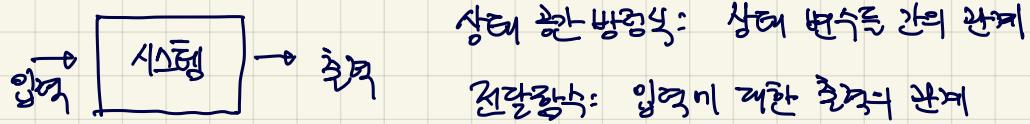


의 을 타 수 측 금 토						
			6	7		
✓ 번역봉사	8	9	10	11	12	13
✓ 기공실 보고서	15	16	17	18	19	20
- QP						21
✓ 지능제어 강의	22	23	24	25	26	27
✓ 영역학 강의						28

권보록 : 자동주행 차량 제어 트랙 인천대 강한국 교수님.

1. 시스템의 상호작용 → 수식 미분방정식 수학적 모델 : 상태공간 모델, 전달함수



ex)

조향 시스템 : 헤드-바퀴 헤드와 바퀴의 관계는 물리법칙으로 구함

입력, 출력이 다르면 수학적 모델끼리 달라짐.

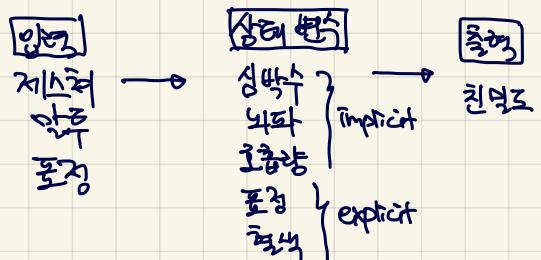
행렬 각도  $\rightarrow$  조향시스템  $\rightarrow$  바퀴 각도

행렬의 속도  $\rightarrow$

+이분모델 추가

제어 잘 하면 인간관계도 잘함?  $\Rightarrow$

사람을 시스템으로 모델링하기



모델링: 상관관계를 수식으로 표현

# 01. 전달함수

↳ 시간영역에서 정의된 입력과 출력을 라플라스 변환하여 주파수 영역에서 나타낸  
입력과 출력의 비를  $G(s)$ 로 표시하여 전달하는 함수.



라플라스 변환 : 미분 방정식을 쉽게 푸는 방법.

시스템  $\xrightarrow{\text{이분방정식}}$  수학적 모델.

예. 직선, 회전 운동의 라플라스 변환

$$f = m\ddot{x} + f_v \dot{x} + kx = m\ddot{x} + f_v \dot{x} + kx$$

$$T = J\ddot{\theta} + D\dot{\theta} + K\dot{\theta}$$

$$\begin{aligned} & \xrightarrow{\text{라플라스변환}} F(s) = (ms^2 + f_v s + k) X(s) \\ & \xleftarrow{\text{라플라스역변환}} T(s) = (J s^2 + D s + K) \Theta(s) \end{aligned}$$

라플라스 변환 공식

$$\mathcal{L}(f(t)) = \int_0^\infty e^{-st} f(t) dt$$

$$\mathcal{L}(y(t)) = Y(s) = \int_0^\infty e^{-st} y(t) dt$$

$$\mathcal{L}(y'(t)) = \int_0^\infty e^{-st} y'(t) dt = [y(t)e^{-st}]_0^\infty + s \int_0^\infty e^{-st} y(t) dt = -y(0) + s Y(s)$$

$$\mathcal{L}(y''(t)) = \int_0^\infty e^{-st} y''(t) dt = s^2 Y(s) - s y(0) - y'(0)$$

$$\text{초기값 } y(0) = y'(0) = 0 \quad \mathcal{L}(y') = s Y(s) \quad \mathcal{L}(y'') = s^2 Y(s)$$

$s$ : 이분연산자.  $\frac{1}{s}$ : 축분연산자.

$$f = m\ddot{x} + b\dot{x} + kx$$

$$F(s) = (ms^2 + bs + k) X(s)$$

$$G(s) = \frac{X(s)}{F(s)} = \frac{1}{ms^2 + bs + k}$$

입력에 대한 출력,

자율주행 차의 조향시스템, 액슬 브레이크, 서스펜션을 전달함수로 표현

## 2. 상태공간 설계

시스템은 전달함수 or 상태공간으로 표현

상태변수: 시스템 특성을 나타내는 변수.

변위 속도, 가속도를 기계시스템의 상태변수로 선정.

상태방정식: 차수가 높은 이분방정식을 여러 개의 1차 이분방정식으로 나타냄.  
*1st ODE*

출력방정식: 출력과 입력과 상태변수와 어떤 관계인지를 알려주는 방정식

상태변수 시스템  $\leftarrow$  상태공간 모델

↑  
시스템의 특성을 나타내는 변수

$$f(t) = m\ddot{x} + b\dot{x} + kx \quad \ddot{x}, \dot{x}, x : \text{상태변수.}$$

상태공간 방정식 구성.

$$\text{상태방정식} : \dot{x} = Ax + Bu \quad x : \text{변수}$$

$$\text{출력방정식} : y = Cx + Du \quad A, B, C, D \in \text{행렬}$$

예)  $f = m\ddot{x} + b\dot{x} + kx \rightarrow \dot{x}_2 = \dot{\dot{x}}_1$

$$\begin{aligned} x_1 &= x(t) \\ \dot{x}_1 &= \dot{x}_2 = \dot{x}(t) \end{aligned}$$

$$f = m\dot{x}_2 + b\dot{x}_2 + kx_1 \rightarrow \dot{x}_2 = \underbrace{-\frac{b}{m}\dot{x}_2 - \frac{k}{m}x_1 + \frac{f}{m}}_{\text{②}} \quad \text{②}$$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{b}{m} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{f}{m} \end{bmatrix} f(t) \Rightarrow \dot{x} = Ax + Bu$$

입력 출력  
시스템 행렬

출력: 선서 출력 가능한 변수

예  $y = \underbrace{Cx_1}_{\text{state matrix}} + \underbrace{Du}_{\text{feedforward word 행렬}}$

### 03 제어 안정성 심화

발산  $\rightarrow$  안정화 수렴  $\rightarrow$  특성한 티성 갖도록

pole :  $G(s)$ 의 분모를 0으로 만드는  $s$

zero :  $G(s)$ 의 분자 0으로 만드는  $s$ .

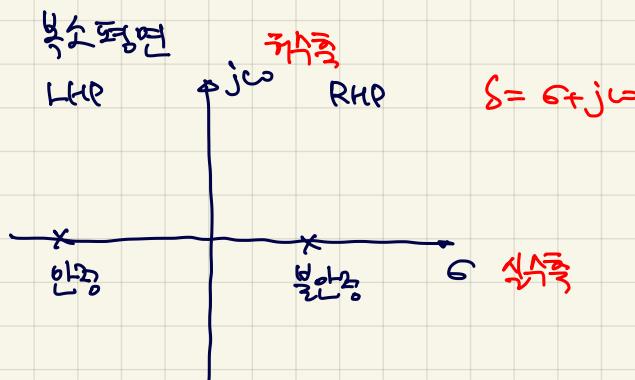
$$G(s) = \frac{(s-5)}{(s+1)(s-3)}$$

$$P = -1, 3$$

$$Z = 5$$

극점이 복소수인 경우  $G(s) = \frac{1}{s^2 + 2s + 2} = \frac{1}{(s+1)^2 + 1} = \frac{1}{(s+1+i)(s+1-i)}$   
 $s = -1 \pm i = -1 - i, -1 + i$

극점의 위치  $\rightarrow$  안정성.



왜  $G(s)$ 의 극점이 LHP에 있을 때 불안정?

$$G(s) = \frac{s-5}{(s+3)(s-1)} = \frac{2}{s+3} + \frac{-1}{s-1} \quad \mathcal{L}^{-1}(G(s)) = 2e^{-3t} - e^t$$

(단지 예시)

$\Rightarrow$  모든 극점 pole 이 LHP에 존재해야 함.

상태 공간 모델에서의 안정성.

$$\boxed{\text{전략함수의 극점}} = \boxed{\text{상태 공간 모델의 모유값}} \Rightarrow \text{LHP에 존재하면 안정성}$$

$$\dot{x} = Ax + Bu \quad A: \text{시스템 행렬}$$

$$A\vec{v} = \lambda\vec{v} \quad (A - \lambda I)\vec{v} = 0 \quad |A - \lambda I| = 0 : \text{특성화 방정식.}$$

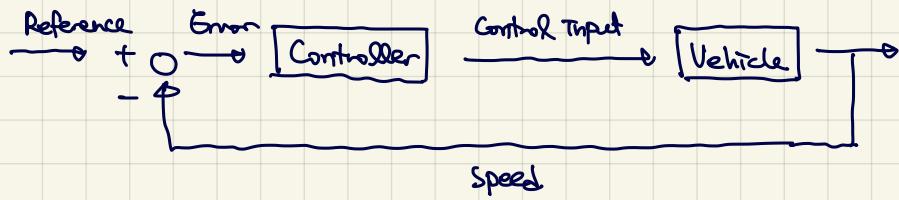
$$\dot{x} = \underbrace{\begin{bmatrix} 0 & 1 \\ -2 & -2 \end{bmatrix}}_A x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u$$

$$|A - \lambda I| = \begin{vmatrix} -\lambda & 1 \\ -2 & -2-\lambda \end{vmatrix} = \lambda(2+\lambda) + 2 = 0 \quad \lambda \neq -2, -1 = 0$$

$$\lambda = -1 + i \quad \lambda = -1 - i,$$

## 04 순항 제어기 Cruise control

↳ 지정한 속도 유지, 종방향 ADAS 시스템

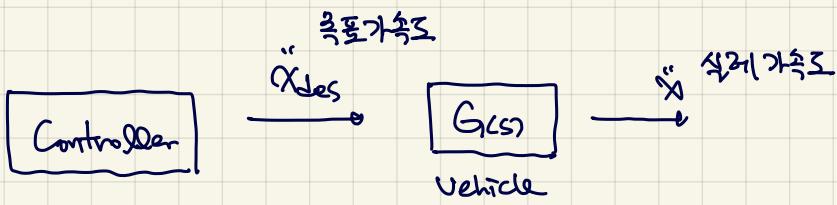
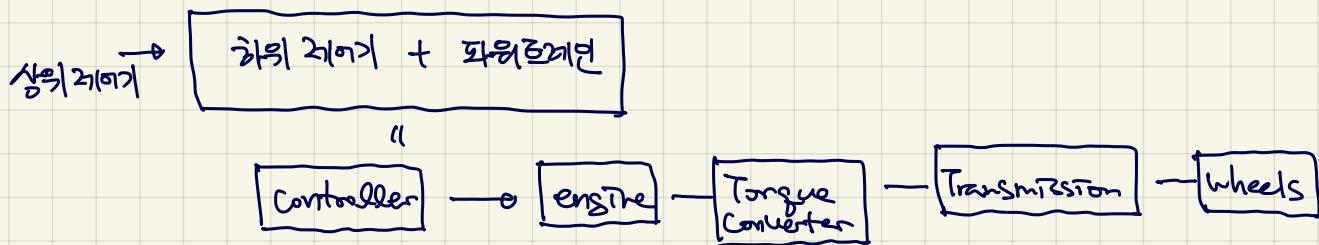


비<math>\rightarrow</math> 회전수, or TCU Sensor → 속도

Control input : 제어 입력. 차량을 움직이기 위하여 가속도가 차량에 인가됨.

Controller : 오차 없도록

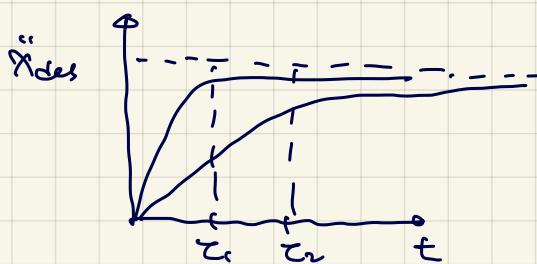
차량의 종방향 모티브



$$G(s) = \frac{\ddot{x}}{\ddot{x}_{des}} = \frac{1}{\tau_{set}}$$

$$\ddot{x} = \frac{1}{\tau_{set}} \ddot{x}_{des}$$

일차 전달함수의 시간 상수.



즉 이 그러면 응답 느림. 그 목표의 63% 도달 전에는 시간

정상상태 오차 없애기 ← [적분 제어기]

$$\ddot{x}_{des}(t) = -k_p(V_r - V_{ref}) - k_i \int_0^t (V_r - V_{ref}) dt$$

proportional gain      integral gain

증상 피드백

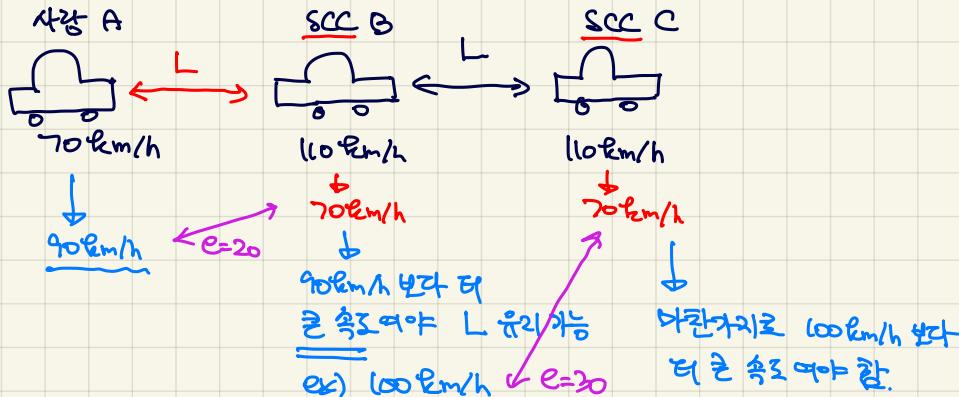
## 05 String Stability ACC, 정체 상황

유령정체 : 차로 변형. 앞 차량의 금지거리로 인한 오차들이 계속 누적되어 뒤 차들이 정체됨.

String Stability : 전체 대역에서 앞 차량의 오차가 뒤 차량 오차로 전파X

### 주행 시나리오

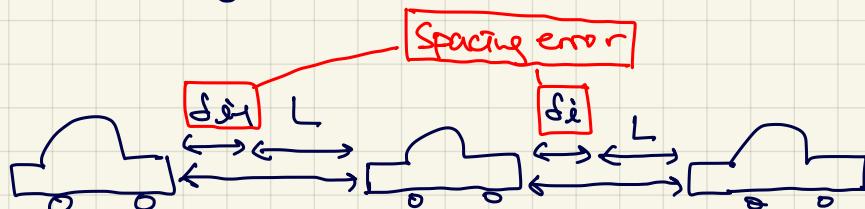
1. 두 차량 간격 L로 유지



Error propagation 오차전파. : 뒤로 갈수록 오차 증가

유령정체는 속도 증가 뿐 아니라 정지 시에도 발생.

String stability : 전체 대역에서 앞 차량의 오차가 뒤 차량 오차로 전파



전달함수를 통한 String Stability 해석.

$$G(s) = \frac{s_i}{s_{i-1}} (s) \quad \begin{cases} < 1 & \Rightarrow \text{오차 뒤로 전파 X} \\ > 1 & \Rightarrow \text{오차 뒤로 전파} \end{cases}$$

String stability 만족한다.  $\Leftrightarrow \|G(s)\|_\infty < 1$

사람이 운전할 경우 개인마다 안전거리 응답 속도가 다름  $\Rightarrow$  String stability 만족 어려움.

오차 전파  $\rightarrow$  사고 발생 시킬 가능성 있음  $\rightarrow$  고정된 목표 상대거리가 아닌  
error propagation  $\rightarrow$  유동적인 상대거리 설정 필요

String stability 만족을 위해 적절한 안전 거리 설정 하는 제어가 필요,

## 06 Constant Spacing Control

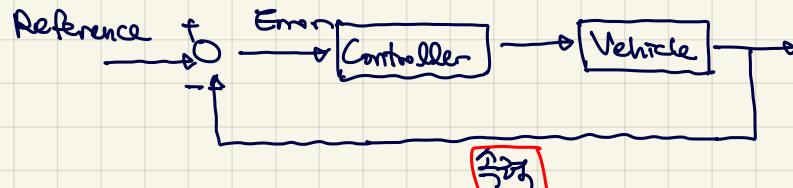
순항 제어 : 내 차량이 고정 속도를 유지하도록 하는 제어.

적응형 순항 제어 : 고정 속도 운행 중 앞 차량이 있다면 안전거리 유지하여 주행.

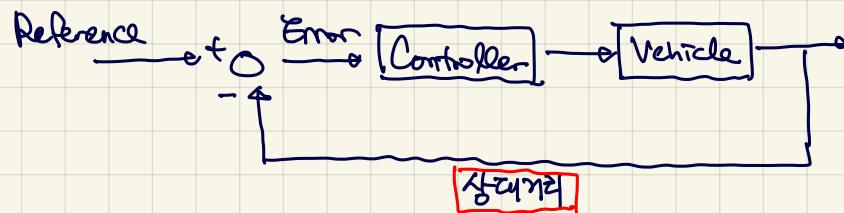
PD 제어기 : 비례+미분 제어기, 차도구간에서 응답 속도 고려 제어

Constant Spacing control : 앞차량과 간격을 상수로 설정.

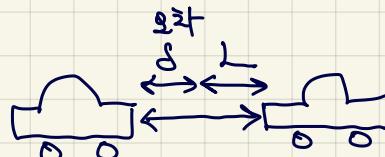
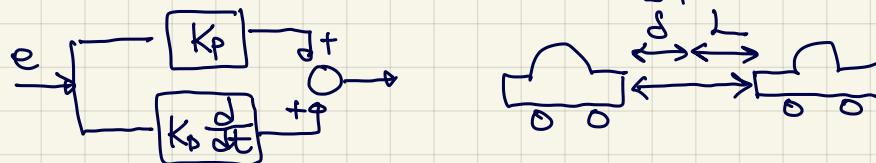
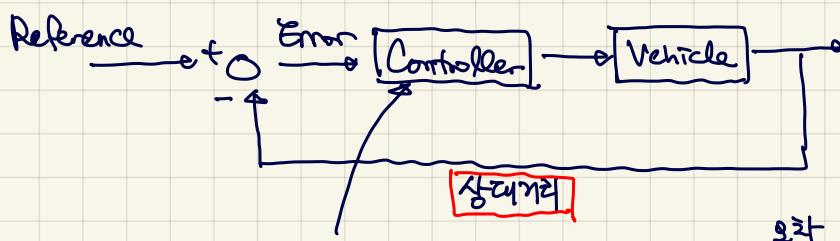
### Cruise Control



### Smart Cruise Control

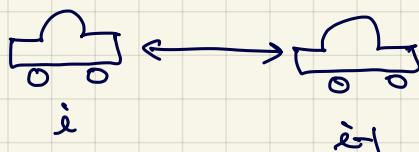


SCC는 앞차량이 움직일 때 반응 빠르기  $\Rightarrow$  PD 제어기



String stability 만족  $\rightarrow$  error propagation 막기

$$G(s) = \frac{f_2}{f_{21}}(s) \quad \| G(s) \|_\infty \leq 1$$



모든 차량이 동일 제어기면

$$G(s) = \frac{S_i(s)}{S_i(s)} = \frac{k_d s + k_p}{s^2 + k_d s + k_p} \rightarrow |G(s)| (s \leq 0) \text{ 단위 } \underline{\text{Bode plot}}$$

$20 \log |G| > 0$  String stability 불안정  
 $\leq 0$  String stability 안정

$k_p, k_d$  가 어떤 상수이려도  $M = 20 \log |G| > 0$  인  $s$  존재.

[ PD 제어기는 여러 차량 관제 String stability 불안정 ]

[ 고정 간격 보다 유동적 간격 ]

## 07 Constant Time Gap Control

SCC에서 앞차와 안전거리 유지 방법.

Constant spacing control은 고정 간격 사용,  $\rightarrow$  String stability 불안정

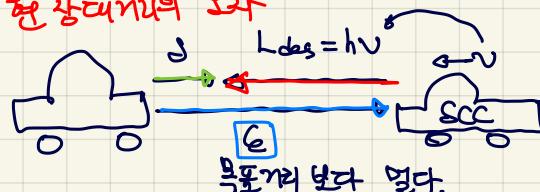
Constant time gap control  $L = h \cdot v$   $\frac{h=2}{\uparrow}$   $v=20 \rightarrow L=40 \text{ m}$   
time gap parameter.

제어기 구성요소

$\delta$ : 목표 상대거리 오차 목표상대거리와 현상대거리의 오차

$L_{des}$ : 목표 상대거리

$\varepsilon$ : 허용 상대거리



$$\delta = \varepsilon + L_{des} = \varepsilon + h \cdot v$$

String stability 만족 제어기 설계

" $\dot{x}_{des}$ : 차량이 가려야 할 목표 가속도 값 SCC  $\rightarrow$  파워트레인 입력값 없  
추가 변수.

$$\ddot{x}_{des} = -\frac{1}{h} (\dot{\varepsilon} + \lambda \delta) : CTG 제어기$$

$\uparrow$  상대거리  
속도비례증수 변화량

목표 상대거리 오차

차량 2대

$$\ddot{x}_{des} = \ddot{x} \text{ 로 가정} \quad \dot{s} = \dot{e} + h\nu \quad \xrightarrow{\frac{d}{dt}} \quad \begin{aligned} \dot{j} &= \dot{e} + h\ddot{x} \\ \dot{e} &= \dot{j} - h\ddot{x} \end{aligned}$$

$$\underline{\ddot{x}_{des} = -\frac{1}{h}(\dot{e} + \lambda s)}$$

연습.

$$\begin{aligned} \dot{e} &= \dot{j} - h \cdot \left( -\frac{1}{h}(\dot{e} + \lambda s) \right) \\ &= \dot{j} - \dot{e} - \lambda s \\ \Rightarrow \underline{\dot{j} = -\lambda s} \quad \Rightarrow \quad &\text{"}\dot{s}\text{" 수렴} \end{aligned}$$

파워트레인 모델: 1차 모델로 간소화

$$\ddot{x} = \frac{1}{z_s+1} \ddot{x}_{des}$$

$$\ddot{x}_{des} = -\frac{1}{h}(\dot{e} + \lambda s) \text{ 대입} \quad \ddot{x} = \frac{1}{z_s+1} \left( -\frac{1}{h}(\dot{e} + \lambda s) \right)$$

$$\underline{z \ddot{x}_i + \ddot{x}_i = -\frac{1}{h}(\dot{e}_i + \lambda s_i)} \quad \textcircled{1}$$

i 번째 차량의 목표 상태에 오차를 두번 미身躯

$$\underline{\dot{s} = \dot{e} + h\nu \quad \dot{s}_i = \dot{e}_i + h\ddot{x}_i} \quad \textcircled{2}$$

$$\begin{aligned} \textcircled{1}, \textcircled{2} \text{ 연습. } \dot{e}_i &= \dot{s}_i + \frac{1}{z} (\dot{e}_i + h\ddot{x}_i + \lambda s_i) \\ &= \frac{1}{z} (\dot{s}_i + \lambda s_i) \end{aligned}$$

$$G(s) = \frac{\dot{s}_i}{\dot{s}} = \frac{s + \lambda}{h^2 s^3 + h s^2 + ((+\lambda)h) s + \lambda}$$

$$\|G(s)\| \leq 1 \quad \forall s \geq 0 \quad \text{인지 확인}$$

제한한  $\lambda, h$  설정  $\rightarrow \|G(s)\| \leq 1$  만족.  $\underline{h \geq 2z}$  스텝의 특성

ex)  $z=0.5 \rightarrow \lambda=0.4 \quad z=0.5 \text{ and } h=1.8 \text{ sec}$

$h$ : 사전이 몇 가지 투입값을 선정  $\rightarrow$  운전자가 버튼으로 선택

운전 스타일에 따라 앞 차량과 거리 조절 가능

ex	$h$	Safe	Normal	Aggressive
		0.4 여기	1.8	1.2 가장자

## 08 협동 적응형 순항제어.

Cooperative Cruise Control CSCC  
Smart

협동 적응형 순항제어: 앞 차량과 간격을 조절 더 가깝게 유지하여 주행 가능.  
군집 차량에 적용 가능

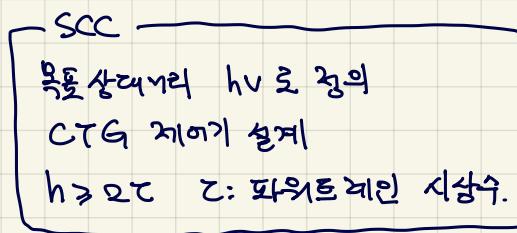
Ex) 여러 대의 트럭이 대역을 이루며 주행

SCC 보다 CSCC는 조금 더 가까이 붙어서 주행.

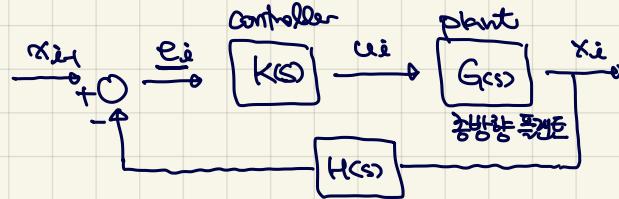
장점 1. 고속도로 효율적 사용.

장점 2. 뒤 차량 연비 향상. 공기저항으로 인한 에너지 소모  
운전자 피로도 사고도

CSCC 원리.



$$\text{목표 상대거리} = hV + \text{CTG 대신 [PD 제어기] 사용.}$$



$$\ddot{x} = \frac{1}{\tau s + 1} \ddot{x}_{des} \Rightarrow \text{출현을 위치로 } x_i = \frac{1}{\tau s + 1} u_i$$

$$G(s) = \frac{x_i}{u_i} = \frac{1}{s^2(\tau s + 1)}$$

$$\text{PD 제어기: } K_p e + K_o \frac{de}{dt} = u_i \quad (K_p + K_o s) E(s) = U(s)$$

$$K(s) = \frac{U(s)}{E(s)} = k_p + k_o s$$

상대거리 티타 = 현재 상대거리 - 목표 상대거리.

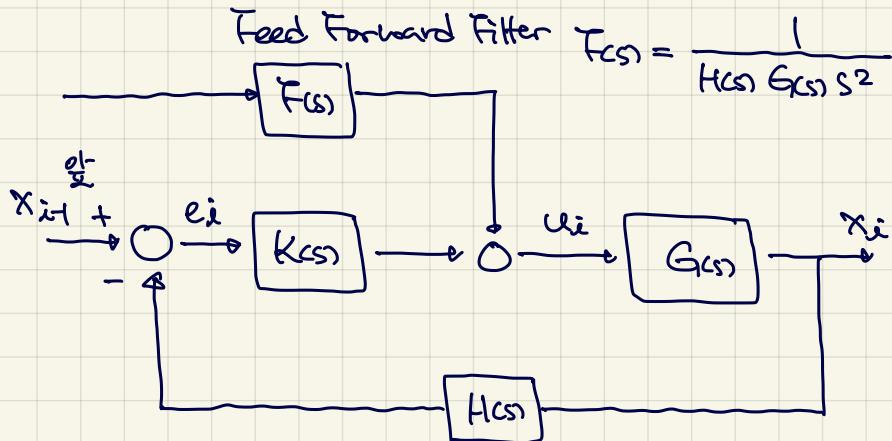
$$e = (\dot{x}_{des} - \dot{x}_i) - h\ddot{x}_i$$

앞 차량 뒤 차량 적차량의 속도

$$E = x_{des} - x_i - h s x_i = x_{des} - \underbrace{(1+h s)}_H x_i$$

$$\underline{H(s) = 1+h s \text{로 정의}}$$

CSCC



$$e_i = X_{i-1} - H(s)X_i \quad [(X_{i-1} - H(s)X_i)K(s) + T_{CS}]G(s) = X_i$$

$$\frac{x_i}{x_{i-1}} = \frac{[T_{CS}s^2 + K(s)]G(s)}{1 + H(s)G(s)K(s)} = \frac{1}{H(s)} = \frac{1}{h + 1}$$

↑  
속도 비례 상수  $h > 0$   
 $\rightarrow |G(s)| \leq 1$

$h$  를 SCC 보다 더 작게 설정 가능.

### 09 Active Steering

능동형 조향 시스템.

차상 기계부품  $\rightarrow$  전자부품  $\times$  by wire

전선.  
AFS  
MDPS

ex Steer by wire

ASS  
AFS  
motor driven  
power steering

SBW  
Steer by wire  
active steering sys.  
active front steering

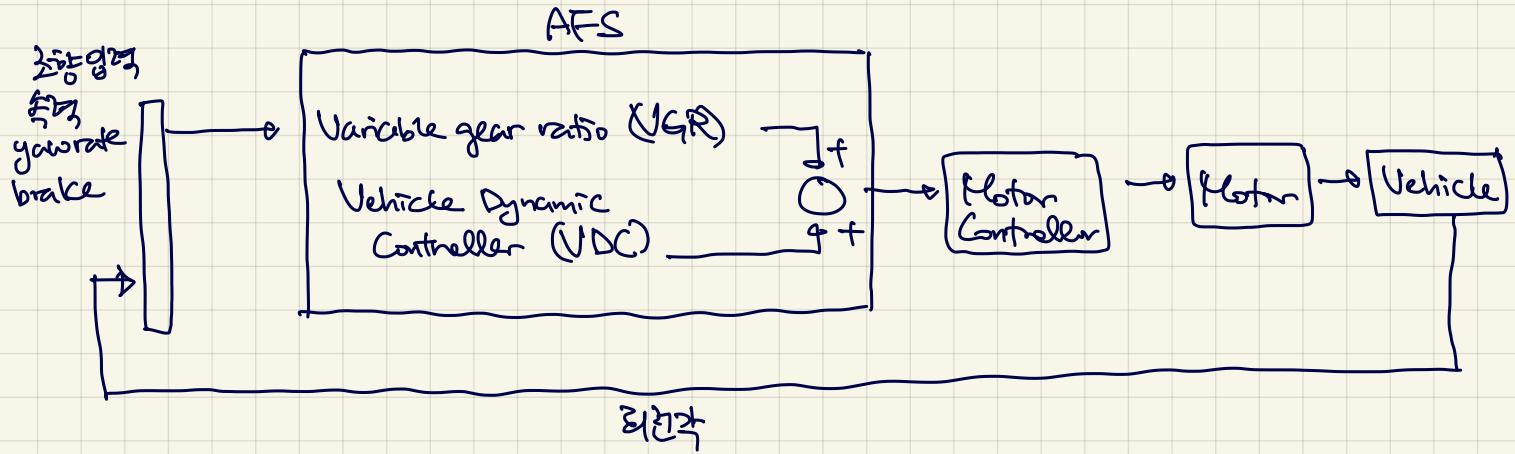
AFS : 유성기어로 기계적 연결 유지  
입력 조향각 대비 출력 조향각 조절 가능

SBW 이 비해 기계적 연결로 고장 대처 가능.

운전자 조향 + 추가 능동 조향각.

실제 자동차에 가해진 조향각  $\longleftrightarrow$  자동차가 가지는 조향각  
 $\rightarrow$  능동적으로 조향각 가감!

## AFS 주행기 구조



VGR 구속 조향비 낮춰고 같은 조향 바퀴 회전↑  
고속 조향비 높인다 같은 조향 바퀴 회전↑

VDC : 미끄러짐 감지 조향 잘 따르게

주차하는 경우 뒷바퀴 앞바퀴 반대 회전 선회반경 ↓ : 연동형 가변 조향  
고속도로 주행 뒷바퀴 앞바퀴 같은 방향 회전 안회반경↑

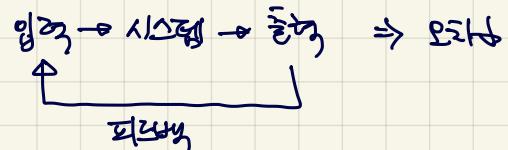
## 2 차관 기술

제어

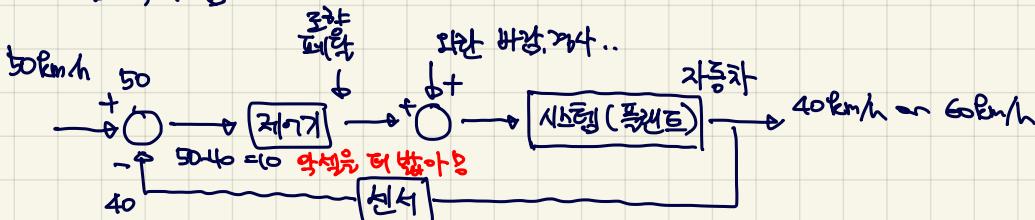
open loop

closed loop; feedback control

↳ 시스템의 정확한 정보 옮기로 원하는 속도값 얻을 수 있음



### 피드백 시스템



출처값은 측정하기 위한 센서  
모차 계산에 사용됨.

사람의 눈

## 3 시스템 테스트 개발 process.

Simulation? : ECU electronic control unit : 자동차 제어 장치

각 부품 제작 실험 → 비용 시간 ↑

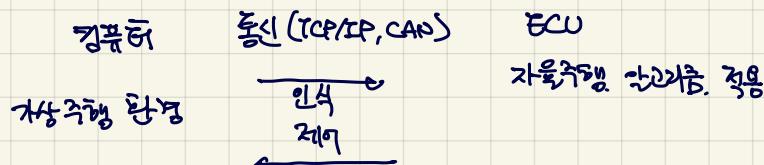
→ 모의 환경에서 검증 테스트

SILS : Software in the Loop 시뮬, HILS hardware in the loop 시뮬, VILS : vehicle in the loop simulation

SILS : 자동차 기능. 설계한 자동주행 구현 과정을 컴퓨터에서 검증.

HILS : 차운위어와 소프트웨어 보드 활용. SW로 보호하기 어려운 부품(시스템)을 직접 사용. Simulation

ex) ECU HILS



VILS : 실제 도로 주행 검증 전 최종 단계

데이터 수집.

제어 차운주행 알고리즘 계산

계산된 값은 액세스, 조향 입력으로 출처.

## 4. 기계적 시스템 모형

자율주행차: 기계적 시스템

ex) 사양: 키나이 체중, 성능, 차: 질량 스프링 상수, 댕퍼.

병진운동, 회전운동.

병진운동: 힘의 역학 관계를 모델링  $\Sigma F = ma$  질량, 스프링, 댕퍼.

질량: 힘과 가속도의 상관 계수

스프링: 힘과 위치의 상관 계수.

댕퍼: 힘과 속도의 상관 계수

차량 전진, 회전, 섬광등.

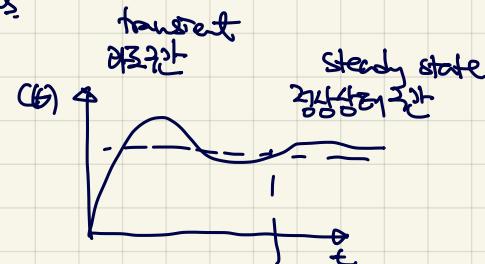
회전운동: 토크 각속도 능률 제모법칙.

한승모멘트: 토크와 각속도의 상관 계수

스프링 상수: 토크와 각도의 상관 계수.

토션倔, 댕퍼: 토크와 각속도의 상관 관계

Steering, wheels



## 5. 시간 응답.

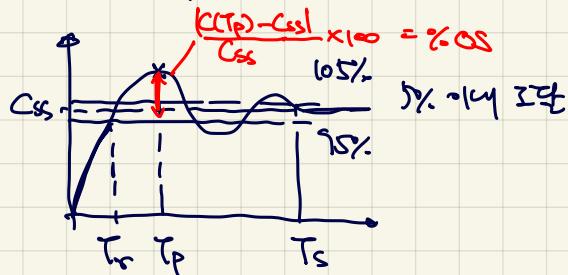
입력에 대한 시간 응답 특성

$\begin{cases} \text{Undamped sys. } \zeta = 0 \\ \text{Underdamped sys. } \zeta < 1 \\ \text{critically damped sys. } \zeta = 1 \\ \text{overdamped sys. } \zeta > 1 \end{cases}$	$\zeta$ : damping ratio.
--	--------------------------

시간 응답: 정상상태 오차 steady state error

정착 시간 $T_s$ settling time	$\left. \begin{array}{l} \text{상승 시간 } T_r \text{ rise time} \\ \text{피크 시간 } T_p \text{ peak time} \\ \% \text{ 오버슈트 } \% OS \end{array} \right\} \text{ 과정 시간} $
---------------------------	--

$T_s$   $T_r$   $T_p$



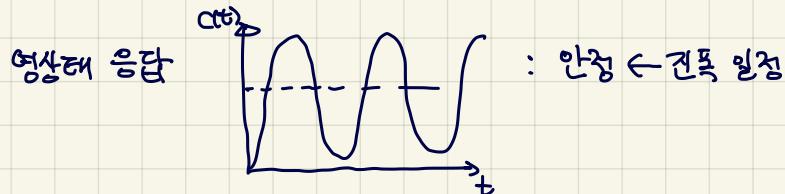
정상 상태 오차: 입력에 의한 최종값과 목표값 사이의 차이.

## 6 제어 안정성

영상태 응답, 영입력 응답

안정한 시스템 : 시스템의 출력이 발산X

전체 응답 [ 영상태 응답 zero state response  
영입력 응답 zero input response 초기 조건에 의존하는



영입력 상태 : 모든 초기 조건에서 안정해야 안정.



제어기 왜 설계?

불안정한 시스템을 안정하게 만들기 위해.

## 09 파워트레인

엔진 → 클러치 → 변속기 → 추진축 → 증감속 장치  
차동기어 장치 → 차축 → 구동 바퀴

클러치 : 바퀴와 엔진 분리 변속시 기어마모 방지

변속기 : 주행 상황별 토크 변화

증감속장치 : 출혁 토크 증대를 위해 감속

차동기어 : 회전각 차이로 달라지는 각 바퀴 속력 보정.  
엔진 동력을 양 바퀴에 분배.

제어기 ECU engine control unit 충기 압축 폭발 배기  
TCU transmission control unit

## 10. Torque vectoring

↳ 양쪽 바퀴 속도 달라져 구동 가능해.

자동차 자세 제어. 인더스터어, 오버스터어, 선회반경 차등 장치.

Torque Vectoring : 자동차의 바퀴로 전달되는 토크 크기/방향 변화  
방향 초기.

understeer / oversteer  $\leftarrow$  트raig. 바퀴 접지력 ↕

Understeer : 운전자가 입력한 조향 강도보다 자동차가 더 돌아감.

ex) 빠른 속도로コーナ링. 앞 바퀴 접지력 ↕  $\rightarrow$  자동차가 바깥으로 밀려남.

Oversteer : 운전자가 입력한 조향 보다 자동차가 더 돌아감.

ex) 뒷바퀴 접지력 ↕, 운행속도 증가. 저속에서도

후륜 구동  $\rightarrow$  도로 마찰력 낮을 때  $\rightarrow$  느린 속도로 오버스터어

선회 반경 : 자동차가 직진  $\rightarrow$  양쪽 바퀴 동일 속도.

자동차가 선회 주행  $\rightarrow$  양쪽 바퀴 회전 속도 다른

차동 기어 : differential gear 양쪽 바퀴의 회전 속도 다른 수 있음.

회전 저항이 차리 설정된 기계적 시스템에 의해 수동적으로 움직이는 시스템

주행 상태에 따라 저항으로 토크 달라짐.

토크 벡터링 : 수동적인 차동장치와 달리 능동적으로 양쪽 바퀴의 토크 다르게

differential gear의 가능 유익

under oversteer 이 발생하지 않도록 안정적 주행 유익.  $\leftarrow$  적절한 토크를 출력.

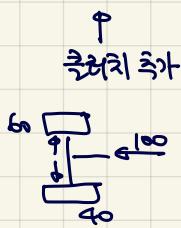
$\Rightarrow$  주행 상황 판단  $\rightarrow$  알맞은 토크 출력.

주행 상태 : 주행 속도. Steering 속도. Steering 각도. 횡방향 가속도.

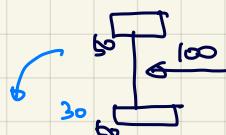
Torque Vectoring 시스템 : "최적" 토크를 계산하는 알고리즘으로 각 바퀴에 적절한 토크 전달.

- 차동기어 제어. 브레이크 제어  $\Rightarrow$  양쪽 바퀴에 토크를 다르게 하여 자동차가 회전하는 모멘트 없을 만큼

yaw moment



좌우 바퀴의 브레이크 제어  
회전 저항 X  
토크의 크기 제한



## II. Brake by wire / X by wire 차로 종류, 구조.

X by wire : 기계식 방식으로 조작하던 것을 전기적인 wire로 조작  
선로

Brake by wire : 전기적인 wire로 제동

- 정교한 제어 가능
- 응답특성 개선
- 설계 수정 용이
- 유압 대비 경량 연비 향상

구성 요소

- 센서: 운전자 밟은 브레이크 압축 측정
- ECU: 센서 측정값으로 감속력 계산, 모터 출력량 강도 계산
- 모터: 모터로 브레이크 퍼드 꽉 잡아줌  
액츄에이터

단점: 기계식 → 전자학

고장이 허락

센서: 측정오차

ECU: 계산 오류, 센서전류X

모터: 영구제동 수행 오류.

## Brake by wire 종류

전기유압식 (EHB) : 유압식 전기기계식 중간 wet type

센서 → ECU → 모터 → 유압 조작,  
(기존 기계식)

전기기계식 (EBS) : 유압X dry type 모터만으로 작동 쇄기형, 디스크 회전 아침 않음.

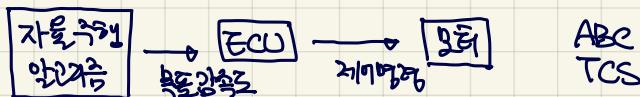
하이브리드 타입 Hybrid EHB: 앞 바퀴 태입 : 고장이 많음.

뒤 바퀴 태입 주차 브레이크도 같이 연동해서 쉽게 구현 가능.

구동 ECU 으로 원하는 제동 구현  
브레이크 밟지 않아도 ECU로 제동,  
자동주행에 적용 등이  
ECU에 원하는 값 → 모터가 제동

제동시스템 : 강속, 긴급정지시스템 자동차 속력, 주행 모드

자율주행 안전제품 강속 → 목표 강속 값



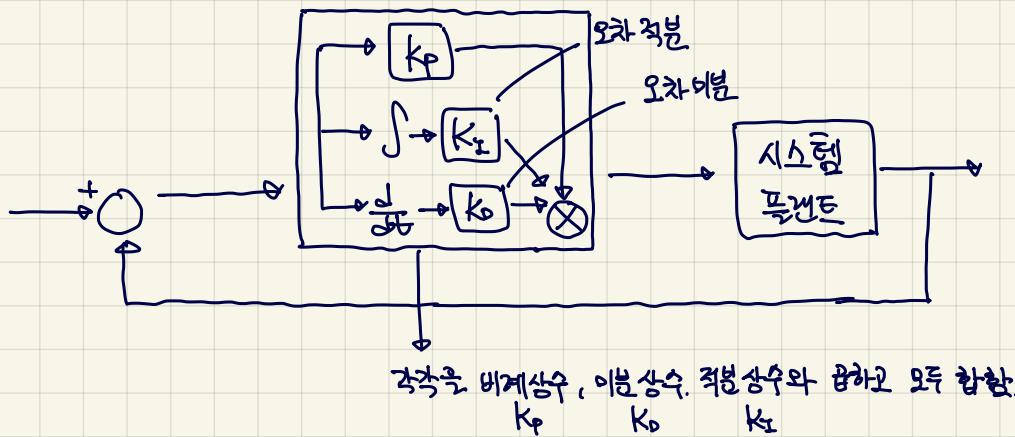
자율주행 제동시스템: 목표 계로를 원하는 속도로

# 07. PID 제어 개요.

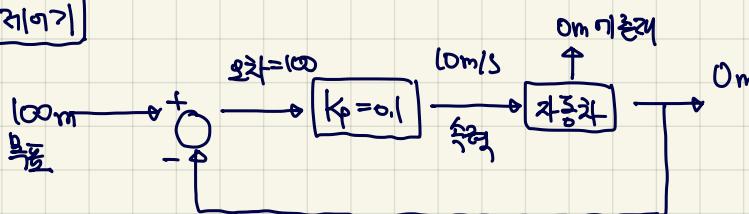
피드백 제어: 목표값과 같임없이 비교함으로써 오차가 발생할 때마다 항상 줄이도록 대상에 조작을 가하는 제어.

PID 제어: 피드백 제어의 가장 대표 방법.

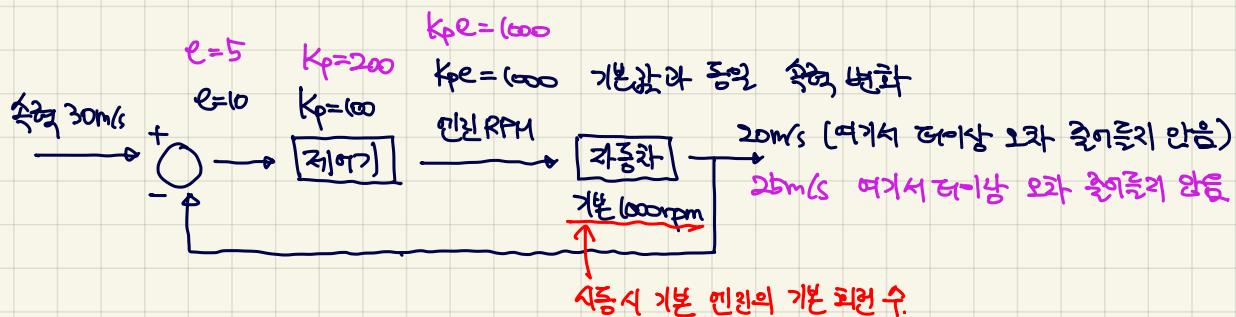
$$\boxed{\text{회피값}} - \text{목표값} = \text{오차}$$



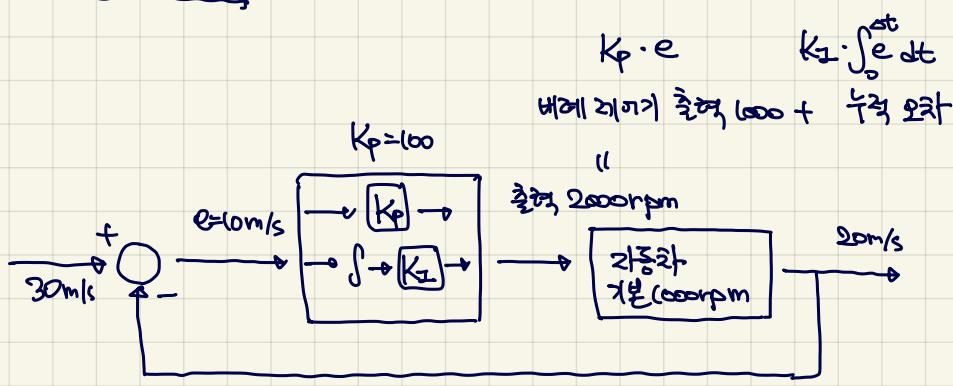
## 비례제어기



$$\text{오차} = 50m \quad v = 5m/s \quad \text{오차} = 10m \quad v = 1m/s \dots \text{오차} \rightarrow 0m$$

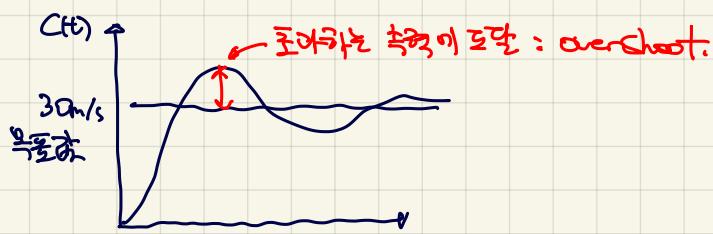


정상상태 오차를 줄이기  $\Rightarrow$  적분제어



$\Rightarrow$  적분제어로 30m/s 도달

적분제어의 이득상수 값이 큰 경우



⇒ 이분제어로 overshoot 완화

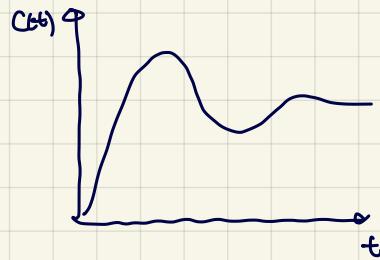
적분제어의 큰 값으로 오차가 빠르게 줄어듦



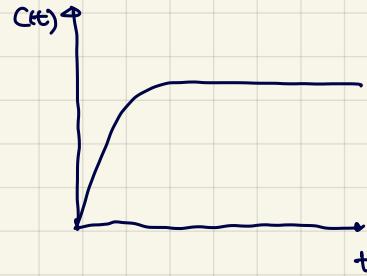
비분제어로 응수 빠른 초기



적분제어의 빠른 수렴 완화



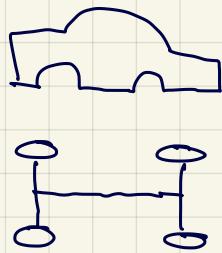
→ 비분제어



## 08. 제동. 구동. 조향. 혼가 시스템

Body : 차체. 겉부분. 사고 학을 싫는 부분

Chassis : 차대 엔진 조향 동력. 제동



Chassis.

엔진.

동력 전달장치

[이륜 엔진 구동바퀴] FF FR RR MR

[사륜]

엔진 바퀴

- 제동시스템 : 브레이크. 사용 브레이크. 보조 브레이크. 파킹 브레이크.

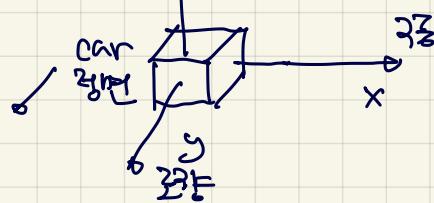
혼가시스템 : 서스펜션. 샤시 스프링. Stabilizer.

승차감

제동. 감속 시 적절한 힘의 액. 확보 조종 안정성. 조향

조향시스템 : 운전자 힘  $\rightarrow$  유압식. 전자식  $\rightarrow$  전동식. 퍼포먼스 EPS

<자동차 운동학>



### ⑩. 제동시스템

상용 브레이크. 보조 브레이크. 주차 브레이크

유압식 브레이크  $\leftarrow$  파스칼의 원리

전자 제어 장치 ABS TCS

Anti Lock Brake System : 타이어 영취도 관성으로 움직임 drift 느낌  $\rightarrow$  브레이크 잡았다 풀었다.  
 $\rightarrow$  최대 정지 마찰력

Traction Control System : 바퀴의 구동력 제어 도로 미끄러울 때.

가속페달, 브레이크 함께 신호.

공회전 바퀴 속력 측정. 각 바퀴 회전수 차이

ESC Electronic Stability Control

= TSC + ABS

## 17. 조향시스템 구동.

조향비: 2. 가변 조향비. 각동 방법.

조향비: 앞 바퀴가 10 회전하는데 필요한 운전대의 회전 각도 비율.

운전대 회전각  
앞 바퀴 회전각



회전 운동의 토크 ← 기울비)

ex) 운전대 토크: 10 바퀴 토크: 50 → 운전대 5번 회전. 바퀴 1번 회전.

$$\text{일} = \text{힘} \times \text{면적} = \text{토크} \times \text{각도}$$

[인증]

조향비 운전대 토크 10 / 바퀴 토크 50 5 : 1

조향비 ↑ 핸들 만이 돌림 힘↑

조향비 ↓ 핸들 크게 돌림 힘↑

가변 조향비

조향비 ↑ → 운전대 회전만 틀므로 조향↑

차량 속도 핸들 각도 → 조향비 선택/안정성↑

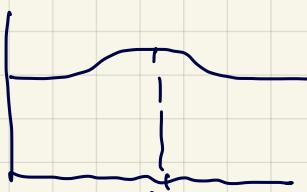
운전대 기울수 ↓

바퀴 기울수 ↑

운전대에 걸은 힘을 둘여 바퀴 돌림.

조향비 ↑ 핸들 많이 돌리는 대신 적은 힘.

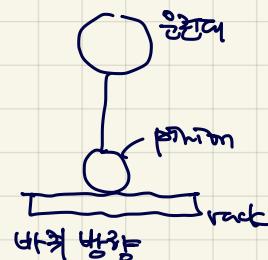
조향비 ↓ 핸들 크게 돌리는 대신 큰 힘



조향각  
운전대

고속으로 조향비↑↑ 적정성  
커브 구간 조향비↓ 음직임

Rack and pinion : 회전 ↔ 직선.

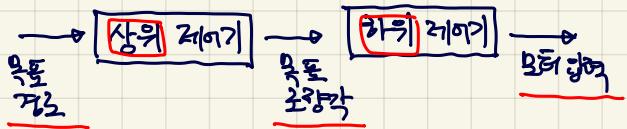
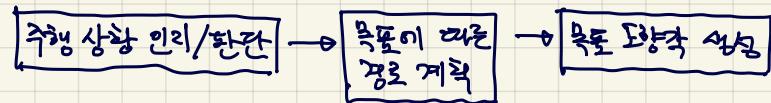


## 20. 횡방향 ADAS system.

조향 → 운전자는 횡방향으로 움직인다라고 느낄.

횡방향 제어 ADAS

상위 레이어  
하위 레이어



Forward collision avoidance

LKS, LCS FCA BSD

Blind spot detect.

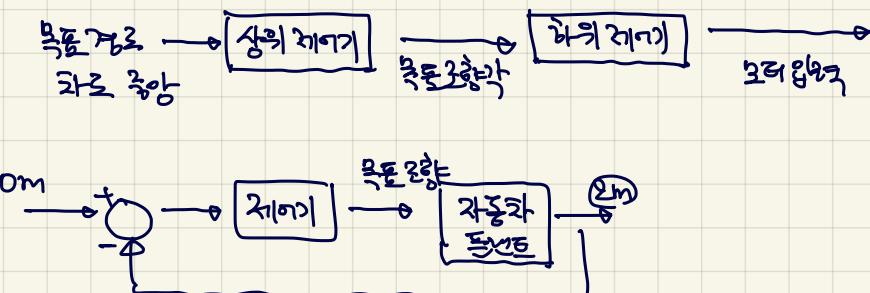
## 21. 차로 이탈 방지 보조 LKA

카메라로 차선 인식 → 조향

카메라 차선 인식 → 가상 차선 생성. → 가상 차선과 차량의 중심 사이의 1차 예상  
횡방위를 오차

오차 기준에 따라 제어

고속도로 차선 폭 : 3.6m



## 24 자동주차시스템

핸들 : 조향 액션/브레이크 : 박차/회전 기어 : 전진 후진

자동주차와 자율주행 : 인지, 판단, 제어.

주차 공간 인지 및 자동차 현재 위치 계산

↓

실시간 연산을 통한 회적 경로 계획

↓

정확 정밀 제어

Rear - Autonomous Emergency Brake R AEB

## 27 ESC Electronic Suspension Control

Anti rolling control

Anti squat / diving control

Anti pitching control

Anti bouncing control

## 11. 차량의 운동학 모델

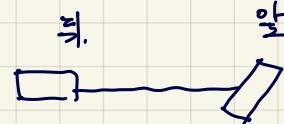
Kinematics.

물체 움직임 동역학 : eq. of motion  운동학 : 시스템 운동방식 보여줌 침x 물체의 이동, 회전, 속도, 가속도 등을 알려져 기하학으로 나타냄.

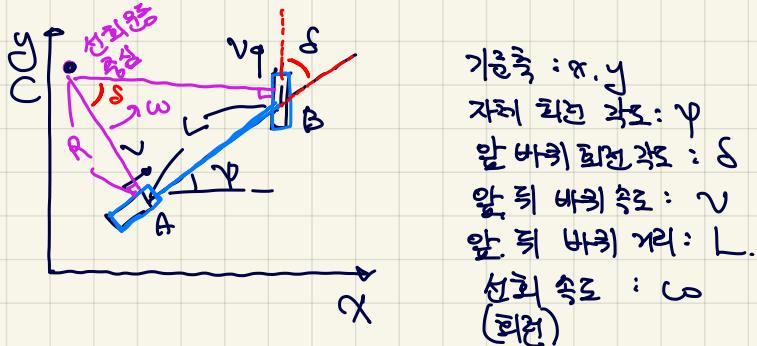
자동차 / 로봇 움직임

"가정"

- ① 평면 위
  - ② 다이어 노면 미끄러짐 X
  - ③ 앞바퀴만 회전
  - ④ 양쪽 바퀴 하나로
  - ⑤ 자동차의 속력 일정
- )  $\Rightarrow$  Kinematic Bicycle model.



기하학을 이용한 자동차 운동학의 변수 정의.



기준축 : x, y  
자체 회전 각도 :  $\psi$   
앞 바퀴 회전 각도 :  $\delta$   
앞 뒤 바퀴 속도 :  $v$   
앞 뒤 바퀴 거리 :  $L$ .  
선회 속도 :  $\omega$   
(회전)

운동학 모델 유도

$$\omega = \frac{v}{R} \quad \tan \delta = \frac{L}{R} \quad \Rightarrow \quad \omega = \frac{v}{\frac{L}{\tan \delta}} = \frac{v \tan \delta}{L}$$

$x_r, y_r$  자동차 뒷 바퀴 위치

$$\dot{x}_r = v \cos \psi \quad \dot{y}_r = v \sin \psi \quad \dot{\theta} = \frac{v \tan \delta}{L}$$

앞 바퀴 기준

$$\dot{x}_f = v \cos(\psi + \delta)$$

$$\dot{y}_f = v \sin(\psi + \delta)$$

$$\dot{\theta} = \frac{v \tan \delta}{L}$$

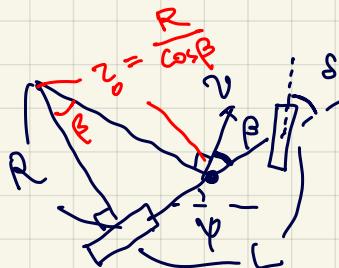
## 목기 중심에서 차가 어떻게 굽게임?

속도 방향은 앞바퀴의 속도 방향과 뒷바퀴의 속도 방향 중간

속도가 자동차의 차체와 각도를 이루 ; 목기 중심 속도 방향과 차체 축 사이 각도.

→ 횡방향 미끄러짐 side slip

횡방향 미끄러짐 각도  $\beta$



$$\dot{x}_c = V \cos(\phi + \beta)$$

$$\dot{y}_c = V \sin(\phi + \beta)$$

$$\omega_c = \frac{V}{R \cos \beta} \quad \dot{\theta}_c = \frac{V \cos \beta}{R} = \frac{V \tan S \cos \beta}{L}$$

$L = R \tan S$  목기 중심의 회전 반경. D

3: 자동차 차체 직선 시 굽토.

운동학 : 낮은 속도, 미끄러짐 없을 때 적용

## 12 차량의 동역학 모델

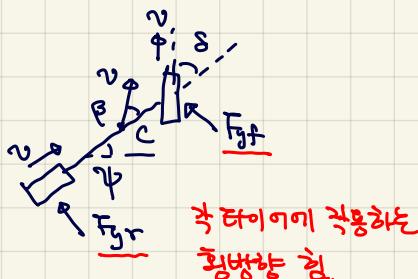
동역학: 운동의 원인이 되는 힘을 고려해서 힘이 시스템에 어떤 영향을 미치는지 분석하는 방식

고속도, 미끄러짐 → 운동학 X  $\Rightarrow$  [동역학]

가정 ① 자동차의 속력 일정

② 서스펜션, 횡방향 경사각, 공기역학 등 무시

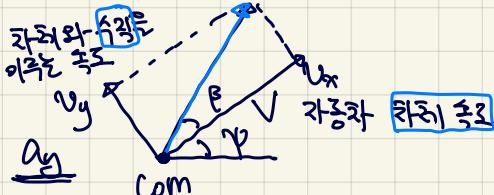
동역학 변수 정의



$$\text{자동차 가속도 총합} = \frac{a_y}{g} = \frac{\text{차체의 횡방향 가속도}}{\dot{\gamma}} + \frac{\text{회전으로 인한 구심가속도}}{\omega^2 R}$$

$$\frac{V^2}{R} = \frac{(RW)^2}{R}$$

Side slip  
“목기 중심의 속도가 차체와 이루는 각도.”



$$\tan \beta = \frac{V_y}{V_x}$$

$$\beta \ll 1 \Rightarrow \beta = \frac{V_y}{V_x}$$

## 각 가속도

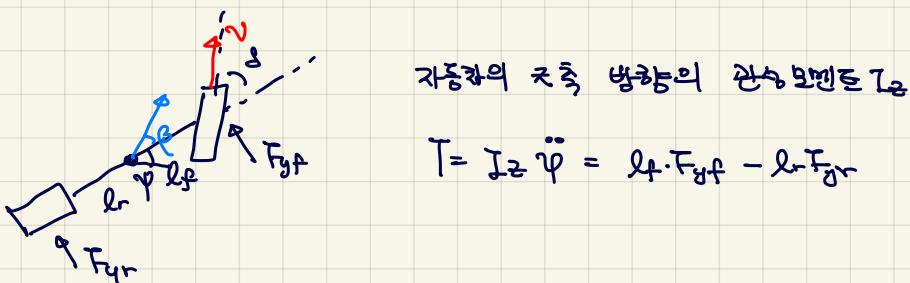
$$V = R\omega \quad \ddot{\gamma}_y = \dot{\gamma} \quad \omega = \dot{\psi} \quad \beta \doteq \frac{\dot{\gamma}_y}{\dot{\gamma}_x} = \frac{\dot{\gamma}}{V}$$

$$\ddot{\gamma}_y = R\omega^2 + \frac{\ddot{\gamma}}{V} = \underbrace{V\ddot{\psi} + V\ddot{\beta}}_{V\ddot{\beta} ?} \quad \alpha =$$

## 횡방향 힘의 합

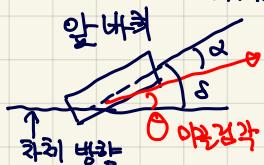
$$F = F_{yf} + F_{yr} = m\ddot{\gamma}_y = mV(\ddot{\beta} + \ddot{\psi})$$

## 관성 보멘트      $T = I_z \times \alpha$



## 타이어 미끄럼 각도

타이어 코너링 강성  
Stiffness



조향각과 실체 각도의 차  $\alpha = \delta - \theta$

미끄럼각에 의해 자동차의 횡방향 힘 발생

Tire Cornering Stiffness:  $C_a$  ← 각도, 환경에 따라 비선형적일 수도

$$F_y = C_a \cdot \alpha$$

$$\left[ \begin{array}{l} F_{yf} = C_{af} \alpha_f = C_{ar} \left( \delta - \beta - \frac{l_f \ddot{\psi}}{V} \right) \\ F_{yr} = C_{ar} \alpha_r = C_{ar} \left( \beta + \frac{l_r \ddot{\psi}}{V} \right) \end{array} \right]$$

$$F_{yf} + F_{yr} = mV(\ddot{\beta} + \ddot{\psi})$$

$$I_z \ddot{\psi} = l_f F_{yf} - l_r F_{yr}$$

$$\ddot{\beta} = \frac{-(C_{af}l_f + C_{ar}l_r)}{mV} \beta + \left( 1 - \frac{C_{af}l_f^2 + C_{ar}l_r^2}{mV^2} \right) + \frac{C_{af}}{mV} \delta$$

$$\ddot{\psi} = \frac{(C_{af}l_f - C_{ar}l_r)}{I_z} \beta - \frac{C_{af}l_f^2 + C_{ar}l_r^2}{I_z V} + \frac{C_{af} \delta}{I_z}$$

$$x = [y \dot{y} \varphi \dot{\varphi}] \quad u = \delta$$

$$\ddot{x} = Ax + Bu \quad \text{상태 공간 모델}$$

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{(C_{ap} + C_{ar})}{mV} & 0 & -\frac{(C_{ap} \dot{L}_p - C_{ar} \dot{L}_r)}{mV} - \nu \\ 0 & 0 & 1 & 0 \\ 0 & -\frac{(C_{ap} \dot{L}_p - C_{ar} \dot{L}_r)}{I_2 V} & 0 & -\frac{(C_{ap} \dot{L}_p^2 + C_{ar} \dot{L}_r^2)}{I_2 V} \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ C_{ap}/mV \\ 0 \\ C_{ap} \dot{L}_p / I_2 \end{bmatrix}$$

### (3). Pure Pursuit 대포 경로 추이 횡방향 제어.

주어진 경로를 기하학 (기하학)을 이용해 추종하는 방식

차량 이끄러짐, 윤식, 다이나믹한 상황 무시.

고속, 횡방향 가속도 충격 시 사용 불가

타이어 비선형적 움직임 없을 때 사용.

Pure pursuit의 구성요소 : 전방 목표점 Look ahead distance  $\ddot{P}$   
자동차 운동학 - Bicycle model.

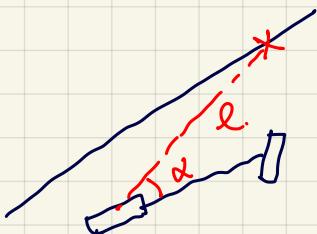
Parameter 정의 방법.



앞-뒤 바퀴 사이 거리: L.

Look ahead distance ( $l$ )는 차량의 뒷 바퀴 기준

선회운동 시 차량 선회 궤적의 중심과 수렴되는 곳. 이 뒷바퀴

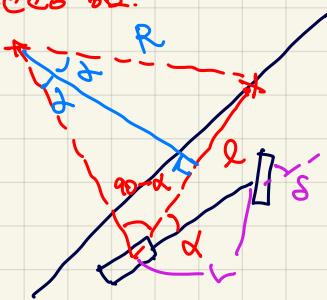


뒤 바퀴 기준으로 차량이 가야 할 임의의 목표 지점 설정

$l$ : look ahead distance

자동차 바퀴 얼마나 회전 시 목표 경로 따라갈 수 있겠지

회전운동 중심.



$$\frac{l}{\sin \alpha} = \frac{R}{\sin(\frac{\pi}{2} - \alpha)}$$

$$K = \frac{1}{R} = \frac{\frac{2\sin \alpha}{l}}{\frac{\pi}{2}}$$

$$\frac{l}{2\sin \alpha \cos \alpha} = \frac{R}{\cos \alpha}$$

$$\frac{l}{2\sin \alpha} = R$$

$$\frac{l}{\sin \alpha} = \frac{2R}{\frac{1}{\cos \alpha}}$$

선회 반경

선과 반경 설정  $\rightarrow$  자동차 바퀴 회전 각 계산.

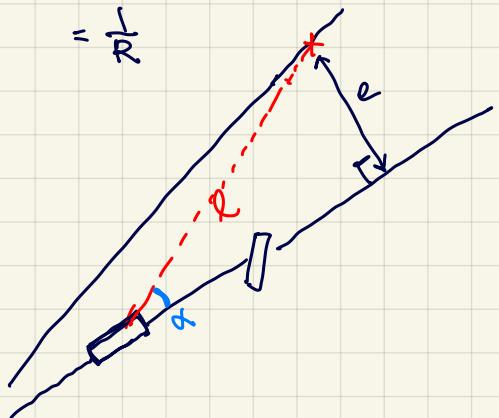
$$K = \frac{\frac{2\pi m}{l}}{R} = \frac{l}{R}$$

$$\tan \delta = \frac{l}{R} \Rightarrow$$

$$\text{조향각} \quad ③ = \tan^{-1} \frac{l}{R} = \tan^{-1} k l = \tan^{-1} \frac{\frac{2\pi m}{l} l}{l} = \tan^{-1} 2\pi m$$

앞차  
look ahead point

따라 뛰다  
따라 뛰다



$$\sin \alpha = \frac{e}{l} \quad K = \frac{\frac{2\pi m}{l}}{l} = \frac{2}{l^2} e$$

$$K = \frac{2}{l^2} e : \text{상수} \times \text{오차 } e \Rightarrow \text{비례비율}.$$

자동차 조향각 =  $\boxed{\frac{2}{l^2} e}$ 에 의해 결정  $\rightarrow$  비례비율이 실제

$K = \frac{2}{l^2} e$ 에서  $l$  look-ahead distance는 engineer가 설정해주는 값

- 〈  $l$  크면 큰 오차 조향 각도
- 〈  $l$  작으면 작은 오차 조향 각도

예)  $l$  작은 경우: 전방주시거리 작은 경우.

고속에서 전방주시거리가 작으면 즉  $l=1$  작으면 오차가 작아도 자동차 차동차 속도 조향값이 큼  
 $\rightarrow$  조향값 빠르게 변화  $\rightarrow$  위험

제어?

$\Rightarrow$  전방주시거리  $l$ 을 속도에 비례하는 상수로 설정.

$l = \boxed{K v_x}$  engineer 등의 tuning 해야 하는 parameter

고속에서 큰  $l$ 을 가지도록 함.

오차가 천천히 줄어들게 하여 안정적 주행 가능

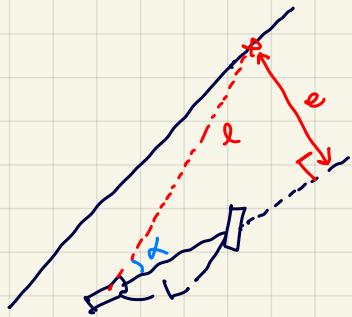
## 14. Stanley 기법. : 기학을 이용한 경로 следing 기법

↳ Sebastian Thrun 교수 팀에서 사용

DARPA : 자율주행 대회

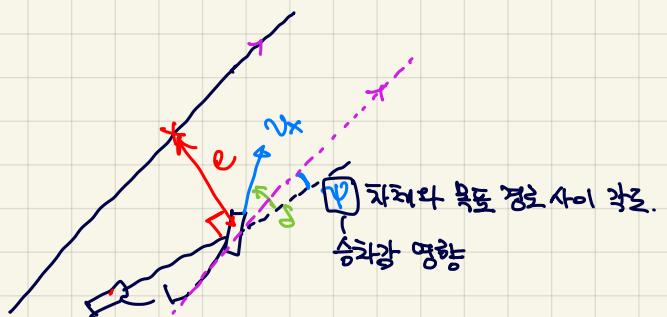
Stanford 팀의 자동차 이름 Stanley

Pure Pursuit 기법.



일정한 look ahead distance 설정 후  
그 위치에서의 횡방향 오차 정의  
횡방향 오차만으로 조향각 계산

Stanley 기법.



look ahead distance 없이 앞바퀴 기준  
횡방향 오차를 정의 (수직거리)  
횡방향 오차, 헤딩 오차를 모두 고려

Stanley 기법 원리

1. 헤딩 오차 고려

차량 경로 방향 및 차등차의 축방향에 별도로 헤딩 오차를 없애기 위한 조향각 계산

$$\text{조향각} = \text{헤딩 오차}: \delta = \psi$$

2. 횡방향 오차 고려

Pure Pursuit 방법 응용.

- 조향각을 횡방향 오차에 비례하여 제어  $\delta = \tan^{-1} k L = \tan^{-1} \frac{2}{L^2} e L$
- 차량 속력에 따라 조향 속도 비례 크지 않게 제어.

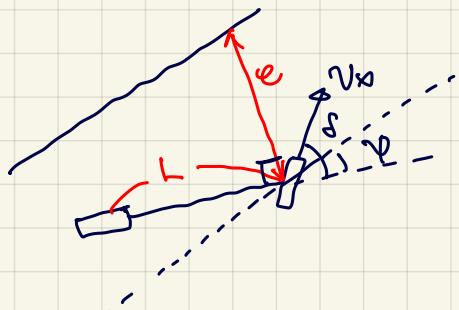
$$\Rightarrow \delta = \tan^{-1} \frac{k e}{v_x} \quad \text{tuning parameter} \\ \text{---} \quad l = k v_x \quad \text{---} \quad \text{횡방향 속도}$$

횡방향 오차를 줄이기 위한 조향 각도.

$$\boxed{\delta = \psi} + \boxed{\delta = \tan^{-1} \left( \frac{k e}{v_x} \right)} \Rightarrow \boxed{\delta = \psi + \tan^{-1} \left( \frac{k e}{v_x} \right)}$$

어떻게 경로 추종 (횡방향 오차  $\rightarrow 0$ )

횡방향 오차 변화량



$$\dot{\psi} = -v_x \sin(\psi - \delta) \quad \delta = \gamma + \tan^{-1} \frac{ke}{v_x}$$

$$\dot{\psi} = -v_x \sin \left( \tan^{-1} \left( \frac{ke}{v_x} \right) \right) = \frac{-ke}{\sqrt{1 + \left( \frac{ke}{v_x} \right)^2}} \quad e \ll 1 \text{ 이면}$$

$$\Rightarrow \dot{\psi} \propto -\frac{ke}{v_x} \text{ 미지의 변화량}$$

$e^{-kt}$ 로 시간이 지남에 따라 오차가 지속함수 형태로 감소  $\rightarrow 0$ 으로 수렴.

제어 성능 향상을 위한 응용 방식.

1. 속도 상수 추가

$$\delta = \gamma + \tan^{-1} \frac{ke}{v_x} \quad v_x \text{ 가 작으면 } e \text{ 가 작아도 조향각 변화가 큼.}$$

$$\rightarrow \delta = \gamma + \tan^{-1} \left( \frac{ke}{h + v_x} \right)$$

속도가 작아도 0 근접 방지

2. 텁乒乓 추가

조향 각도가 채팅 오차 횡방향 오차에 비례하게 설계  $\rightarrow P\Delta$  설계

(+)

텅乒乓  $\rightarrow PD$  설계

텅乒乓  $\downarrow$

P $\Delta$  설계  $\delta = \gamma + \tan^{-1} \left( \frac{ke}{v_x} \right)$

PD 설계  $\delta = [k_p] \gamma + [k_d] \dot{\gamma} + \tan^{-1} \frac{ke}{v_x}$

채팅오차

## 15. Full State Feedback.

상태공간 모델: 물리적계는 암력, 질량 상태변수의 1차 미분방정식으로 표현하는 수학적 모델.

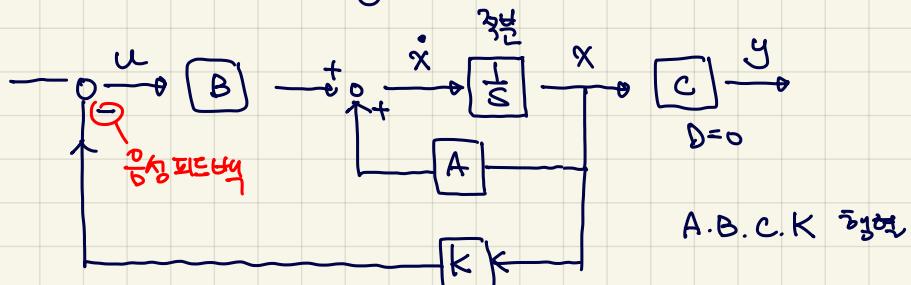
피드백: 결과를 다시 원인에 적용, 그 결과를 늘어거나 줄임.

특성방정식: 일반함 구하기 위해 각 항의 특성 살면서 방정식 만드는 것.

Full State Feedback: 상태변수로 제어.

상태공간 모델  $\dot{x} = Ax + Bu$

$$y = Cx + D \quad (D=0 \text{ 으로 가정})$$



$$\text{상태제어기 } u = -Kx$$

$\uparrow$   
제어기 이득값 설정

$$\text{시스템 안정성 확인: } \dot{x} = Ax + Bu \quad u = -Kx \text{ 대입}$$

$$\begin{aligned} \dot{x} &= Ax - BKx \\ &= (\underline{A-BK})x \end{aligned}$$

시스템 행렬의 고유값 속수 부분이 음수 일 때 안정. P

$A-BK$ : 고유값 확인해서 시스템 안정 상태 확인.

$A-BK = \boxed{Ac}$ : closed loop 시스템 행렬

분모=0인 모양  
극점은 제거시킨다.

Pole Placement 방식: 상태 행렬이 원하는 고유값을 갖도록

상태공간에서의 시스템 행렬의 고유값 = 전달함수의 극점과 같은 역할 pole

$$\begin{bmatrix} \ddot{x}_1 \\ \ddot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{f}{m} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} f(t)$$

입력: 힘

상태변수: 위치, 속도

출력 확인 가능 (위치 속도)

$u = -Kx$  대입. 응성 피드백,

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -1 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u = \begin{bmatrix} 0 & 1 \\ -1 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} - \begin{bmatrix} 0 \\ 1 \end{bmatrix} K \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$K = [k_1 \ k_2]$

시스템 행렬의 모든 파라미터 : 1

$$= \begin{bmatrix} 0 & 1 \\ -1 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} - \begin{bmatrix} 0 \\ 1 \end{bmatrix} [k_1 \ k_2] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 \\ -1-k_1 & -1-k_2 \end{bmatrix}}_{Ae} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$Ae\tilde{x} = \lambda \tilde{x} \quad (Ae - I\lambda) \tilde{x} = 0 \quad |Ae - I\lambda| = 0$$

$$\begin{vmatrix} -\lambda & 1 \\ -1-k_1 & -1-k_2-\lambda \end{vmatrix} = 0 \quad \lambda(\lambda + k_2 + 1) + (1 + k_1) = 0$$

$$\lambda^2 + (1+k_2)\lambda + (k_1+1) = 0 \quad \lambda \in \text{LHP or RHP or 각각 한 쪽.}$$

Pole 선정  $\rightarrow K = [k_1 \ k_2]$  결정.

안정적인 시스템을 위한 Pole의 위치  $-1, -5$

$$(\lambda+1)(\lambda+5) = 0 \quad \lambda^2 + 6\lambda + 5 = 0 \Rightarrow \underbrace{k_2 = 5}_{\text{"}}, \underbrace{k_1 = 4}_{\text{"}}$$

## 6. Linear Quadratic 제어 (LQ 제어)

비용함수. 최적화 ARE Algebraic Riccati Equation.

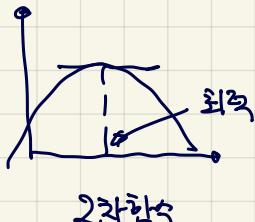
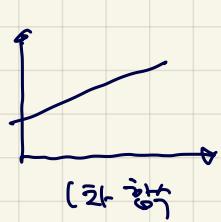
LQ controller : 최적화 방식을 이용한 제어기

LQ controller 설계 방식

01. 시스템의 상태방정식 구하기 State Space equation
02. Performance Index / 비용함수 정의 목적함수.
03. Algebraic Riccati Equation 풀어서 K값 구하기.

Cost function : 최적화하고자 하는 변수의 상관관계를 통해 이루어진 함수

$$\text{성과} = f(\text{시간})$$



최적화: 최대/최소값을 찾는 과정. 비용함수 2차 4차.

LQ controller Cost function

$$J = \int_0^{\infty} (x^T Q x + u^T R u) dt$$

기준치

오차 × 시스템의 속도  
시스템의 입력.

자동차의 속도 제어

$Q \uparrow R \downarrow$  양쪽 제어의 연비 ↓ /  $Q \downarrow R \uparrow$  연비 ↑ but 목표 속도 추구↑

Algebraic Riccati Equation.

LQ 제어가 실제 시스템의 조건을 반영하도록 하는 수식.

- 시스템 안전
- 비용함수 최적화

$$PA + A^T P - PBR^{-1}B^T P + Q = 0$$

$\dot{x} = Ax + Bu$   $Q, R$  는 weight matrix. 방정식에서 미지수는  $P$  뿐.

$P$ 의 해는 2차의 조건 만족

$$u = -\underline{R^{-1}B^T P x} \quad u = -Kx \text{에서 } K = R^{-1}B^T P$$

## 17. State Observer

(X) 추정값: 실제 시스템이 인가된 입력을 수학적 모델에 동일하게 입력하여 추정한 상태 변수의 값

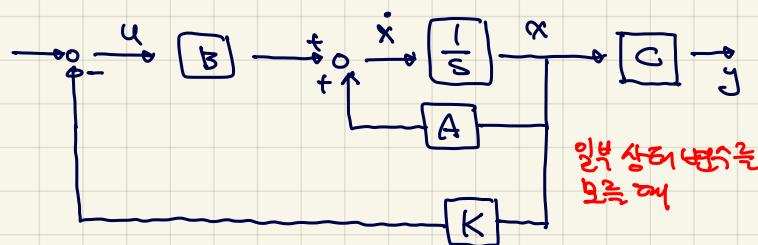
State observer: 시스템이 가지고 있는 상태를 알기 위한 장치

관측기 이득: 관측 또는 추정한 모델의 상태 변수와 실제 모델의 상태 변수가 같아지도록 설계하는 것

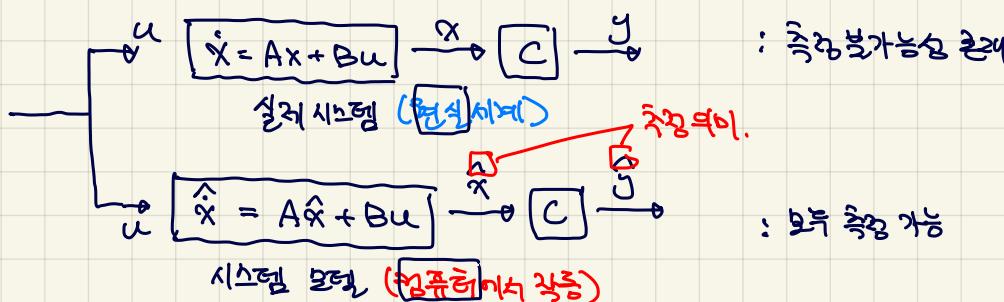
State observer: 입력,  출력, 상태 변수를 정의하고 시스템 모델링. → 비용  $J$

or State estimator: 선서로 측정 가능/불가능

: 선서로 얻을 수 없거나 측정 불가한 값을 수식으로 만들 또는 추정하는 것



상태 관측기 설계 ← 상태 공간 모델.



목표  $\dot{\hat{x}} = \dot{x}$  정확한 시스템 모델 구현 수식으로 예상 가능하도록.

각각 동일한 입력  $\rightarrow$  풀역 비교. 틱, 바울하면 수학적 모델 체크

시스템 모델 상태변수가 실제 시스템 상태 변수와 같아지면?

$$\dot{\hat{x}} = A\hat{x} + Bu + L(y - \hat{y}) : \text{Luenberger Observer 방식}$$

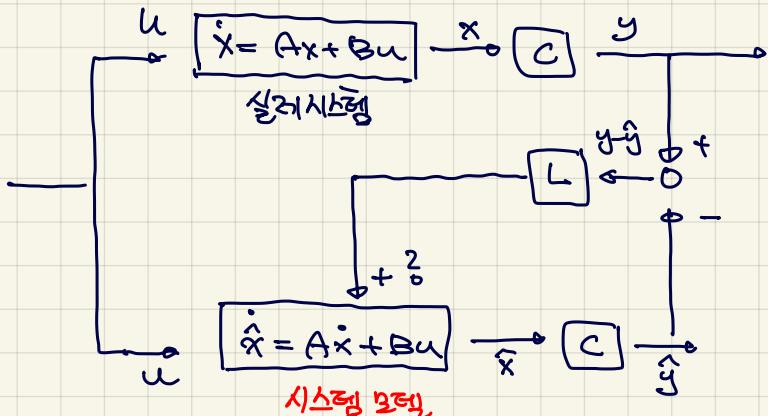
출역 차이  $\rightarrow$  L 만큼 이득  $\rightarrow$  x 보정.

$$L \text{은 } \dot{\hat{x}} = \dot{x} \text{ 이 되도록 정의 } e = \dot{x} - \dot{\hat{x}} \quad e \rightarrow 0$$

$$\begin{aligned} \dot{\hat{x}} - \dot{x} &= Ax + Bu - [A\hat{x} + Bu + L(y - \hat{y})] = A(x - \hat{x}) - L(y - \hat{y}) \\ &= (A - LC)(x - \hat{x}) \end{aligned}$$

$$\dot{e} = (A - LC)e$$

$$\dot{e} = (A - LC)e \quad A-LC \text{의 고유값이 LHP에 존재 시 } 0 \text{ 으로 수렴.} \rightarrow \dot{x} = \ddot{x}$$



현실에서 알 수 없던 x 값을  $\dot{\hat{x}}$ 을 통해 알 수 있게 됨.  $y$ 와  $\dot{\hat{x}}$ 의 오차가 0이 되도록 할 때  
 $\dot{\hat{x}} \rightarrow x$ ;  $\dot{\hat{x}}$ 을 오차가 보정된 상태로 초기화함으로

### 18. 상태 관측기 기반 제어기 설계

State observer & State Feed back.

$$\dot{e} = (A - LC)e$$

$\downarrow$

$$\dot{x} - \dot{\hat{x}} \rightarrow 0 \text{ 이도록 설계.}$$

Pole Placement : 고유값을 이리 정해두고 L값 찾기

$$\dot{x} = (A - BK)x$$

Full state feedback

$$\dot{e} = (A - LC)e$$

$e = \dot{x} - \dot{\hat{x}}$  State observer

기계시스템

State feed back에서는 모든 상태 변수 측정 가능하다는 가정.

State observer에서는 측정만 알 수 있다고 가정.

$$y = C \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$y = C \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = [1 \ 0] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \underline{\underline{x_1}} \quad \text{축소 방정식.}$$

$$\dot{e} = (A - LC)e \quad \begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \end{bmatrix} = \left( \begin{bmatrix} 0 & 1 \\ -1 & -1 \end{bmatrix} - \begin{bmatrix} l_1 \\ l_2 \end{bmatrix} [1 \ 0] \right) \begin{bmatrix} e_1 \\ e_2 \end{bmatrix} = \left( \begin{bmatrix} 0 & 1 \\ -1 & -1 \end{bmatrix} - \begin{bmatrix} l_1 & 0 \\ 0 & l_2 \end{bmatrix} \right) \begin{bmatrix} e_1 \\ e_2 \end{bmatrix}$$

$$= \begin{bmatrix} -l_1 & 1 \\ -l_2 & -1 \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \end{bmatrix}$$

특수화 방정식  $(A - LC) - I \lambda | = 0$

$$\begin{vmatrix} -l_1 - \lambda & 1 \\ -l_2 - \lambda & 1 - \lambda \end{vmatrix} = 0 \quad (l_1 + \lambda)(\lambda + l_2) + (l_1 + l_2) = 0 \quad \lambda^2 + (l_1 + l_2)\lambda + l_1 l_2 = 0$$

고유값을  $\underline{\lambda_1, \lambda_2}$  으로 가정.  $(\lambda_1 + \lambda_2)(\lambda_1 + \lambda_2) = 0 \quad \lambda_1^2 + 5\lambda_1 + 6 = 0 \quad \lambda_1 = -1, \lambda_2 = -4$   
LHP에 존재해야 안정; 0으로 수렴

### 관측기 이득 L 설정 시 고려할 점

- L 값을 설정할 때 오차 수렴 속도 고려
- 상태 변수 추정치가 시스템 동역학보다 빨리 수렴해야 함  
L이 K보다 빠르게 수렴.

자율주행 차의 청방향 움직임.

1. 자동차 청방향 움직임을 동역학 모델로 표현.

$$\dot{x} = Ax + Bu \quad x = [y \dot{y} \gamma \dot{\gamma}]^T \quad u = \delta$$

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{(C_{af} + C_{ar})}{mV} & 0 & -\frac{(C_{af} l_f - C_{ar} l_r)}{mV} - V \\ 0 & 0 & 0 & 1 \\ 0 & -\frac{(C_{af} l_f - C_{ar} l_r)}{I_2} & 0 & -\frac{(C_{af} l_f^2 + C_{ar} l_r^2)}{I_2 V} \end{bmatrix}$$

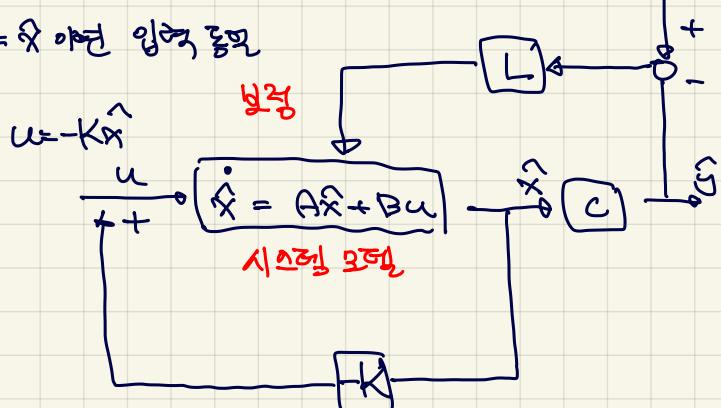
$$B = \begin{bmatrix} 0 \\ \frac{C_{af}}{mV} \\ 0 \\ \frac{C_{af} l_f}{I_2} \end{bmatrix}$$

$$x = [y \dot{y} \gamma \dot{\gamma}]$$

모른다고 가정



$\dot{x} = f(x)$  이면 입력 출력



# 19. 모델 예측 제어 MPC

자율주행 + 여려 불의

이산시간 상태공간 모델, 제약 조건

자율주행  $\leftarrow$  조향각 너무 빠르면 안됨

횡방향 가속도 너무 크면 안됨.

자동차 엔진 RPM 너무 높으면 안됨

브레이크의 최대 감속도 양 정해짐

$\Rightarrow$  제약 조건을 만족하며 동시에 복도 성능 한계  $\Rightarrow$  MPC

상태변수나 출력을 예측  $\rightarrow$  적절한 비용함수/제약 조건 설정  $\rightarrow$  최적화 실행.  $\rightarrow$  제어 입력 계산  
여러 제약 만족 가능

Discrete time state space model

이산시간 상태공간 모델

이산화 Discretization

Sampling time 기준으로 연속적인 시간  $\rightarrow$  불연속적 시간  $\rightarrow$  계산 수행

Analog

Digital

연속적인 시간

- 현재 입력, 상태변수 기반
- 시스템의 상태 변수가 어떻게 변하는지 알기 위해 미분 방정식 사용.

이산시간

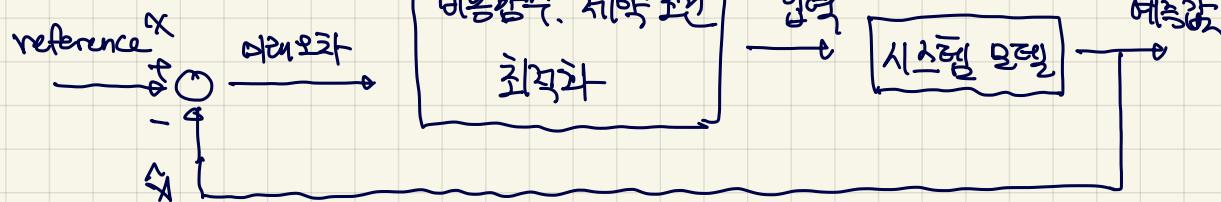
- 현재 입력, 상태변수 기반
- 시스템의 상태 변수가 다음 Sampling time에서 어떻게 바뀌는지 알기 위한 형태로

$$x(k+1) = \Phi x(k) + T u(k)$$

Sampling time

이산화된 [시스템 행렬, 입력 행렬]

MPC 구조



이산화 시스템 모델로  $k+N$  번째 상태변수와 툴박스 등을 예측

$$x(k+1) = \phi x(k) + T u(k)$$

$$\begin{aligned} x(k+2) &= \phi x(k+1) + T u(k+1) = \phi(\phi x(k) + T u(k)) + T u(k+1) \\ &= \phi^2 x(k) + \phi T u(k) + T u(k+1) \end{aligned}$$

$$\begin{aligned} x(k+3) &= \phi x(k+2) + T u(k+2) = \phi^3 x(k) + \phi^2 T u(k) + \phi T u(k+1) + T u(k+2) \\ &\vdots \\ x(k+N) \end{aligned}$$

유리는 시간에 예측한  $(k+1)$  번째 예측값  $x(k+1|k)$   $\vartheta = 1 \sim N$

→ 미래 오차를 악수 있음.

최적화 :  $k+N$  번째 시간까지 입력

$$\underbrace{u(k+1|k) \ u(k+2|k) \ u(k+3|k) \dots \ u(k+N|k)}_u$$

최적화를 통해 툴박스의  $\sqcup$  을 찾음.

$k+N$  번째 상태변수 / 툴박스를 최소

$$J = \sum_{j=k}^{k+N} \underbrace{x^T(j|k) Q x(j|k)}_{\text{제약함수}} + \underbrace{u^T(j|k) R u(j|k)}_{\text{제약함수}} + \text{제약함수}$$

$$\begin{aligned} u_{min} &\leq u \leq u_{max} \\ x &\leq x_{max} \end{aligned} \quad \left. \begin{array}{c} \text{제약함수} \\ \vdots \end{array} \right.$$

MPC에서 사용한 최적화 계산 Solver QP solver

## 20. 지능제어 퍼지제어, 실증환경, 강화학습.

인공지능 활용 제어.

### 퍼지 제어 Fuzzy control

예/아니오, 참/거짓, if/else., on/off  $\Rightarrow$  이분법적 논리. 특별 논리.

SCC 예시 Smart Cruise Control  
앞 차량과의 간격을 조절하는 SCC

간격이 100m 보다 크면  $\rightarrow$  멀다.  
간격이 20m 보다 작으면  $\rightarrow$  가깝다.

현재 간격 50m 라면 간격이 멀다/가깝다? 이 때의 제어 방식은?

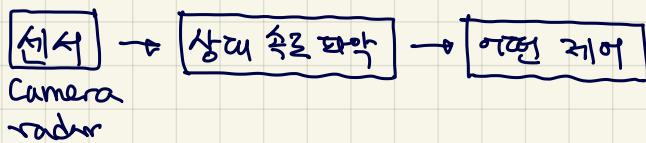
Fuzzy 제어 : 예측한 것을 숫자로 나타내고 이런 것을 바탕으로 어떤 제어를 할지 나타내어주는 방식.

0~1 사이의 양의 실수 값을 반환.  $\rightarrow$  제어

퍼지 제어기를 이용해 앞차량과의 간격을 조절하는 SCC 설계.

### 상대 거리, 상대거리 변화량

상대 거리 변화량			
	감소	유지	
상대 거리	가깝다.	감속	감속
	적당하다	감속	속도유지
	멀다	속도유지	가속
		가속	가속



선서로 측정한 값의 합침



[상대거리]

[상대거리 변화: 상대속도]

가깝다 20m

감소

-5 m/s 다가온다

적당하다 60m

유지

0 m/s 유지

멀다 100m

증가

5 m/s 떠나간다.

## 상대거리, 상대속도 $\Rightarrow$ 상대변수

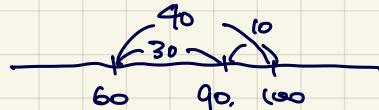
9개의 집합 중 하나에 포함 시 수용하는 제어

감속 현재속도 0.8 20% 감소

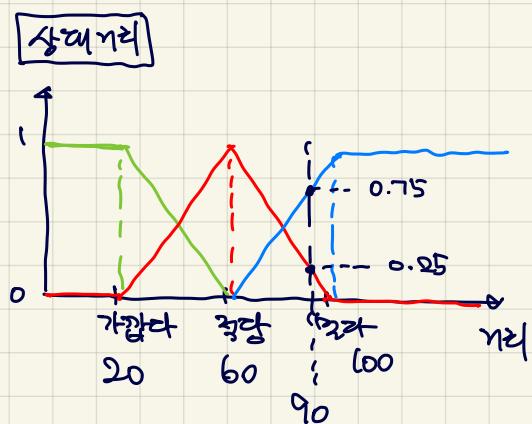
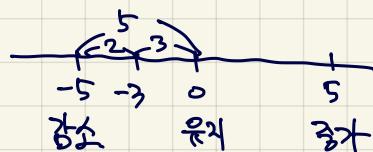
속도유지 현재속도 1

가속 현재속도 1.2 20% 증가

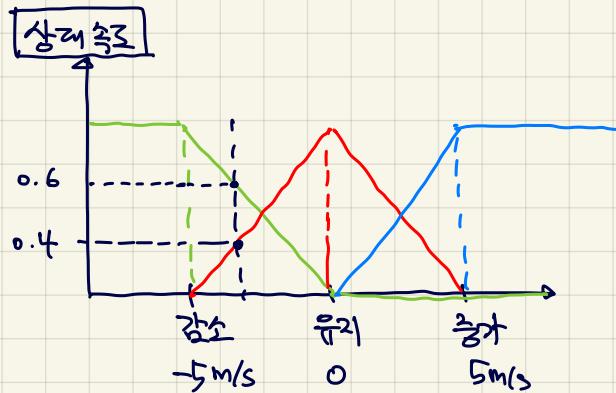
상대거리 90m    상대속도  $\rightarrow 3m/s$   $\Rightarrow$  멀다  $\times 0.25$  적당하다  $\times 0.25$



$\Rightarrow$  유지  $\times 0.4$ , 감소  $\times 0.6$



$$\text{멀다} = \max(0.75, 0.25)$$



$$\text{감소} = \max(0.6, 0.4)$$

상대거리: 멀다  
상대속도: 감소

$\Rightarrow$  속도유지 가속 = 0,  $a=0$

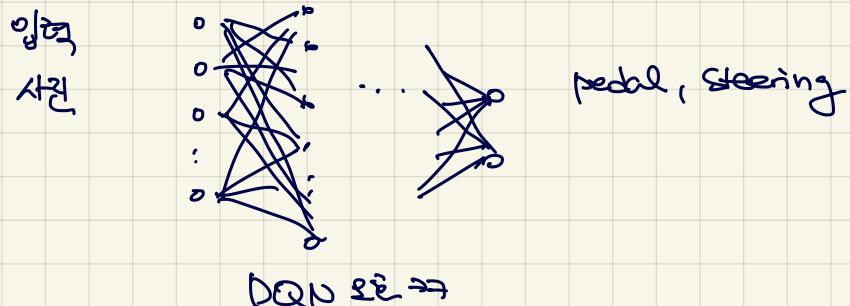
**도로상태**

매매한 것  $\rightarrow$  어느 집합에 속하든지 0과 사이 숫자로 구분  $\rightarrow$  제어

# 심층신경망 Deep Neural Network.

End to End 자율주행 방식.

입력에서 출현까지 신경망에 의해서만 연결된 방식.



센서 장착하고 운전자가 지정된 구역을  $\Rightarrow$  학습  $\Rightarrow$  주행 상황 입력되면  
달리며 데이터 수집  
학습된 데이터로 주행.

단점: 제어의 이유 분석 어렵음.

강화학습.

