

Overview

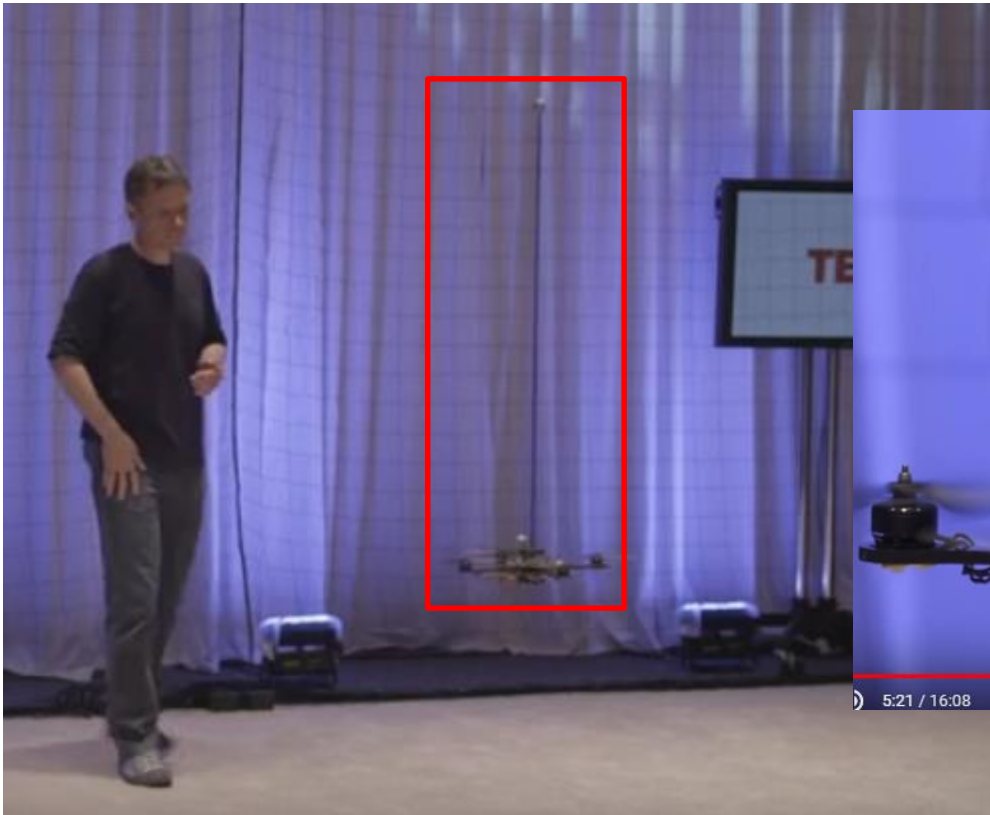
1. 역진자 제어시스템의 구성 요소 및 센서의 종류
2. 날아다니는 역진자 시스템
3. 역진자 제어시스템의 응용 분야

Introduction Inverted Pendulum

역진자 시스템에 대해 알아보도록 하겠습니다.

먼저 역진자 시스템에는 미사일, 로켓(위성, 북한이 좋아하는 ICBM), 휴머노이드, 세그웨이 같은 것들이 있습니다. 그리고 TED 에 나오는 어떤 분은 쿼드콥터 위에 역진자 시스템을 구현해서 나오기도 하였습니다. 심지어는 물병을 올려놓고도 떨어뜨리지 않고 있습니다(물리 모델링 기반이라)

<https://www.youtube.com/watch?v=w2itwFJCgFQ>



위와 같은 역진자 시스템에 대해 조금 더 생각해보겠습니다.

첫번째 사람이 나와있는 사진에는 만들고자 하는 역진자 시스템의 물리적 모델링을 수행한 것입니다.
너무 변태 같은 시스템을 사람이 모두 손으로 계산하고 수식을 세우기에는 너무 힘들기 때문에
Matlab 이나 Labview 와 같은 시스템을 기반으로 개발 속도를 끌어올립니다.

그리고 Embedded Coder 나 HDL Coder 등을 사용하여 실제 보드에 SW 를 올립니다.

이후 실제로 설계한 모델대로 잘 동작하는지 검증을 해보고 잘 되지 않았다면 다시 모델을 손봅니다.

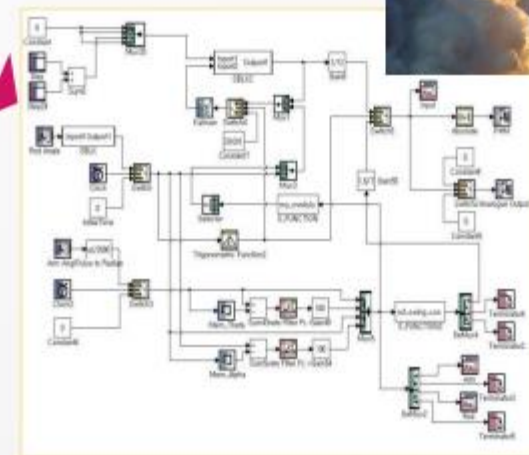
이와 같이 Matlab 및 Labview 가 제공된다 하더라도

기본적인 물리 모델링이 잘 되지 않는다면 시스템이 정상적으로 동작할 수 없다는 것을 생각해볼 수 있었습니다.

① 역진자 제어시스템의 소개

$$u = \begin{cases} \frac{u_a(|\operatorname{sgn}(E_p) \cos \alpha \dot{\alpha} - 2\lambda\dot{\theta}|) + \operatorname{sgn}(E_p) \left(\frac{F_p}{mPl}\right) \sin \alpha \cos \alpha \dot{\theta}^2 + 2\gamma\dot{\theta}\ddot{\theta}}{\operatorname{sgn}(E_p) \cos \alpha \dot{\alpha} - 2\lambda\dot{\theta}} & \text{when } |\operatorname{sgn}(E_p) \cos \alpha \dot{\alpha} - 2\lambda\dot{\theta}| > \epsilon \\ \frac{u_a(|\operatorname{sgn}(E_p) \cos \alpha \dot{\alpha} - 2\lambda\dot{\theta}|) + \operatorname{sgn}(E_p) \left(\frac{F_p}{mPl}\right) \sin \alpha \cos \alpha \dot{\theta}^2 + 2\gamma\dot{\theta}\ddot{\theta}}{\epsilon \operatorname{sgn}(\operatorname{sgn}(E_p) \cos \alpha \dot{\alpha} - 2\lambda\dot{\theta})} & \text{when } |\operatorname{sgn}(E_p) \cos \alpha \dot{\alpha} - 2\lambda\dot{\theta}| \leq \epsilon \end{cases}$$

1. Swing-up : Energy Control



Introduction Inverted Components

역진자 시스템의 구성 요소에 대해 생각해보겠습니다.

학교에서 제공하는 훌륭한 자료가 있어 이를 활용하였습니다.

기존에 만들어봤던 RC 제품군들과 유사하게 동작을 만들어주기 위해서 구동기(모터 등등)이 필요하고

어떤 동작을 취했을때 상태를 확인하기 위해 계측해주는 엔코더나 IMU 등의 다양한 센서가 필요할 것입니다.

이 정보를 기반으로 Feedback 시스템을 구성하여 동작을 취하고 오차를 보정하는 형태로 구성된다고 볼 수 있습니다.



Let's understand Flying Inverted Pendulum

TED 의 Flying 역진자에 대해 보고 약간 도전을 받아 이론적인 정리를 수행해보고자 합니다.

아래 자료를 기반으로 내용을 정리하도록 하겠습니다.

https://multivariablecontrol.files.wordpress.com/2014/12/erekson-ishmaal_783661_54867259_paper.pdf

Abstract

이 프로젝트의 목적은 Quadrotor 꼭대기에 거꾸로된 진자의 구제거 제어를 입증하는 것입니다.
이 논문에서는 진자 팁의 궤적 제어를 위해 결합 시스템의 차동 평탄도를 어떻게 활용하는지 논의합니다.
일부 작업은 거꾸로된 진자를 비행할 때 수행되었습니다 [1]
그러나 지금까지는 도립 진자와 Quadrotor 를 결합한 시스템에서는
미분 평탄도를 이용한 임의의 탄도 제어가 시도되지 않았습니다.

Introduction

역진자는 가장 중요한 고전적 제어 문제중 하나로 잘 알려져 있습니다.
잘 기술되어왔고 실현 가능한 비선형 불안정 시스템을 제공합니다.
이 문제는 수십년 동안 연구되어왔습니다 [2] [3]
Quadrotors 는 보강 학습 [4], 신경 네트워크 [5] 및 퍼지 제어 [6] 와 같은
다양한 제어 기술을 시연하는 인기있는 플랫폼으로 부상하고 있습니다.
Quadrotor 상부의 역진자의 복잡한 제어 문제는 최근에야 성취되었습니다 [1]
이것은 Infinite-Horizon Linear-Quadratic Regulator(LQR) 설계를 사용하여 수행되었습니다 [7]
이러한 System 의 조합은 상대적으로 새롭고
현재는 고정밀 실내 제어를 제공하는 시각적 피드백 시스템의 가용성에 의해 제한됩니다. [8]
이 논문에서는 차동 평탄성의 특성을 이용하는 결합 시스템에 대한 제어 전략을 제안하고 시연합니다.

이제부터 Controller 설계에 사용되는 동적 모델을 소개하고 차동 평탄 제어기 설계 및 상응하는 상태 맵핑을 제시합니다. 또한 시뮬레이션 모델과 결과를 살펴보고 결론을 도출한 후 향후 계획들에 대해 살펴봅니다.

Dynamic Models

Quadrotor 및 역진자에 대한 모든 운동 방정식은 궤도와는 독립적이므로 다양한 경우로 일반화할 수 있습니다. 이 모델의 유도에서 Quadrotor 에 대한 진자의 영향은 무시된다는 점도 중요합니다. 이 가정 및 관련 정당화는 Hehn [1] 에 의해 논의되었습니다.

Frames of Reference

이 글 [9] 에서 프레임의 Reference 가 채택되었다. 관심을 가진 좌표 Frame 은 아래와 같이 정의됩니다.

F_i = 관성 좌표계 단위 벡터 i_i 는 북쪽을 향하고, j_i 는 동쪽을 향하며, k_i 는 지구 중심을 향합니다. 이 프로젝트에 필요한 작은 작업 공간으로 인해 이 프레임의 원점을 지구 중심에서 편의를 위해 지역적으로 결정된 지점으로 변경했습니다.

F_v = Vehicle Frame 이다.
이 축은 F_i 의 축과 일직선으로 놓입니다.
그러나 원점은 Quadrotor 의 질량 중심에 있습니다.

F_{v_1} = Vehicle 1 Frame 이다.
 F_{v_1} 은 F_v 와 같은 원점을 공유합니다.
그러나 F_{v_1} 는 k_v 에 대해 Yaw 각도 ψ 만큼 회전하므로 기체가 rolling 하지 않거나 pitching 하지 않으면 i_v 는 기수를 가리킵니다.

F_{v_2} = Vehicle 2 Frame 이다.
 F_{v_2} 는 F_v 및 F_{v_1} 와 동일한 원점을 공유합니다.
이 프레임은 F_{v_1} 을 j_{v_1} 축을 중심으로 오른쪽 방향으로 pitch 각도 θ 만큼 회전하여 덩습니다.
Roll 각도가 0 인 경우 i_{v_2} 이 기체의 기수를 가리킵니다.

F_b 는 Body Frame 이다.

이것은 F_{v_2} 는 Roll 각도 ϕ 로 i_{v_2} 만큼 우회전하면 됩니다.

따라서 원점은 중력의 중심이며 i_b 는 기체의 기수를 가리키고 j_b 는 우익을 가리키며 k_b 는 기체의 하단을 가리킵니다.

Quadrotor

Quadrotor 의 상태 변수는 아래의 12 개에 해당합니다.

$p_n = F_i$ 의 i_i 기저에 대한 변위입니다.

$p_e = F_i$ 의 j_i 기저에 대한 변위입니다.

$h = F_i$ 의 $-k_i$ 기저에 대한 변위입니다.

$u = F_b$ 의 i_b 기저에 대한 속도입니다.

$v = F_b$ 의 j_b 기저에 대한 속도입니다.

$w = F_b$ 의 k_b 기저에 대한 속도입니다.

$\phi = F_{v_2}$ 에 대해 정의된 Roll 각도입니다.

$\theta = F_{v_1}$ 에 대해 정의된 Pitch 각도입니다.

$\psi = F_v$ 에 대해 정의된 Yaw 각도입니다.

$p = F_b$ 의 i_b 기저에 대한 측정된 Roll 비율입니다.

$q = F_b$ 의 j_b 기저에 대한 측정된 Roll 비율입니다.

$r = F_b$ 의 k_b 기저에 대한 측정된 Roll 비율입니다.

이 상태에서 주어진 Quadroter Dynamics 에 대한 아래의 모델의 유도는 여기서 찾을 수 있습니다. [9]

$$\begin{bmatrix} \dot{p}_n \\ \dot{p}_e \\ \dot{h} \end{bmatrix} = \begin{bmatrix} \cos(\theta)\cos(\phi) & \sin(\phi)\sin(\theta)\cos(\phi) - \cos(\phi)\sin(\psi) & \cos(\phi)\sin(\theta)\cos(\psi) + \sin(\phi)\sin(\psi) \\ \cos(\theta)\sin(\phi) & \sin(\phi)\sin(\theta)\sin(\phi) - \cos(\phi)\cos(\psi) & \cos(\phi)\sin(\theta)\sin(\psi) + \sin(\phi)\sin(\psi) \\ \sin(\theta) & -\sin(\phi)\cos(\theta) & -\cos(\phi)\cos(\theta) \end{bmatrix}$$

$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} rv - qw \\ pw - ru \\ qu - pv \end{bmatrix} + \begin{bmatrix} -g\sin(\theta) \\ g\cos(\theta)\sin(\phi) \\ g\cos(\theta)\cos(\phi) \end{bmatrix} + \frac{1}{m} \begin{bmatrix} 0 \\ 0 \\ -F \end{bmatrix}$$

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin(\phi)\tan(\theta) & \cos(\phi)\tan(\theta) \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \frac{\sin(\phi)}{\cos(\theta)} & \frac{\cos(\phi)}{\cos(\theta)} \end{bmatrix}$$

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} \frac{J_y - J_z}{J_x} qr \\ \frac{J_z - J_x}{J_y} qr \\ \frac{J_x - J_y}{J_z} qr \end{bmatrix}$$

J_x, J_y 및 J_z 는 Quadroter 의 관성 상수, g 는 중력 가속도 (9.80665 m/s^2), F 는 roter 에 의한 추력입니다.

Inverted Pendulum

진자의 상태 변수는 아래의 4 가지 양에 해당합니다.

$x_v = F_v$ 의 i_v 기저에 대한 변위입니다.

$y_v = F_v$ 의 j_v 기저에 대한 변위입니다.

$\dot{x}_v = F_v$ 의 i_v 기저에 대한 속도입니다.

$\dot{y}_v = F_v$ 의 j_v 기저에 대한 속도입니다.

진자 상태 방정식에 대한 다른 관심 변수는 아래와 같습니다.

L = 진자의 길이입니다.

$\xi = F_v$ 의 $-k_v$ 기저에 대한 변위이며 이 값은 $\sqrt{L^2 - x_v^2 - y_v^2}$ 입니다.

$\ddot{x} = F_v$ 의 i_v 기저에 대한 Quadroter 의 가속도입니다.

$\ddot{y} = F_v$ 의 j_v 기저에 대한 Quadroter 의 가속도입니다.

$\ddot{z} = F_v$ 의 k_v 기저에 대한 Quadroter 의 가속도입니다.

$$\alpha = \frac{-\xi^2}{(\xi^2 + x_v^2)} \left(\ddot{x} + \frac{\dot{x}_v^2 x_v + \dot{y}_v^2 x_v}{\xi^2} + \frac{\dot{x}_v^2 x_v^3 + 2\dot{x}_v \dot{y}_v x_v^2 y_v + \dot{y}_v^2 y_v^2 x_v}{\xi^4} - \frac{x_v(\ddot{z} + g)}{\xi} \right)$$

$$\beta = \frac{-\xi^2}{(\xi^2 + y_v^2)} \left(\ddot{y} + \frac{\dot{y}_v^2 y_v + \dot{x}_v^2 y_v}{\xi^2} + \frac{\dot{y}_v^2 y_v^3 + 2\dot{y}_v \dot{x}_v y_v^2 x_v + \dot{x}_v^2 x_v^2 y_v}{\xi^4} - \frac{x_v(\ddot{z} + g)}{\xi} \right)$$

진자의 운동 방정식은 아래와 같습니다.

$$\begin{bmatrix} \ddot{x}_v \\ \ddot{y}_v \end{bmatrix} = \begin{bmatrix} \left(\alpha - \frac{\beta x_v y_v}{(L^2 - y_v^2) \xi^2} \right) \left(1 - \frac{x_v^2 y_v^2}{(L^2 - y_v^2)^2 \xi^4} \right) \\ \beta - \frac{\ddot{x}_v x_v y_v}{L^2 - x_v^2} \end{bmatrix}$$

또한 제어 설계는 궤도 제어를 위해 진자팁의 상태를 측정해야 합니다.
이들은 이전에 정의된 상태의 함수이며 아래와 같습니다.

$x_i = p_n + x_v$ 로 F_i 의 i_i 기저에 대한 변위입니다.

$y_i = p_e + y_v$ 로 F_i 의 j_i 기저에 대한 변위입니다.

$z_i = h - \xi$ 로 F_i 의 k_i 기저에 대한 변위입니다.

$\dot{x}_i = \dot{p}_n + \dot{x}_v$ 로 F_i 의 i_i 기저에 대한 속도입니다.

$\dot{y}_i = \dot{p}_e + \dot{y}_v$ 로 F_i 의 j_i 기저에 대한 속도입니다.

$\dot{z}_i = \dot{h} - \dot{\xi}$ 로 F_i 의 k_i 기저에 대한 속도입니다.

Controller Design

2 층 계층 Controller 는 System 의 차동 평탄성을 이용하여
전체 상태 Feedback 및 Feedforward 방법을 사용하여 임의의 궤적에 맞게 설계되었습니다.

Differential Flatness Control

차동적으로 평평한 시스템은 궤도 추적에서 많은 유용한 특성을 가지고 있습니다.

이러한 유용한 특성중 하나는 상태 및 입력 변수가 미분 방정식을 통합하지 않고

Flat 출력 및 그 파생값을 유한한수로 직접 표현할 수 있다는 것입니다.

이 일반적인 개념은 D. Hilbert 와 E. Cartan [11] 에 의해

미분 방정식의 수식이 미지의 수보다 엄격히 적으면서

미분 방정식 시스템에 대한 연구로 거슬러 올라갈 수 있습니다.

Pendulum Tip Trajectory Controller

계층적 Controller 의 첫 번째 계층은 간단한 점 질량(우리의 진자 팁)의 궤적을 제어하려고 합니다. Feedback 을 위해 Infinite-Horizon Linear-Quadratic Regulator(LQR) Design 을 사용하고 주어진 궤적으로부터 Feedforward 를 제공합니다.

주어진 궤도에 적합한 상태는 아래와 같습니다.

$$\bar{x}_r = [x_{ir} \quad y_{ir} \quad z_{ir} \quad \dot{x}_{ir} \quad \dot{y}_{ir} \quad \dot{z}_{ir}]^T$$

진자 팁의 실제 상태는 아래와 같습니다.

$$\bar{x} = [x_i \quad y_i \quad z_i \quad \dot{x}_i \quad \dot{y}_i \quad \dot{z}_i]^T$$

LQR Gain Matrix 를 계산하기 위한 상태 공간은 아래와 같습니다.

$$\dot{\bar{x}} = A\bar{x} + B\bar{u}$$

$$A = \begin{bmatrix} \mathbf{0}_{3 \times 3} & I_{3 \times 3} \\ \dots & \mathbf{0}_{3 \times 6} \end{bmatrix}$$

$$B = \begin{bmatrix} \mathbf{0}_{3 \times 3} \\ I_{3 \times 3} \end{bmatrix}$$

LQR Gain Matrix 인 K 가 주어지면 원하는 가속도를 고려한다.

$$\bar{u} = -K(\bar{x}_r - \bar{x})$$

주어진 궤적에 대한 입력은 아래와 같습니다.

$$\begin{aligned} \bar{u}_r &= [\ddot{x}_{ir} \quad \ddot{y}_{ir} \quad \ddot{z}_{ir}]^T \\ u_{pm} &= \bar{u} + \bar{u}_r \end{aligned}$$

점 질량에 대한 총 입력은 아래와 같습니다.

$$u_d = u_{pm} + [0 \quad 0 \quad g]^T$$

Mapping to Quadrotor/Pendulum System

진자 틱에서 원하는 가속도 u_{pm} 을 실현하려면 입력 변수에서 System 상태 변수로의 맵핑이 필요합니다. 이것은 Quadrotor 가 진자 샤프트의 방향으로 진자 틱에만 작용할 수 있다는 가정하에 제공됩니다.

따라서 필요한 System 상태는 아래와 같습니다.

$$x_{vr} = L \frac{u_d(1)}{\|u_d\|}$$

$$y_{vr} = L \frac{u_d(2)}{\|u_d\|}$$

$$p_{nr} = x_i - x_{vr}$$

$$p_{er} = y_i - y_{vr}$$

$$h_r = z_i - \sqrt{L^2 - x_{vr}^2 - y_{vr}^2}$$

북쪽 방향을 유지할 수 있도록 합니다.

$$\psi_r = 0$$

적절한 맵핑이 진자 틱에 적용되도록
요구되는 jerk 를 추정해야 하기에 아래를 0 으로 설정합니다.

$$\dot{x}_{vr} = \dot{y}_{vr} = u_r = v_r = w_r = 0$$

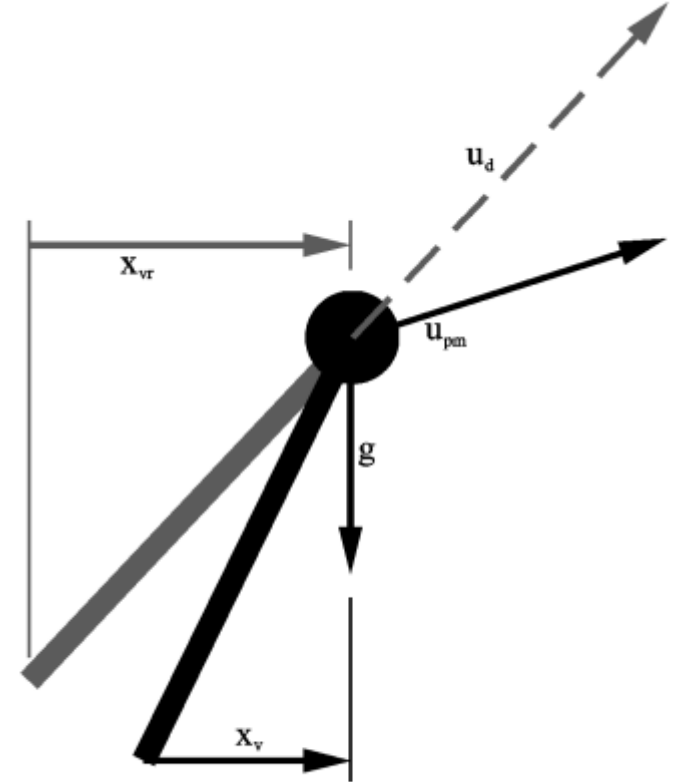


Figure 1. Acceleration Mapping

Quadrotor/Pendulum System Controller

계층적 제어기의 두 번째 계층은 역진자와 Quadrotor의 결합된 시스템의 상태를 제어하려고 합니다. 첫 번째 계층에서 설명한 것과 동일한 방법을 사용합니다. 그러나 reference 가속은 없습니다.

이전 맵핑이 주어진 적절한 상태는 아래와 같습니다.

$$x_r = [x_{vr} \ y_{vr} \ \dot{x}_{vr} \ \dot{y}_{vr} \ p_{nr} \ p_{er} \ h_r \ u_r \ v_r \ w_r \ \psi_r]^T$$

시스템의 실제 상태는 아래와 같습니다.

$$x_r = [x_v \ y_v \ \dot{x}_v \ \dot{y}_{vr} \ p_n \ p_e \ h \ u \ v \ w \ \psi]^T$$

LQR Gain Matrix를 계산하기 위한 상태 공간을 아래와 같습니다.

$$\dot{\hat{x}} = A\hat{x} + Bu$$

$$A = \begin{bmatrix} \mathbf{0}_{2 \times 2} & I_{2 \times 2} & \mathbf{0}_{2 \times 7} \\ \dots & \mathbf{0}_{2 \times 11} & \dots \\ \mathbf{0}_{3 \times 7} & I_{3 \times 3} & \mathbf{0}_{3 \times 1} \\ \dots & \mathbf{0}_{4 \times 11} & \dots \end{bmatrix}$$

$$B = \begin{bmatrix} \dots & \mathbf{0}_{2 \times 6} \\ I_{2 \times 2} & \mathbf{0}_{2 \times 4} \\ \dots & \mathbf{0}_{3 \times 6} \\ \mathbf{0}_{4 \times 2} & I_{4 \times 4} \end{bmatrix}$$

LQR Gain Matrix인 K가 주어지면 원하는 가속도를 고려한다.

$$u = -K(x_r - x)$$

Mapping to Quadrotor Inputs

진자 상태가 작은 값이라고 가정하면 균형 잡힌 진자에 대해 System 이 선형화됩니다.
결과는 아래와 같습니다.

$$\begin{bmatrix} \ddot{x}_v \\ \ddot{y}_v \end{bmatrix} = \begin{bmatrix} \left(\alpha - \frac{\beta x_v y_v}{(L^2 - y_v^2) \xi^2} \right) \left(1 - \frac{x_v^2 y_v^2}{(L^2 - y_v^2)^2 \xi^4} \right) \\ \beta - \frac{\ddot{x}_v x_v y_v}{L^2 - x_v^2} \end{bmatrix}$$

그러므로 아래와 같다.

$$\begin{bmatrix} \ddot{x}_v \\ \ddot{y}_v \end{bmatrix} \approx \begin{bmatrix} -\ddot{x} \\ -\ddot{y} \end{bmatrix}$$

따라서 u 는 아래와 같이 나타낼 수 있습니다.

$$u_{p1} = u_3 - u_1$$

$$u_{p2} = u_4 - u_2$$

$$u_{p3} = u_5$$

$$u_{p4} = u_6$$

Quadrotor 의 일반 모델의 경우 입력 벡터를 아래와 같이 정의합니다.

$$v = \begin{bmatrix} T^d \\ \Phi^d \\ \theta^d \\ r^d \end{bmatrix}$$

On-Board 자세 제어기는 원하는 입력 명령에 대해 Loop 를 닫음으로써 System 을 더 이상 제어할 필요가 없습니다.
 이 문서 [10] 으로부터 u 에서 v 로의 맵핑이 채택되었습니다.
 m_q = Quadrotor의 질량

$$T = m_q \left\| \begin{bmatrix} u_{p1} \\ u_{p2} \\ u_{p3} \end{bmatrix} \right\|$$

$$z = -R(\psi) \begin{bmatrix} u_{p1} \\ u_{p2} \\ u_{p3} \end{bmatrix} \frac{m_q}{T}$$

$$\Phi = \sin^{-1}(-z_2)$$

$$\theta = \tan^{-1} \left(\frac{z_1}{z_3} \right)$$

$$r_d = u_{p4}$$

$$v = \begin{bmatrix} T \\ \Phi \\ \theta \\ r_d \end{bmatrix}$$

이 접근법의 한계와 이점은 [10] 에 설명되어 있습니다.

AR 에 동일한 모델을 적용하는데 필요한 입력 사항입니다.
 드론은 약간의 변경이 필요합니다.
 여기에는 축척 계수, 부호 변화 및 고도 비율에 대한 항이 포함됩니다.

$$v_{AR} = \begin{bmatrix} \dot{z} \\ \Phi \\ \theta \\ r_d \end{bmatrix}$$

\dot{z} 는 아래와 같이 구할 수 있습니다.
 δ = 예상 지연, Δt = 동작 주파수의 주기, $C = (\delta + \Delta t)$ 입니다.
 이 상수는 상태 추정치의 노이즈에 따라 크기를 조정해야 합니다.

$$\dot{z} = \dot{h}_r + \ddot{z}_r C$$

그러므로 아래와 같이 다시 적을 수 있습니다.

$$v_{AR} = \begin{bmatrix} -\frac{\dot{z}}{\text{max altitude rate}} \\ \Phi \\ -\frac{\text{max euler angle}}{\theta} \\ -\frac{\text{max euler angle}}{r_d} \\ \text{max yaw rate} \end{bmatrix}$$

부호 변경 및 축척 계수에 유의하도록 합니다.
 이것은 AR Drone 에 대한 입력의 적절한 선상에서 해석이 필요합니다.

Inverted Pendulum Applications

역진자 시스템이 사용된 응용단에 대해 알아보겠습니다..

세그웨이: 도보 이동시 걷는것보다 빠르다는 장점이 존재합니다만 군사 관계자들에 입장에서선 ... 그다지 좋지 못했습니다.

NineBot



SegWay



단순히 상상력을 발휘해보는 것으로 서빙 드론입니다.

저기에 떠오르는 것이 물컵이나 와인등이 될 수도 있고 음식등을 서빙할 수도 있을 것입니다.
(물론 웨이터나 웨이트레스는 엄청나게 반대하겠지만요)



이 두 가지 사실을
종합해 보면

5:21 / 16:08

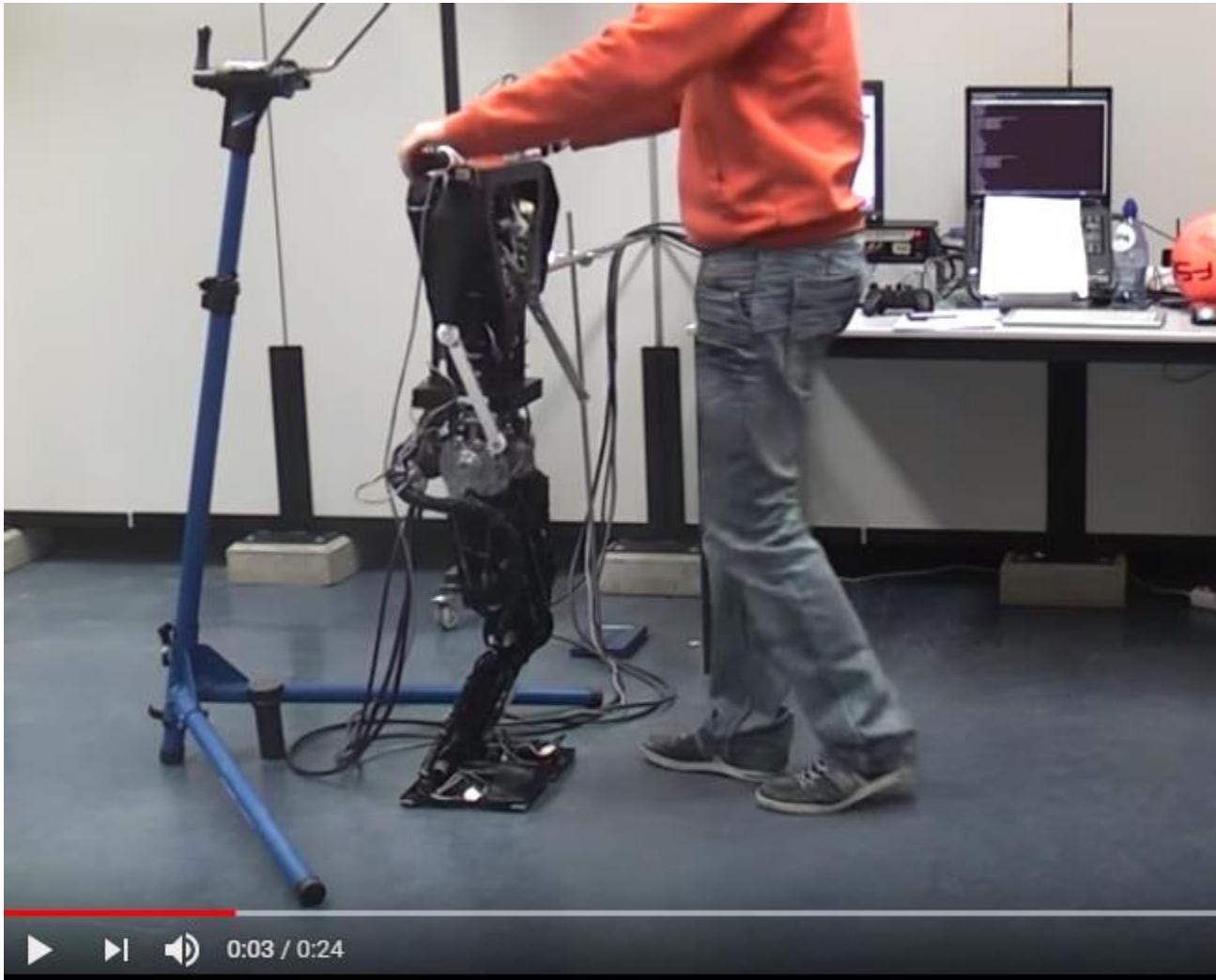
아래 영상을 보며 생각해본 것으로 자율 주행 오토바이도 가능할것 같습니다.

오토바이를 자율적으로 주행시키기 위해서도 도립 진자 제어 시스템이 필요할 것입니다.



또한 이족 보행을 구현하기 위해서도 도립 진자 제어 시스템이 필요합니다.

최근 Boston Dynamics 가 덤블링까지 수행하는 휴머노이드를 개발하였습니다.



3D Linear Inverted Pendulum Walk on Tulip

References

1. Inverted Pendulum
<http://www.engr.usask.ca/classes/EE/480/Inverted%20Pendulum.pdf>
2. Flying Inverted Pendulum On AR.Drone Using Differential Flatness Control and RISC Multi-Agent Analysis Platform
https://multivariablecontrol.files.wordpress.com/2014/12/erekson-ishmaal_783661_54867259_paper.pdf
3. 3D Linear Inverted Pendulum Walk on Tulip
https://www.youtube.com/watch?v=jYb9TJ_4J0c
4. What's new, Atlas ?
<https://www.youtube.com/watch?v=fRj34o4hN4I>
5. The astounding athletic power of quadcopters
<https://www.youtube.com/watch?v=w2itwFJCgFQ&t=321s>
6. Lit Motors Gyro Techno
<https://www.youtube.com/watch?v=68ixyfa0l84>