Introduction

모델예측 제어
(Model Predictive Control)

상태변수나
출력을 예측

↓

적절한
비용함수와
제약조건 설정

다양한 형태의
제약조건 만족

↓

최적화 실행

↓

제어입력 계산

Introduction

컴퓨터

✓ Sampling Time 기준

연속적인 시간 → 불연속적 시간 → **계산 수행**

Analog / Digital

이산화 (Discretization)	연속적인 시간에서 나타내었던 시스템을 컴퓨터가 처리할 수 있도록 바꿔주는 과정
이산화된 시간 (Discrete-Time)	이산화된 시간
이산시간 상태공간 모델 (Discrete-Time State-Space Model	이산화된 시간에서 시스템에 상태공간 모델을 나타낸 것

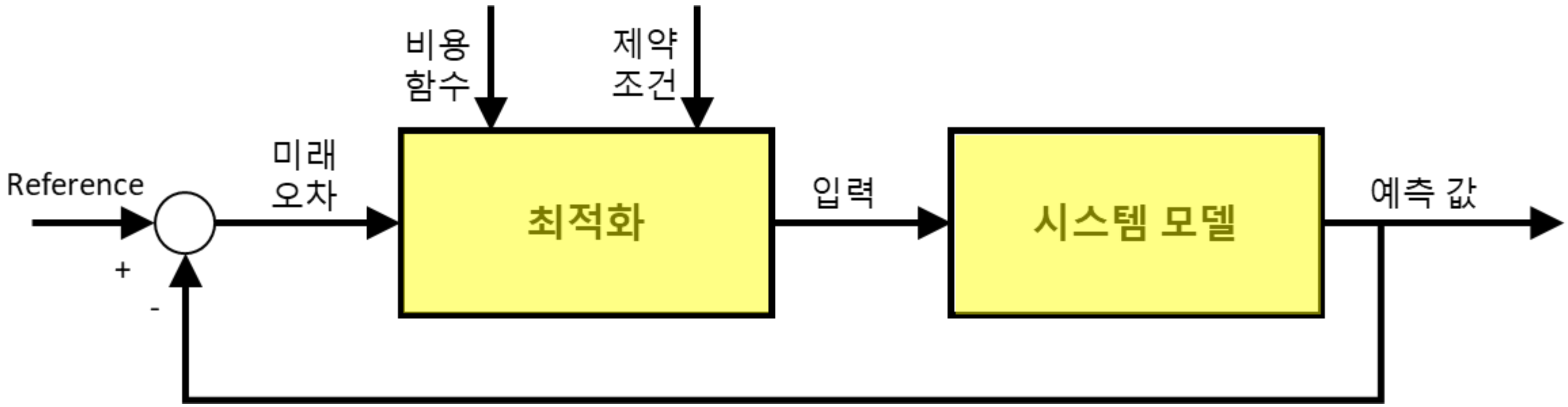
이산시간 상태공간 모델

$$\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + B\mathbf{u}(t)$$

$$\mathbf{x}(k+1) = \Phi\mathbf{x}(k) + \Gamma\mathbf{u}(k)$$

이산시간

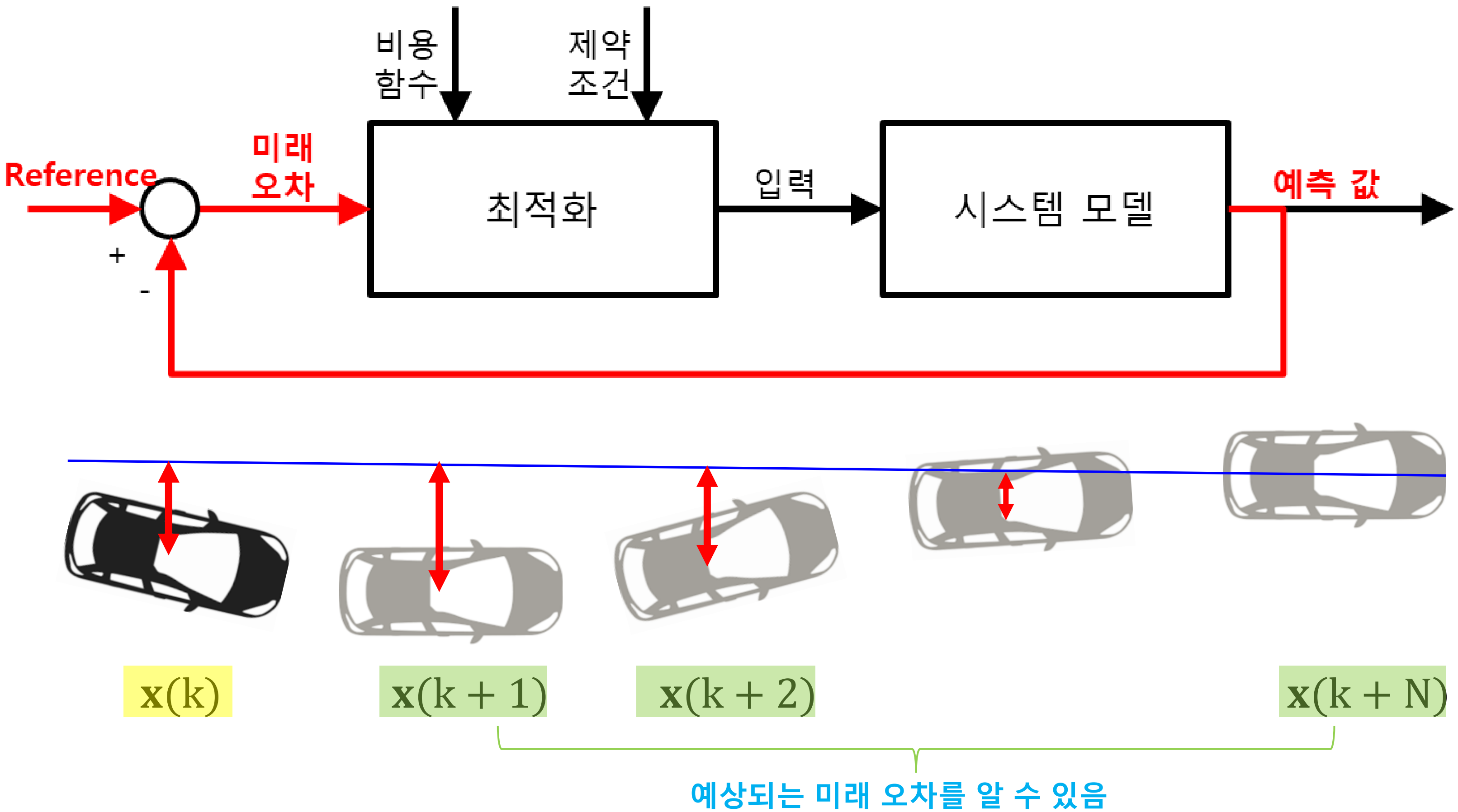
Introduction



k+N번째의 상태변수와
출력 등을 이산화된 상태공간
모델을 이용하여 예측

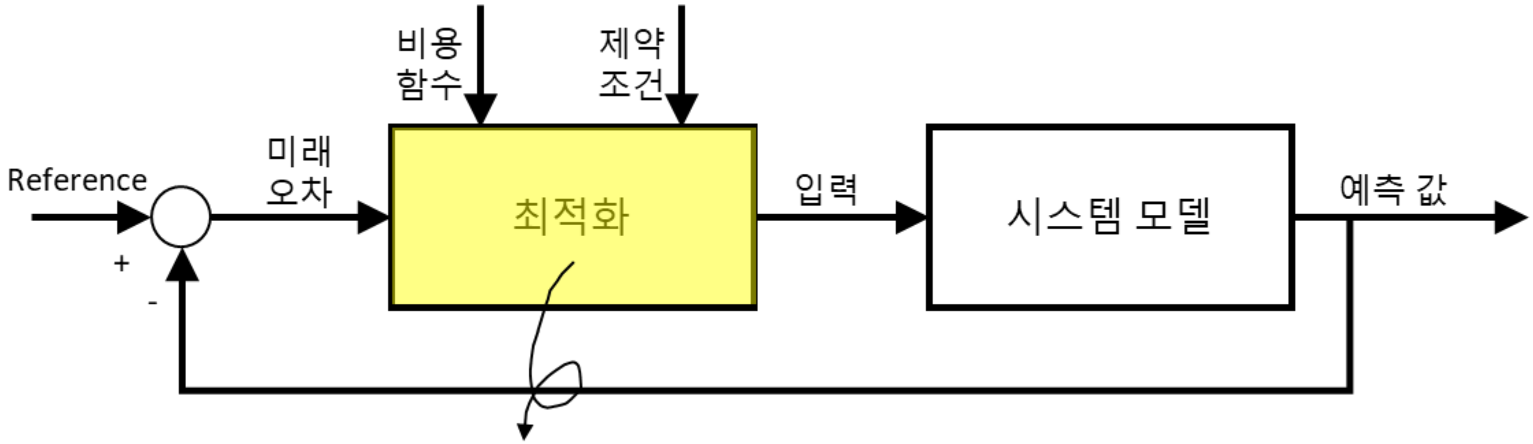
$$
\begin{aligned}
\mathbf{x}(\mathrm{k}+1) &= \Phi\mathbf{x}(\mathrm{k}) + \Gamma\mathbf{u}(\mathrm{k}) \\
\mathbf{x}(\mathrm{k}+2) &= \Phi\mathbf{x}(\mathrm{k}+1) + \Gamma\mathbf{u}(\mathrm{k}+1) \\
&= \Phi(\Phi\mathbf{x}(\mathrm{k}) + \Gamma\mathbf{u}(\mathrm{k})) + \Gamma\mathbf{u}(\mathrm{k}+1) \\
&= \Phi^2\mathbf{x}(\mathrm{k}) + \Phi\Gamma\mathbf{u}(\mathrm{k}) + \Gamma\mathbf{u}(\mathrm{k}+1) \\
&\vdots \\
\mathbf{x}(\mathrm{k}+\mathrm{N}) &= \Phi^N\mathbf{x}(\mathrm{k}) + \cdots
\end{aligned}
$$

Introduction



k라는 시간에서 예측한 k+i번째의 예측 값 $x(k+i|k)$ $x(k+N|k)$

Introduction



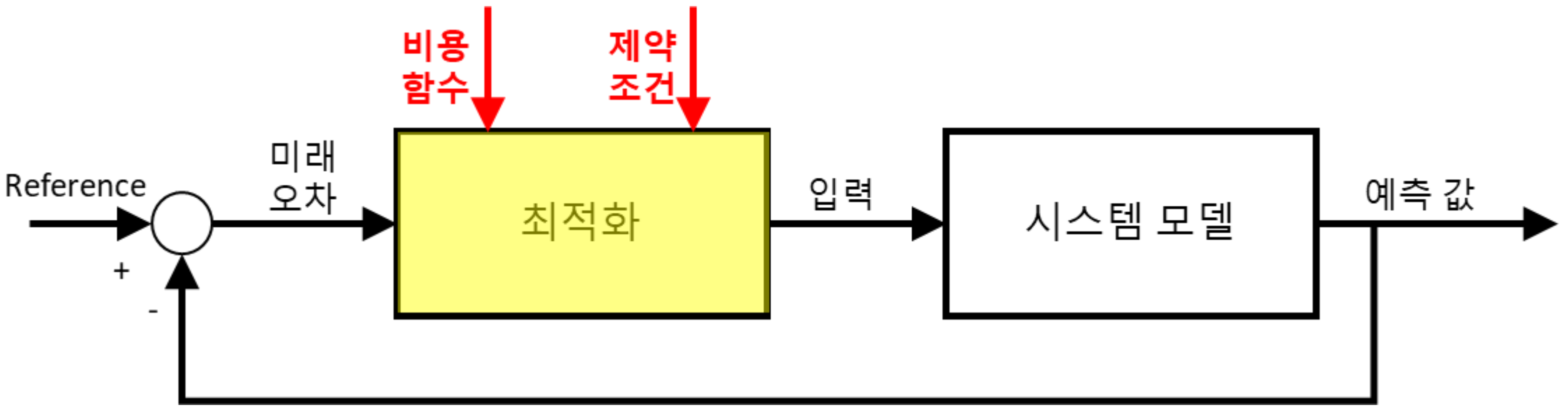
제어의 성능과 제약조건을 동시에
만족하는 최적의 입력값을 만들어야 함

k+N번째까지 어떠한
입력들을 넣어야 조건을 만족

$$U = \underbrace{u(k|k)}_{\text{현재입력}}, \underbrace{u(k+1|k), u(k+2|k), \ldots, u(k+N|k)}_{\text{미래입력}}$$

최적의 입력을
찾는 문제

Introduction



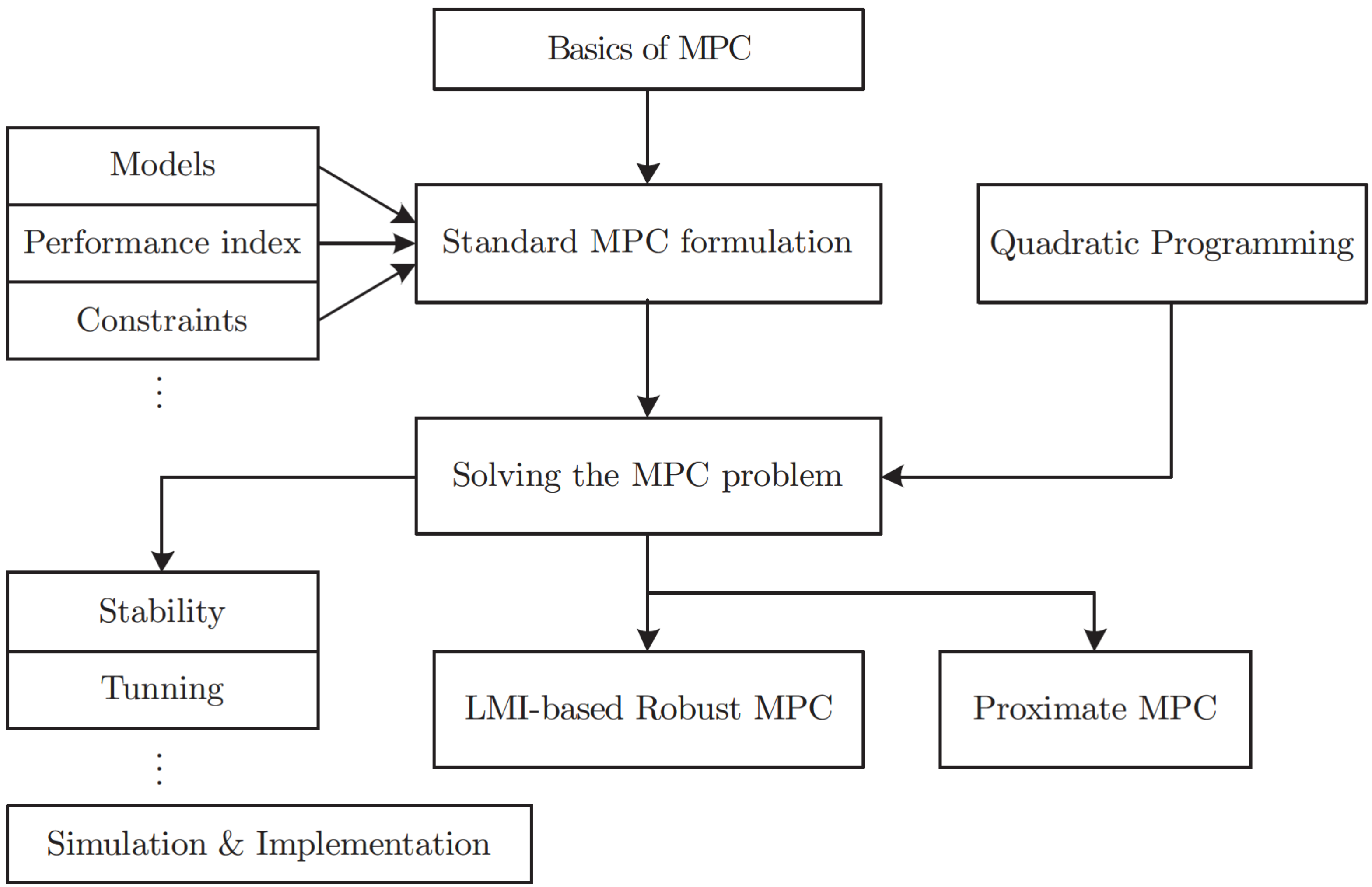
k+N번째까지의 상태변수와
입력을 고려하여 설계

$$J = \sum_{j=k}^{k+N} \underbrace{x^T(j|k)Qx(j|k)}_{\text{상태변수}} + \underbrace{u^T(j|k)Ru(j|k)}_{\text{입력}}$$

$$u_{min} \leq u \leq u_{max}$$
$$x \leq x_{max}$$
$$\vdots$$

최적 입력 U^*

$$\boldsymbol{U^* = u^*(k|k), u^*(k+1|k), u^*(k+2|k), \dots, u^*(k+N|k)}$$


Basics of MPC
Models
Performance index
Constraints
Standard MPC formulation
Quadratic Programming
Solving the MPC problem
Stability
Tunning
LMI-based Robust MPC
Proximate MPC
Simulation & Implementation

Introduction

Models are crucial in any engineering project

Do you want to:

Design complex systems?

Simulate and test your system early and often?

Analyze and validate your design?

Optimize system performance?

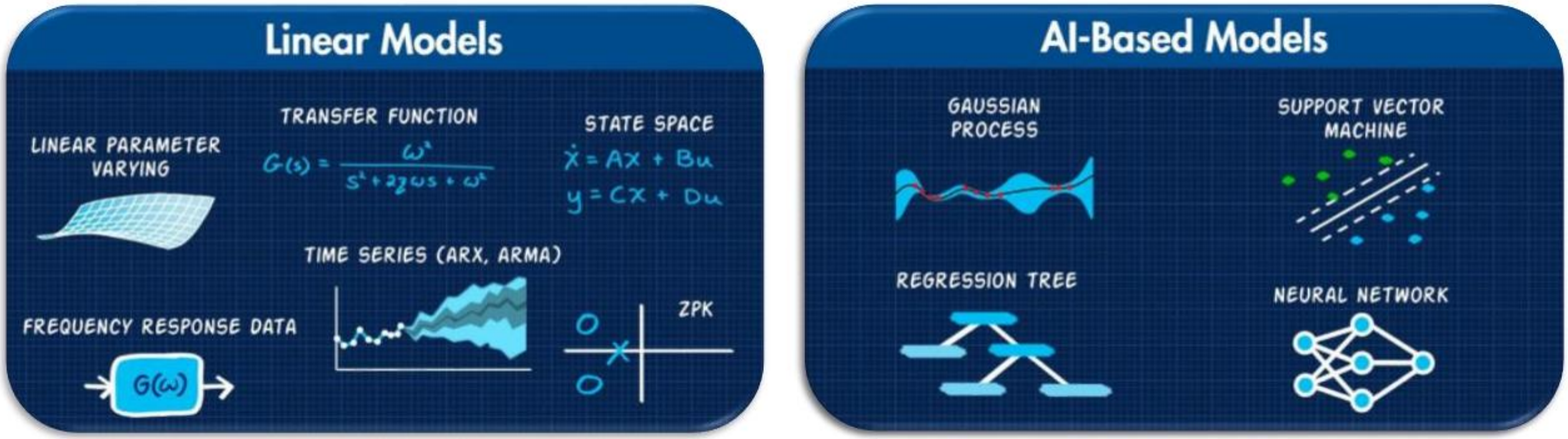
You need a model

And tools and methods for creating those models

Introduction

There are many different capabilities that can help you model dynamic systems

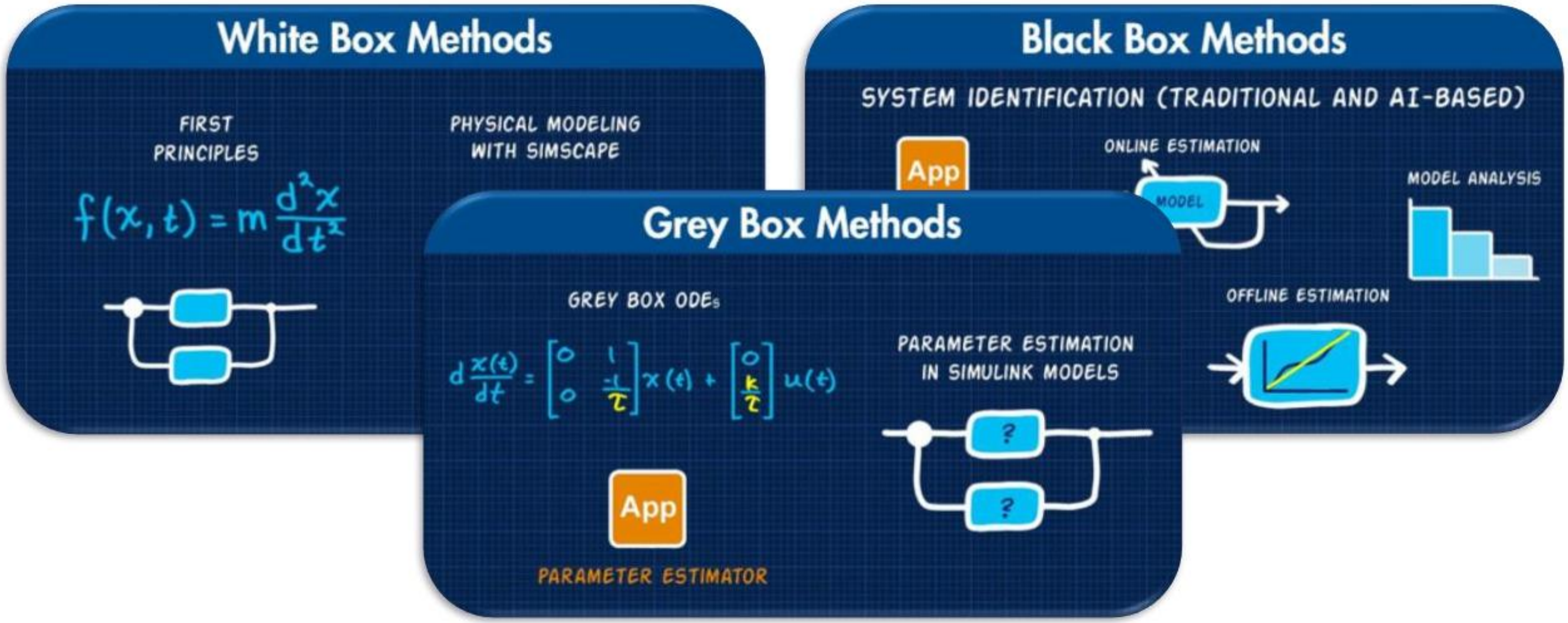
Model Structure



Introduction

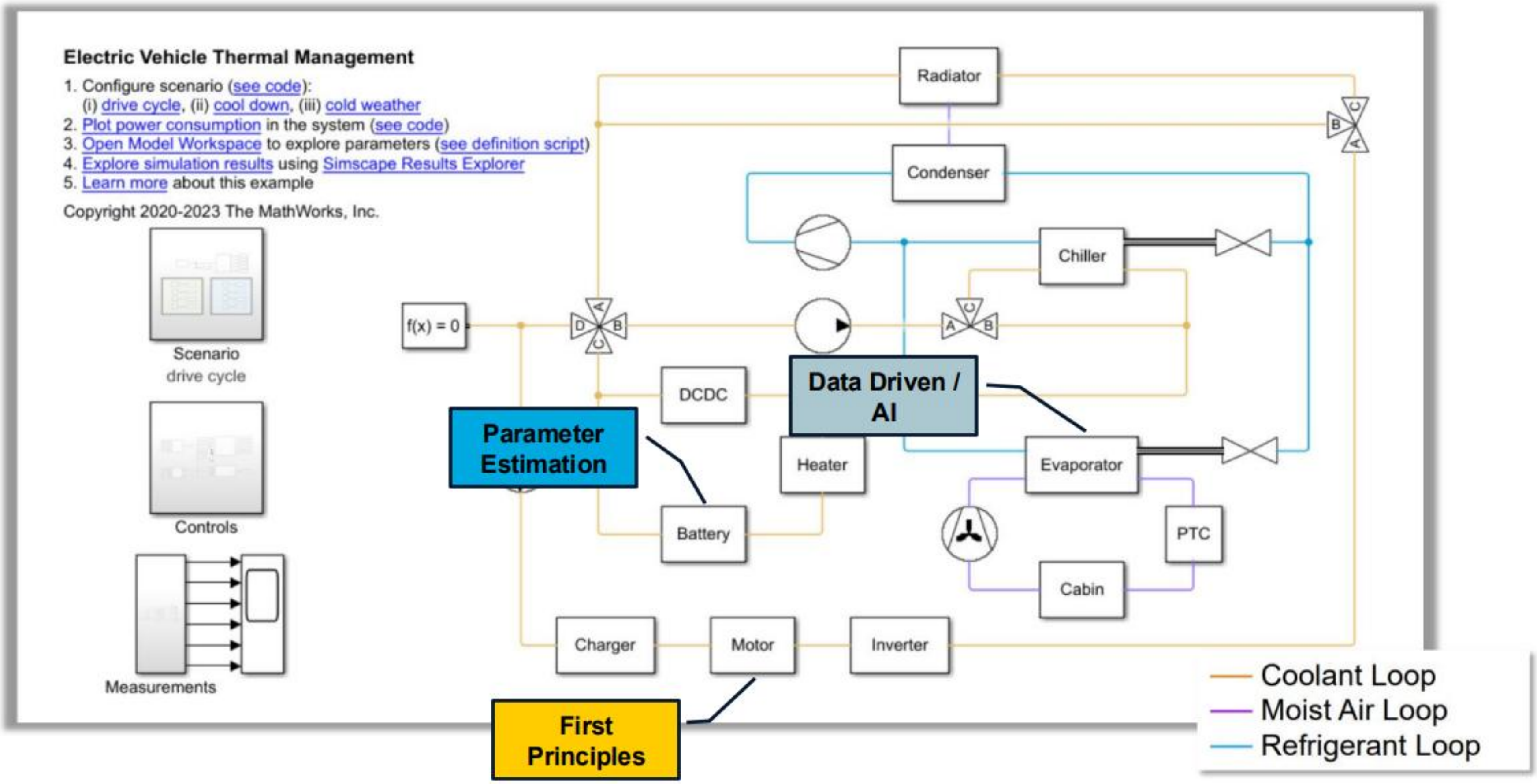
There are many different capabilities that can help you model dynamic systems

Model Parameter



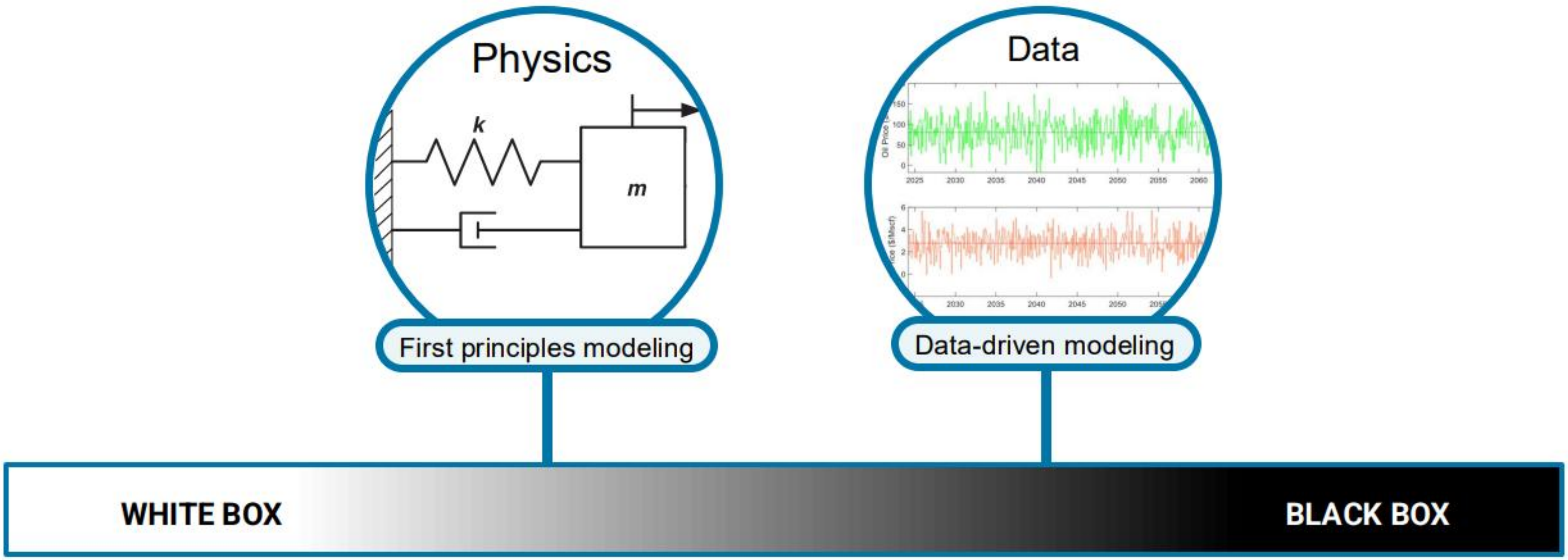
Introduction

Components could be modeled with **different methods**



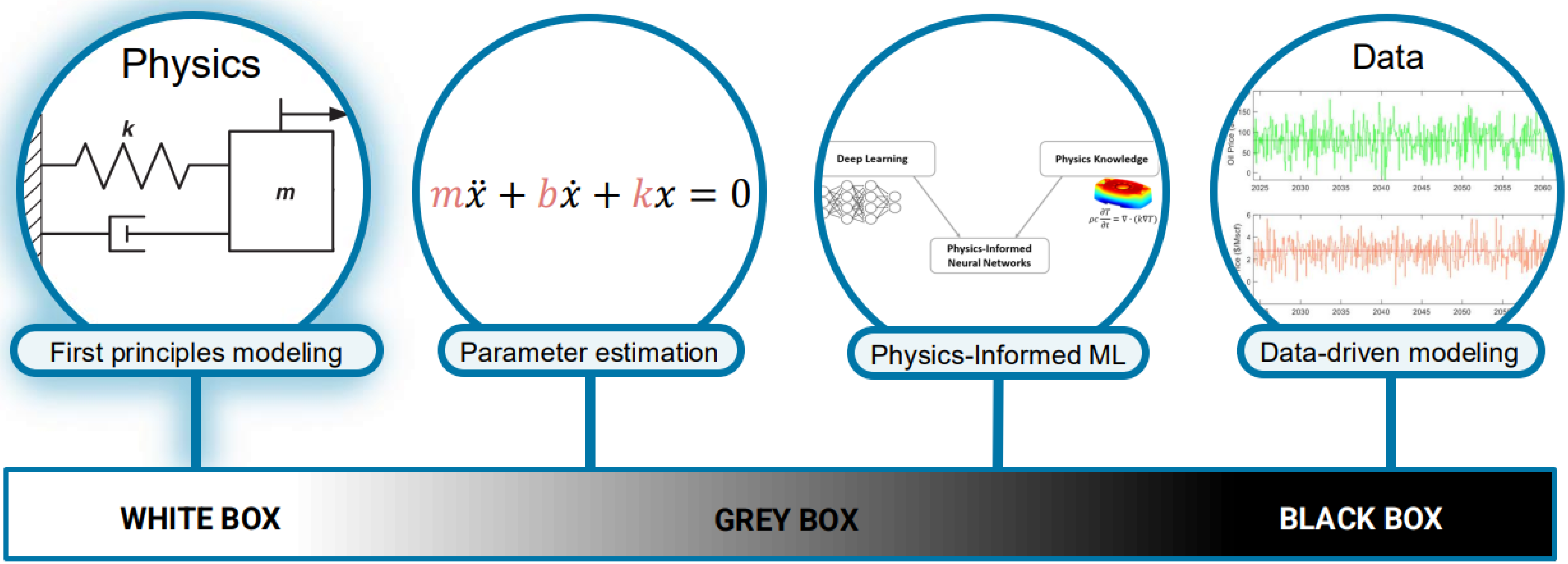
Introduction

Where do you get the information to create a model?



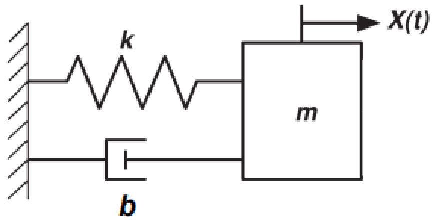
Introduction

Where do you get the information to create a model?



Introduction

With first principles modeling you build models that are based on physical laws



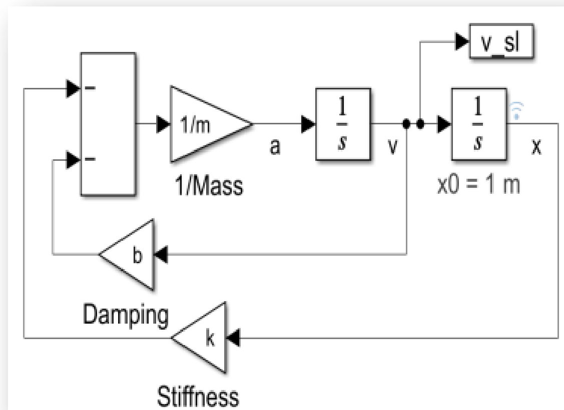
$$m\ddot{x} + b\dot{x} + kx = 0$$

Text-based code

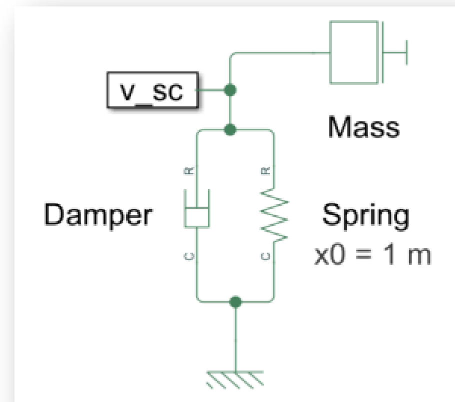
msd.m

```matlab
% Mass-Spring-Damper System Simulation

% Parameters
m = 1.0;  % Mass (kg)
b = 0.5;  % Damping coefficient (Ns/m)
k = 10.0; % Spring constant (N/m)

% Initial conditions
x0 = 1.0;  % Initial displacement (m)
v0 = 0.0;  % Initial velocity (m/s)
initial_conditions = [x0; v0];

% Time span for the simulation
t_span = [0 10]; % From 0 to 10 seconds

% Define the system of ODEs
% dx/dt = v
% dv/dt = -(b/m)*v - (k/m)*x
mass_spring_damper = @(t, y) [y(2); -(b/m)*y(2) - (k/m)*y(1)];

% Solve the ODE
[t, y] = ode45(mass_spring_damper, t_span, initial_conditions);
```

Executable block diagrams



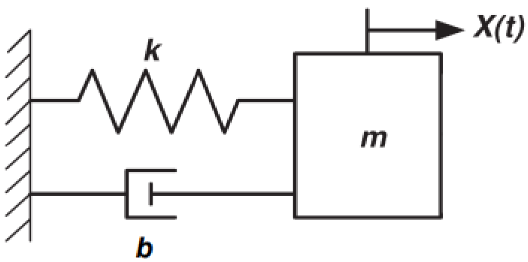
Physical modeling



Introduction

There are benefits to first principles modeling

Insight and interpretability



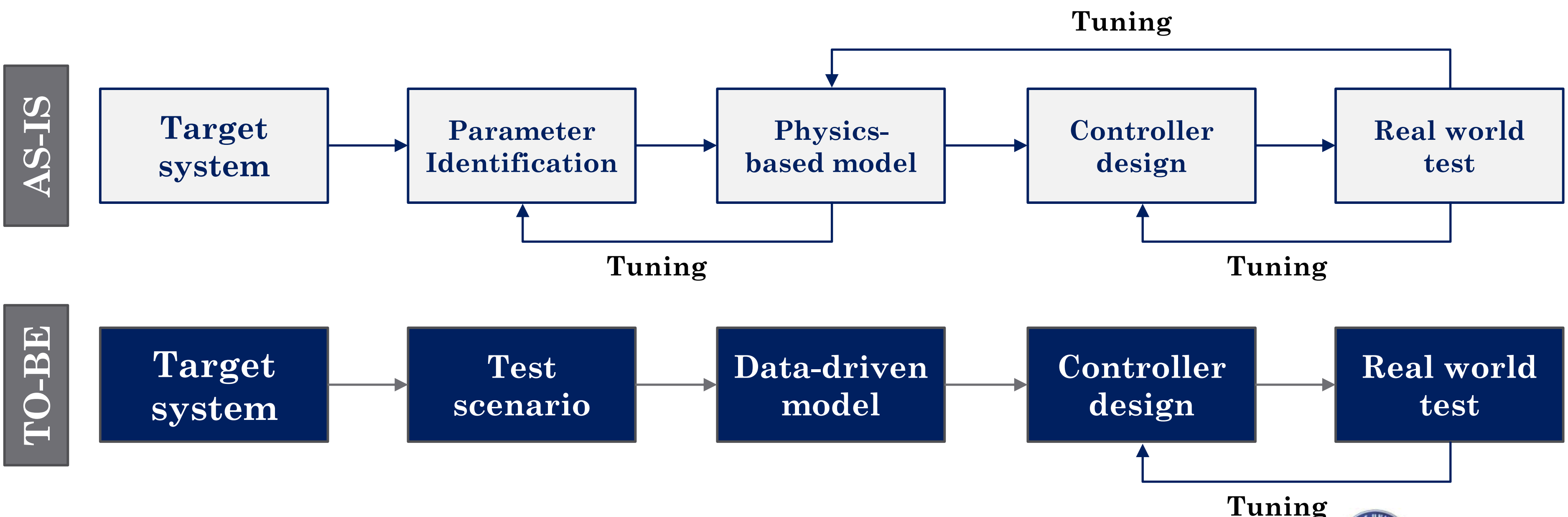
First principles model

$$\underbrace{m\ddot{x} + b\dot{x} + kx = 0}_{\text{This is understandable}}$$

Introduction

■Background

- 제어는 시스템의 수학적 혹은 물리적인 모델 기반으로 원하는 움직임을 가지도록 목표 하고 있음.
- 제어 성능을 개선시키기 위해서는 시스템의 움직임을 표현하는 **모델의 정확성**이 필요하나 대부분 현실세계의 모델은 비선형성과 불확실한 파라미터를 가지고 있기에 정확한 모델을 찾기 어려움
- 현재 이러한 문제점을 해결하기 위해 많은 시간 및 경제적 비용이 발생하며, 이러한 방법을 개선하기 위해 data-driven modeling 및 control 연구가 진행되고 있음



Introduction

■Background

- Shen과 Hong은 DMD를 활용해 legged robot의 동적 제어 성능을 향상시키는 새로운 MPC 프레임워크를 제시함
  - 계산 복잡성을 줄이기 위해 로봇의 동적 시스템을 DMD를 활용
  - 안정성 약 20% 향상 및 계산 비용 약 15% 감소

  ※ Shen, J., & Hong, D. (2021, May). A novel model predictive control framework using dynamic model decomposition applied to dynamic legged locomotion. In 2021 IEEE International Conference on Robotics and Automation (ICRA) (pp. 4926-4932). IEEE.

- Nonomura 와 Takaki는 데이터를 EKF DMD를 사용하여 보다 정밀한 시스템 식별 및 예측함
  - 기존 data driven 방식은 시스템의 noise와 data의 불확실성에 취약
  - EKF를 활용해 DMD에 적응형 필터링을 적용하여 모델 정확도 향상

  ※ Nonomura, T., Shibata, H., & Takaki, R. (2019). Extended-Kalman-filter-based dynamic mode decomposition for simultaneous system identification and denoising. PloS one, 14(2), e0209836.

- Goel 와 Bernstein는 비선형 매개변수 추정을 위해 새로운 데이터 기반 접근법인 RCMR(retrospective cost model refinement)을 제안함
  - RCMR을 통해 데이터 기반으로 매개변수 추정의 정확도 향상
  - 단일 매개변수 외 다중 매개변수의 비선형 의존성을 고려 가능

  ※ Goel, A., & Bernstein, D. S. (2018, December). Data-driven parameter estimation for models with nonlinear parameter dependence. In 2018 IEEE Conference on Decision and Control (CDC) (pp. 1470-1475). IEEE.

Introduction

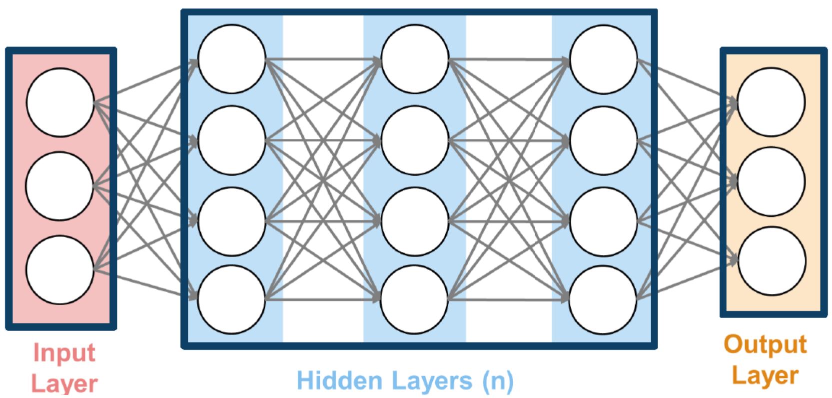
- **신경망 기반 모델링:**
  - MLP, RNN, LSTM, GRU, CNN, Transformer
- **비선형 시스템 모델링:**
  - DMD, Koopman Operator, SINDy, Reservoir Computing
- **생성 및 압축 모델링:**
  - Autoencoder, VAE, GAN
- **확률적 접근법:**
  - Gaussian Process (GP)
- **강화학습 기반 제어:**
  - 최적 제어 정책 학습

신경망 기반 모델링

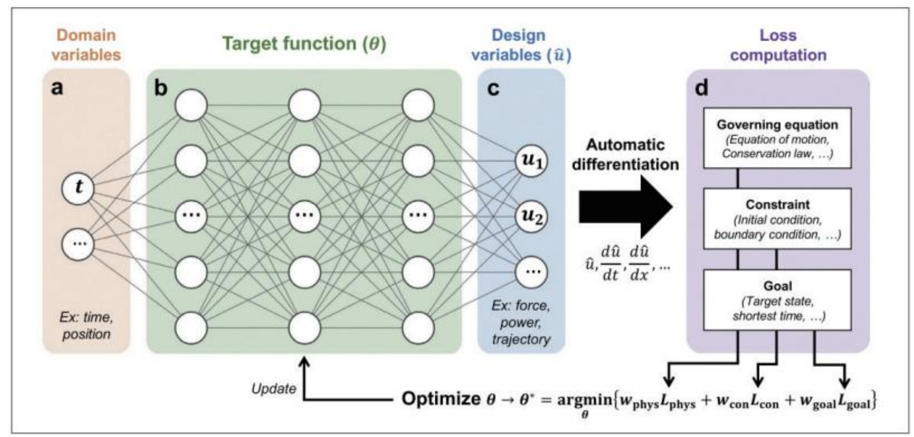
• Multi Layer Perception (MLP)

Alexander G. Parlos, Senior Member, IEEE, Kil T. Chong, and Amir F. Atiya, "**Application of the Recurrent Multilayer Perceptron in Modeling Complex Process Dynamics**", IEEE Transactions on Neural Networks

Long Jin , Longqi Liu , Xingxia Wang , Mingsheng Shang , and Fei-Yue Wang , "**Physical-Informed Neural Network for MPC-Based Trajectory Tracking of Vehicles With Noise Considered**", IEEE TRANSACTIONS ON INTELLIGENT VEHICLES



<MLP>



<Physics informed neural network(PINN)>

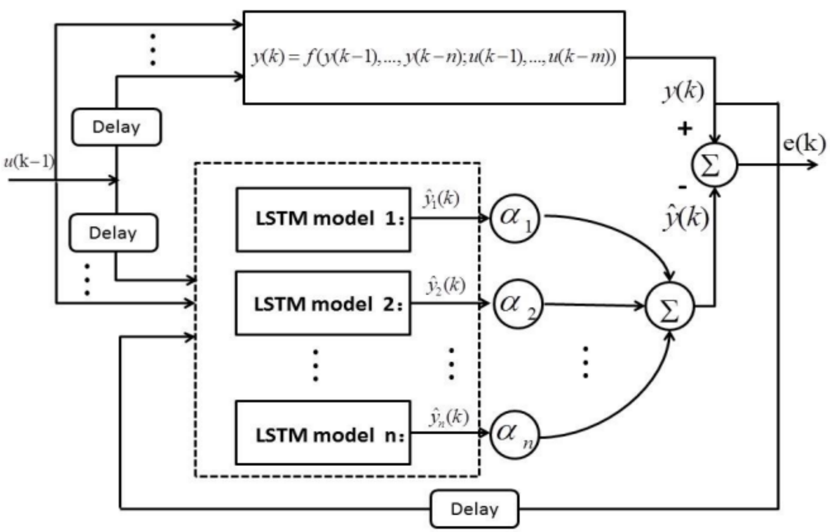
* PINN은 비선형 PDE로 표현되는 주어진 물리적 법칙을 준수하도록 훈련함

신경망 기반 모델링

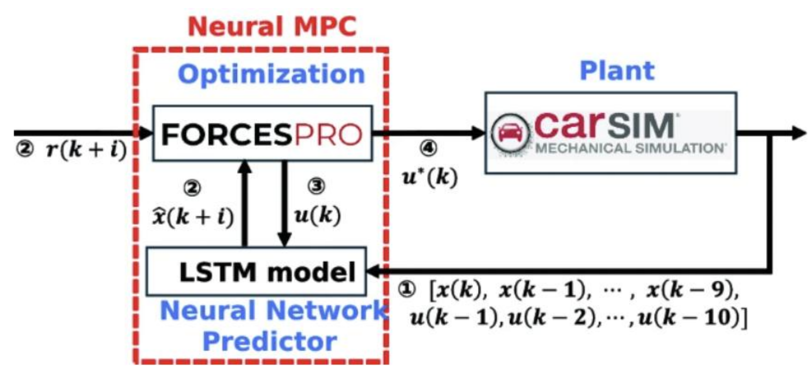
- **Recurrent Neural Network (RNN)**
- **Long-Short Term Memory (LSTM)**

Yu Wang, **"A New Concept using LSTM Neural Networks for Dynamic System Identification"**, IEEE American Control Conference

K.H. Kim, C. Jeong, J. Kim, S. Lee & C. M. Kang , **"Data-Driven LSTM Model and Predictive Control for Vehicle Lateral Motion",** Journal of Electrical Engineering & Technology



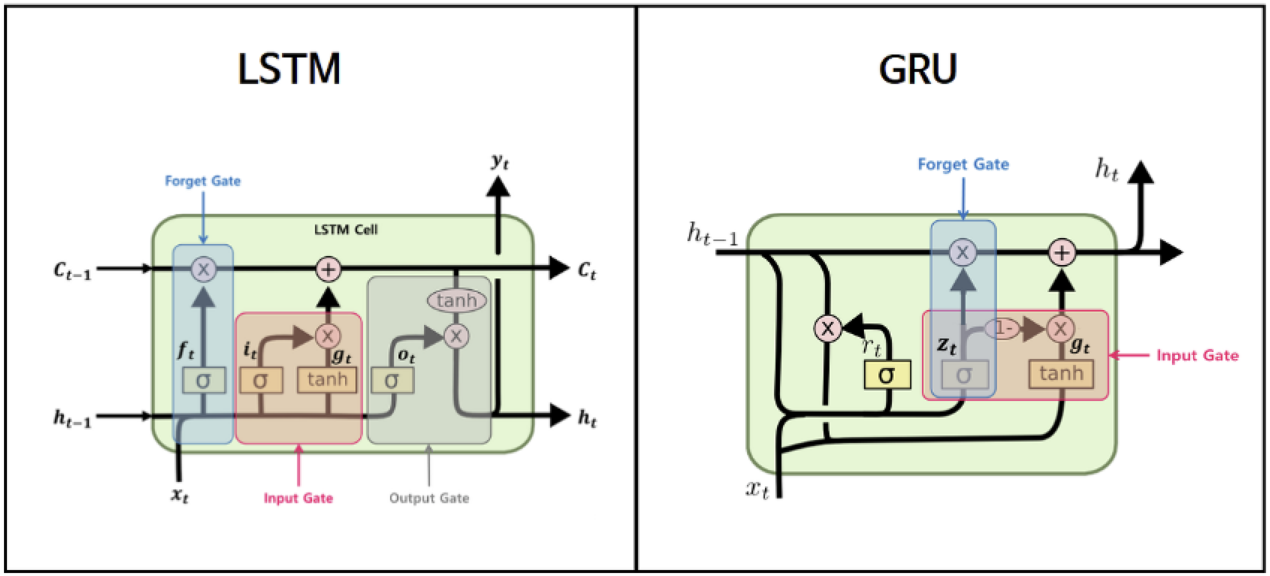
<1-step prediction model & Convex>



<N-step prediction model>

신경망 기반 모델링

• Gated Recurrent Unit (GRU), Bidirectional-GRU



LSTM	GRU
Forget Gate Input Gate Output Gate	Reset Gate Update Gate
Output gate를 통해서 cell state C_t의 일부분만 출력	Output gate가 없기 때문에 전체 cell state를 그대로 출력
Input gate와 output gate가 서로 독립	Input gate와 output gate가 통합되어 update gate를 통해 조절됨
More parameters	Fewer Parameters

- **Forget gate:** 버릴 정보를 결정.
- **Input gate:** 추가할 정보를 결정.
- **Output gate:** 출력할 정보를 결정.
- **Memory cell:** 장기 의존성을 유지.

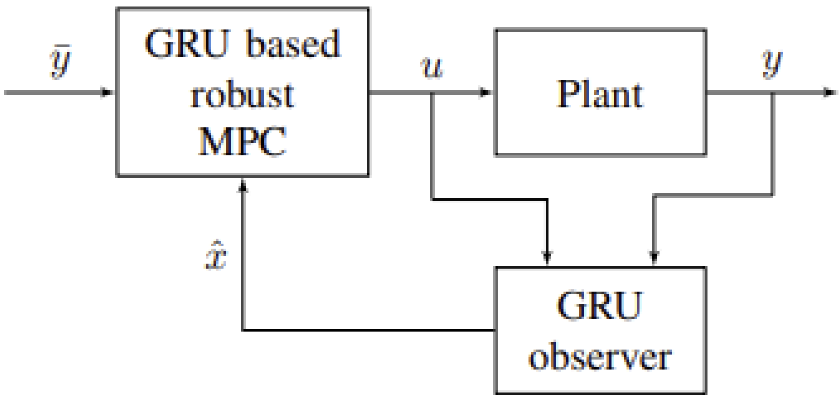
신경망 기반 모델링

• Gated Recurrent Unit (GRU), Bidirectional-GRU

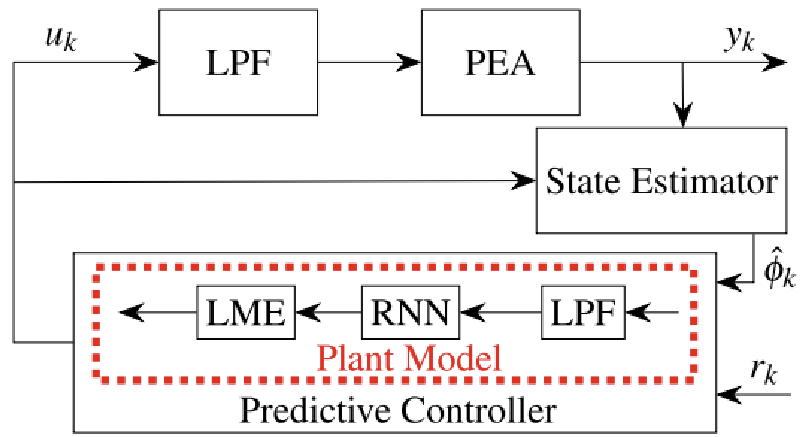
Irene Schimperna, Lalo Magni, **"Robust constrained nonlinear Model Predictive Control with Gated Recurrent Unit Model"**, Automatica

Yunpeng Pan, Jun Wang, "**Model Predictive Control of Unknown Nonlinear Dynamical Systems Based on Recurrent Neural Networks**" IEEE Transactions on Industrial Electronics

Shengwen Xie and Juan Ren, "**Recurrent-Neural-Network-Based Predictive Control of Piezo Actuators for Trajectory Tracking**", IEEE/ASME Transactions on Mechatronics



<GRU-based control>



<RNN-based control>

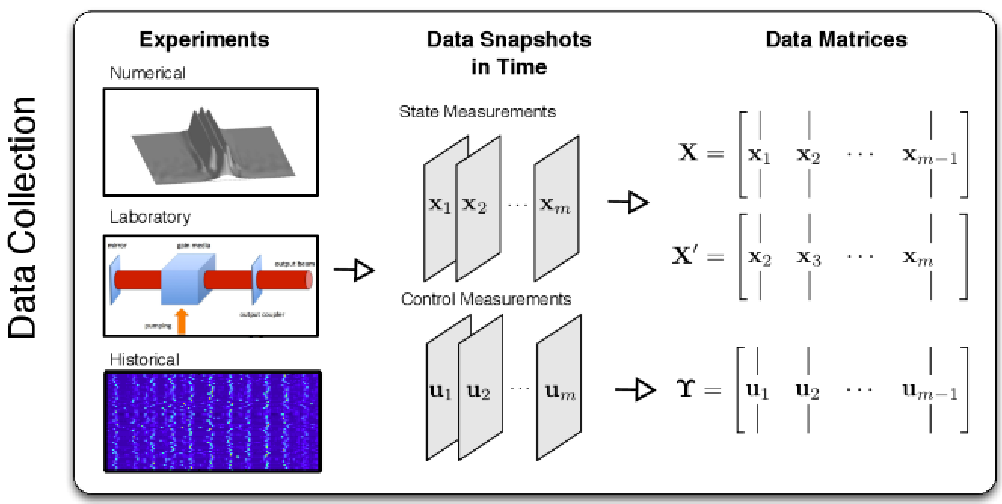
* LME: Linear model embedded / PEA: Piezo actuator

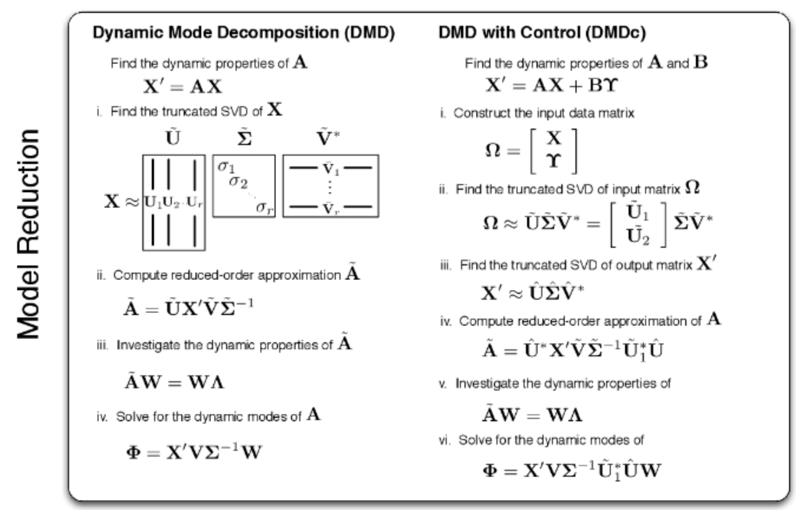
비선형 시스템 모델링

• Dynamic Mode Decomposition (DMD)

Joshua L, Steven L. Brunton, and J. Nathan Kutz, **"Dynamic Mode Decomposition with Control",** SIAM Journal on Applied Dynamical Systems

Guntae Kim et al, **"Vehicle's Lateral Motion Control Using Dynamic Mode Decomposition Model Predictive Control for Unknown Model",** International Journal of Automotive Technology





비선형 시스템 모델링

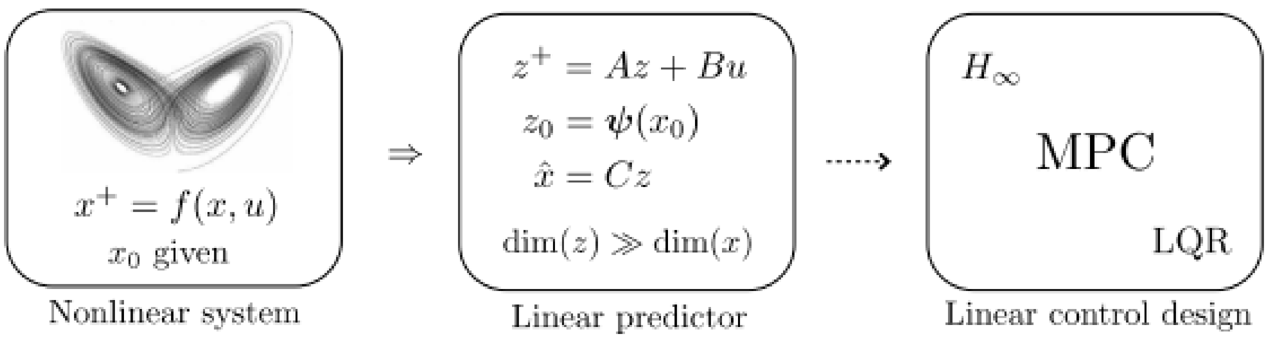
• Koopman Operator

Milan Korda, Igor Mezić, **“Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control”,** Automatica

Petar Bevanda, Stefan Sosnowski, Sandra Hirche, **“Koopman operator dynamical models: Learning, analysis and control”,** Annual Reviews in Control

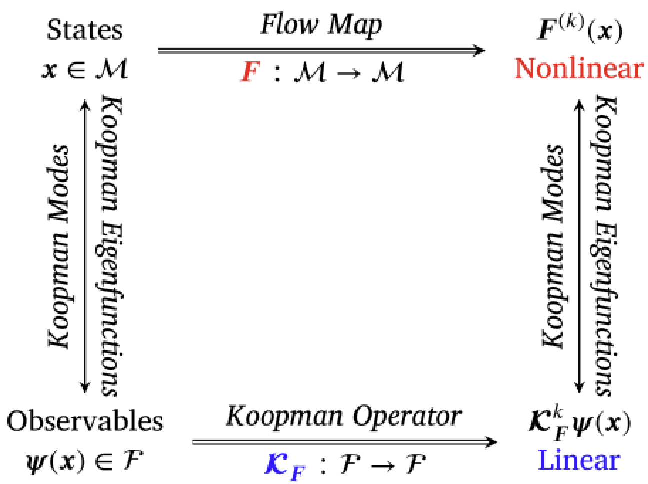
Vehicles, Yongqian Xiao et al, **"Deep Neural Networks With Koopman Operators for Modeling and Control of Autonomous”**, IEEE Transactions on Intelligent Vehicles

Carl Folkestad and Joel W. Burdick, **“Koopman NMPC: Koopman-based Learning and Nonlinear Model Predictive Control of Control-affine Systems”,** IEEE International Conference on Robotics and Automation



비선형 시스템 모델링

• Koopman Operator



Koopman 연산 과정:

- Encoder를 통해 상태 공간 x를 관측 공간 z로 변환:

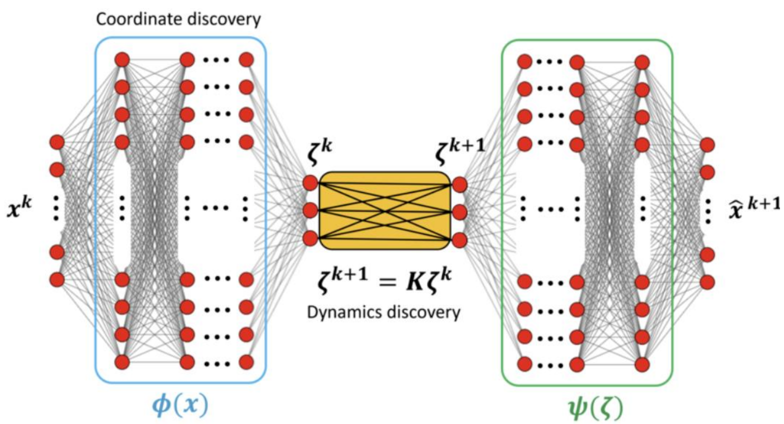
$$\zeta_t = \text{Encoder}(x_t)$$

- 관측 공간에서 Koopman 연산자 K를 적용:

$$\zeta_{t+1} = K\zeta_t$$

- Decoder를 통해 관측 공간의 표현 z_{t+1}를 상태공간으로 복원

$$\hat{x}_{t+1} = \text{Decoder}(\zeta_{t+1})$$



• 적절한 관측 함수 선택의 어려움

- 시스템의 동역학을 정확히 표현하려면 적합한 관측 함수를 설계해야 하나 도메인 지식이나 시행착오가 필요함

Deep Koopman operator

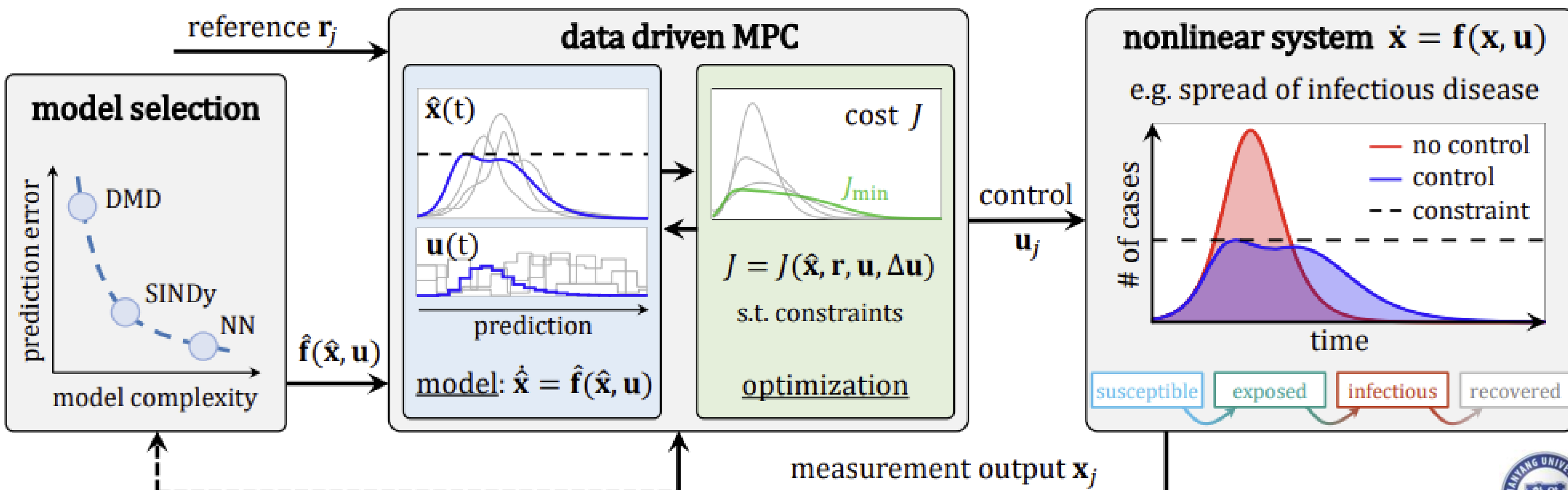
비선형 시스템 모델링

• Sparse identification of nonlinear dynamics (SINDy)

SL Brunton, JL Proctor, JN Kutz , **“Sparse identification of nonlinear dynamics with control (SINDYc)”** IFAC

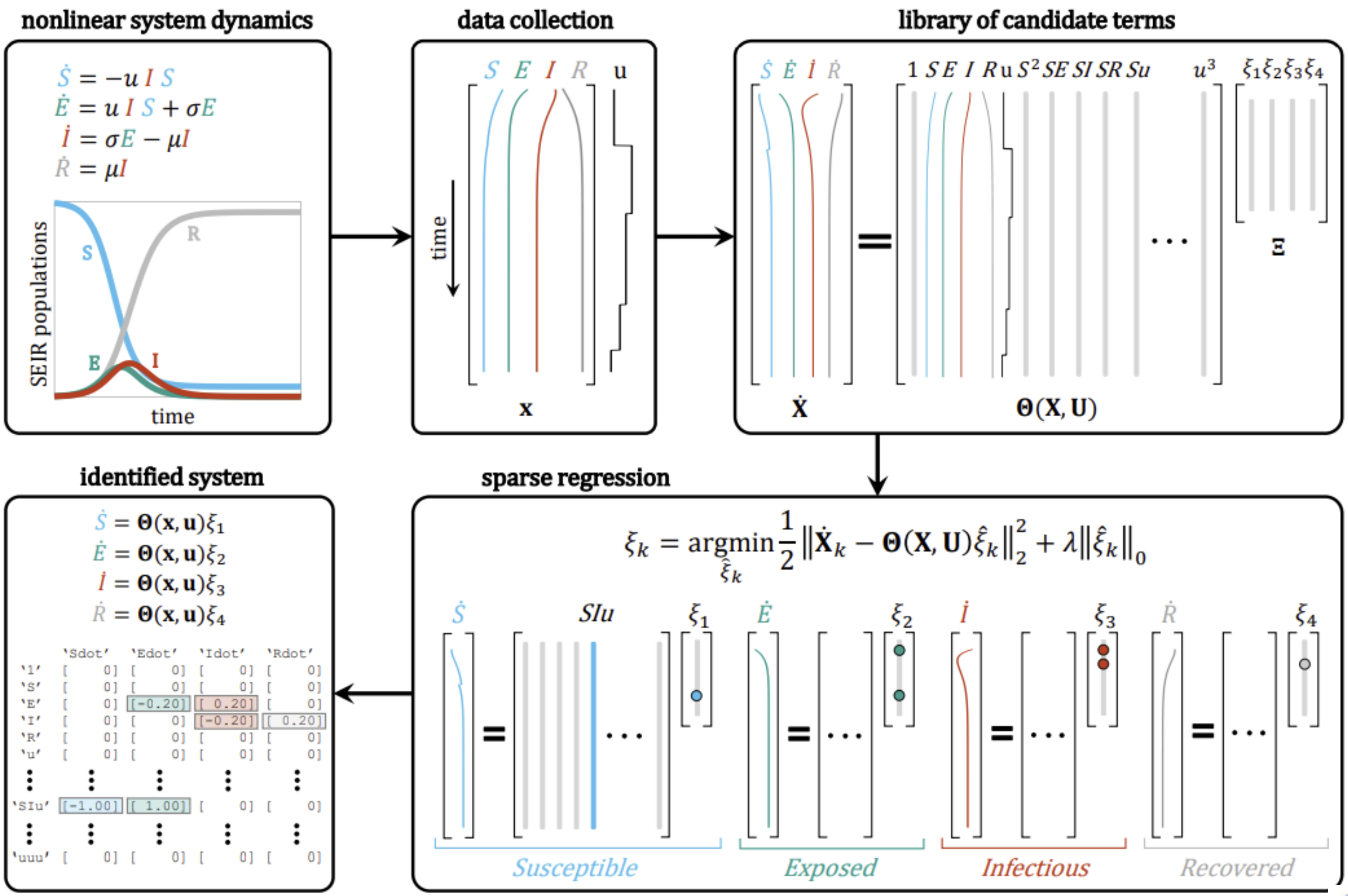
Urban Fasel et al, “**SINDy with Control: A Tutorial**”, IEEE Conference on Decision and Control (CDC)

Javad Khazaei, Ali Hosseinipour, "**Data-Driven Feedback Linearization Control of Distributed Energy Resources Using Sparse Regression**", IEEE Transactions on Smart Grid



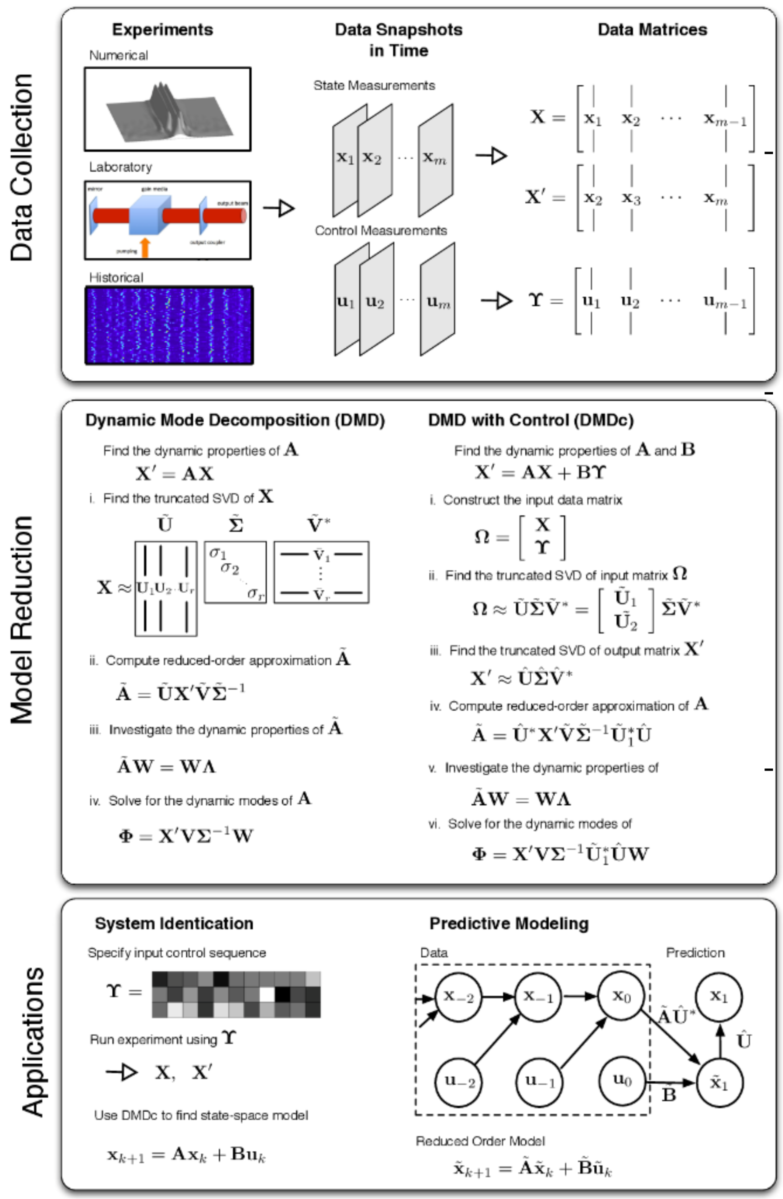
비선형 시스템 모델링

- **Sparse identification of nonlinear dynamics (SINDy)**



Dynamic mode decomposition & control

DMD & control



General system state-space form

$$X' = [A \quad B]\begin{bmatrix} X \\ \Gamma \end{bmatrix} = G\Omega$$

$$\begin{pmatrix} A \in \mathbb{R}^{n\times n}, B \in \mathbb{R}^{n\times m}, \\ G \in \mathbb{R}^{n\times(n+m)}, \Omega \in \mathbb{R}^{(n+m)\times k} \end{pmatrix}$$

Using SVD

$$\Omega = U\Sigma V^T \cong \tilde{U}\tilde{\Sigma}\tilde{V}^T = \tilde{\Omega}$$

$$X' = G\tilde{U}\tilde{\Sigma}\tilde{V}^T$$

$$G = X'\tilde{V}\tilde{\Sigma}^{-1}\tilde{U}^T = X'\tilde{V}\tilde{\Sigma}^{-1}[\tilde{U}_1^T \quad \tilde{U}_2^T] = [\hat{A} \quad \hat{B}]$$

$$\begin{pmatrix} U \in \mathbb{R}^{(n+m)\times(n+m)}, \Sigma \in \mathbb{R}^{(n+m)\times k}, V \in \mathbb{R}^{k\times k}, \\ \tilde{U} \in \mathbb{R}^{(n+m)\times p}, \tilde{\Sigma} \in \mathbb{R}^{p\times p}, \tilde{V} \in \mathbb{R}^{k\times p} \end{pmatrix}$$

Calculated system matrix

$$\hat{A} = X'\tilde{V}\tilde{\Sigma}^{-1}\tilde{U}_1^T$$

$$\hat{B} = X'\tilde{V}\tilde{\Sigma}^{-1}\tilde{U}_2^T$$

DMD & control

- ✓ 시스템의 동작을 예측하기 위해서는 시스템을 나타내는 **모델**이 필요
- ✓ **Dynamic Mode Decomposition(DMD)**은 시스템의 입출력 정보를 활용해 모델유도
  - 입출력 데이터에 기반한 수치적 해석
  - 이산시간 수치 데이터로부터 시스템의 모델을 식별

$$X_{k+1} = AX_k + BU_k$$
$$A \in \mathbb{R}^{n_x \times n_x}, B \in \mathbb{R}^{n_x \times n_u}$$
$$X_k \in \mathbb{R}^{n_x}, U_k \in \mathbb{R}^{n_u}$$

이산시간 state space model

데이터 정리

$$X' = AX + B\Gamma = [A \quad B]\begin{bmatrix} X \\ \Gamma \end{bmatrix} = G\Omega$$
where
$$X = [X_1 \quad X_2 \quad \cdots \quad X_{N-1}] \in \mathbb{R}^{n_x \times N-1}$$
$$X' = [X_2 \quad X_3 \quad \cdots \quad X_N] \in \mathbb{R}^{n_x \times N-1}$$
$$U = [U_1 \quad U_2 \quad \cdots \quad U_{N-1}] \in \mathbb{R}^{n_u \times N-1}$$

시스템의 특성 식별을 위한 각 step의 i/o data 수집

특이점 분해를 활용한 선형모델 유도

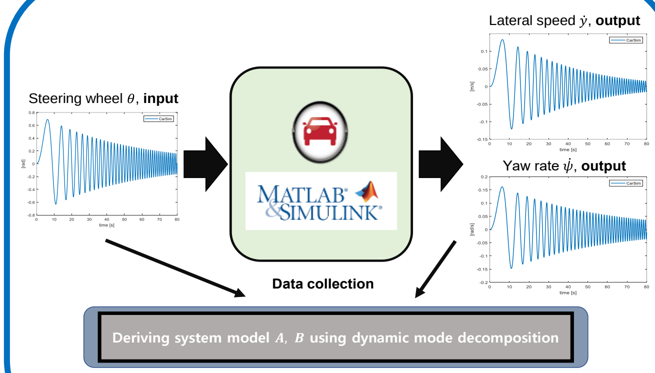
$$G\Omega = X'$$
$$SVD(\Omega) = U\Sigma V^T$$
$$G = X'\Sigma U^T = [\hat{A} \quad \hat{B}]$$
$$\hat{A} = X'V\Sigma^{-1}U_1^T$$
$$\hat{B} = X'V\Sigma^{-1}U_2^T$$

Linear Model by DMD

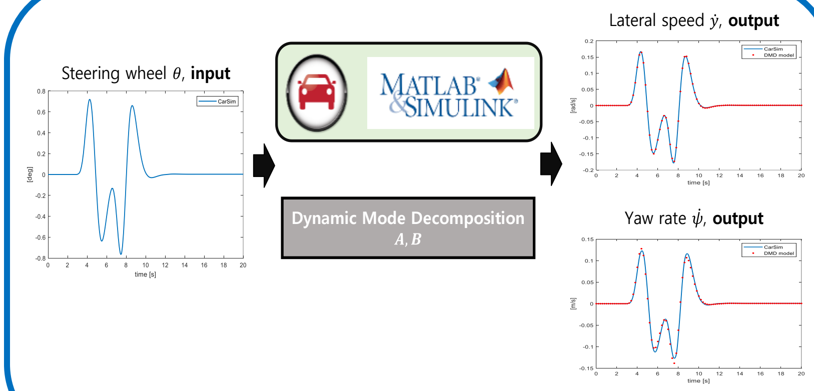
시스템 i/o data set을 SVD(Singular Value Decomposition)으로 분석하여 시스템 특성 유도

DMD & control

▶ Data-driven LKS



차량 data set 수집 및 DMD 를 통한 Modeling



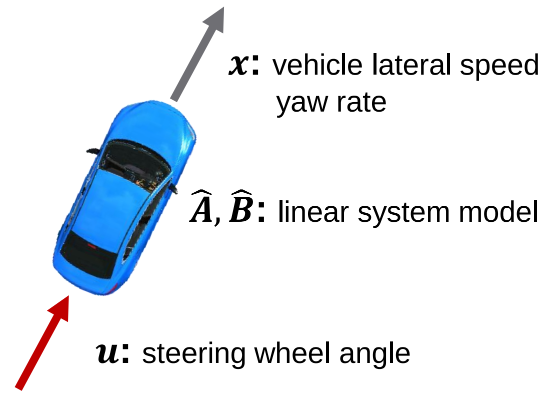
Double Lane change 시나리오를 통한 model 검증

1. Dynamic mode decomposition 기반의 모델링을 진행 하기 위해 차량 시뮬레이션 내에서 steering wheel angle 에 시간에 따라 감소하는 sine sweep 파형 입력하여 데이터 수집
   - Input : $\theta(steering\ wheel\ angle)$
   - state : $\dot{y}(lateral\ velocity), \dot{\psi}(heading\ rate)$
   - 시간 별로 데이터를 수집

2. 수집된 데이터를 사용하여 모델 유도
   - Dynamic mode decomposition을 활용한 모델 도출
   - 각 시간 데이터마다 모델을 도출

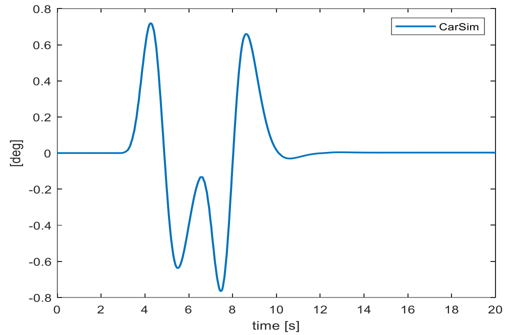
3. 도출한 모델을 주행 시나리오인 double lane change를 통해 검증

4. Fast Fourier Transform (FFT)를 통해 구간 별 동작특성 분석

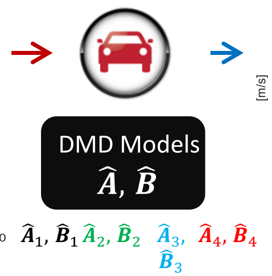


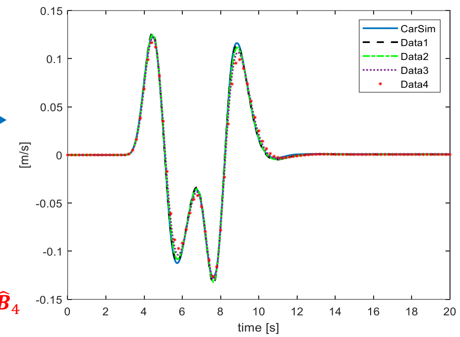
DMD & control

▶ Data-driven LKS

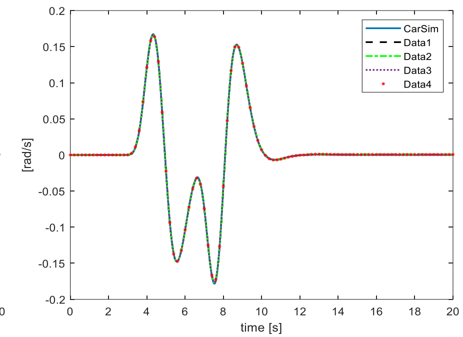


<차량 입력: Double lane change>





<차량 횡방향 속도>



<차량 헤딩 변화율>

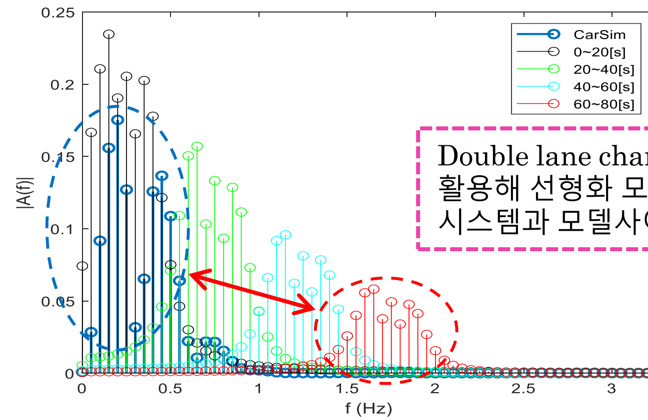
Lateral velocity RMS error

- Data1: $0\sim20$ $[s]$, 17×10^{-4}
- Data2: $0\sim40$ $[s]$, 24×10^{-4}
- Data3: $0\sim60$ $[s]$, 50×10^{-4}
- Data4: $0\sim80$ $[s]$, 83×10^{-4}

육안으로 오차가 확인되는 시뮬레이션 차량의 횡 방향 움직임과 DMD로 유도된 4개 모델의 횡방향 움직임 비교

⇨ 역동적인 데이터를 다수 포함하여 유도된 모델일수록 오차 발생

⇨ FFT를 활용해 double lane change와 sweep sine 입력데이터의 구간별 동작특성을 분석



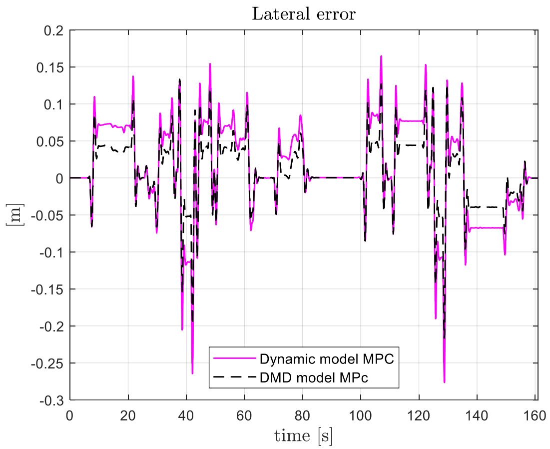
Double lane change와 먼 특성의 데이터를 활용해 선형화 모델을 유도할수록 시스템과 모델사이의 오차가 확인됨

<Double lane change차량 입력과 sweep sine 입력 구간별 동작특성 FFT로 분석>

✓ 모델이 시스템과 유사할 때 모델기반제어의 유효성을 확보할 수 있음

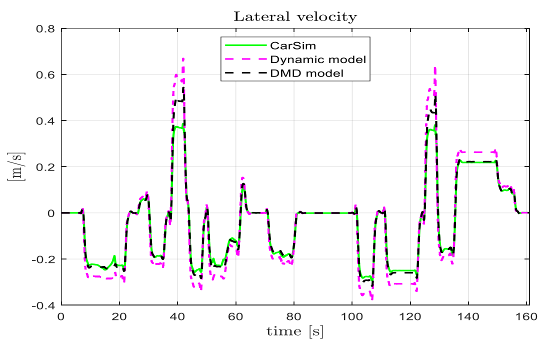
DMD & control

▶ Data-driven LKS

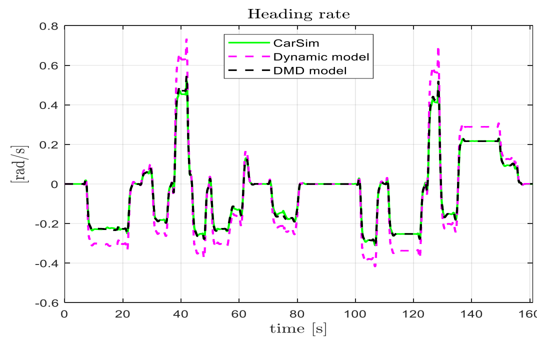


<횡방향 오차 비교>

- 동일한 실험의 조건을 가지기 위해 제어기의 파라미터는 동일하게 설정
- 모델의 유효함을 평가하기 위해 차량 횡방향 제어에 널리 사용되는 차량 다이나믹 모델 사용
- Dynamic mode decomposition을 사용한 모델이 차량 다이나믹 모델 보다 작은 횡방향 오차를 보이는 구간이 많은 것을 확인 가능



<횡 방향 속도 비교>



< 헤딩 변화율 속도 비교>

DMD & Neural network

LSTM & control