Model Predictive Control from theory to practice

Xiaodong Cheng, Assistant Professor, (xiaodong.cheng@wur.nl)

Mathematical and statistical methods group, Wageningen University

Course materials

WAGENINGEN
UNIVERSITY & RESEARCH

Outline

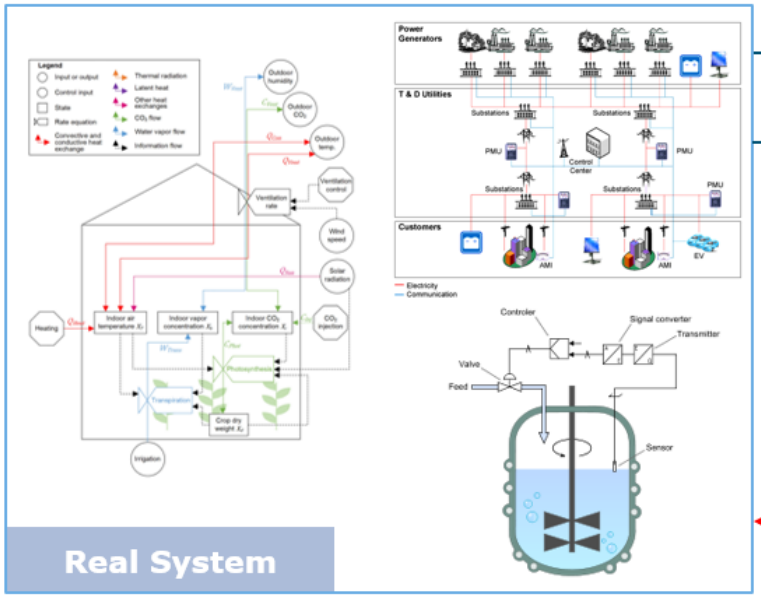
- Preliminaries on Systems and Control
- **Deterministic** MPC
- MPC with **Uncertainty** (Stochastic MPC / Robust MPC)
- Other variations of MPC

90h

- Appling nonlinear MPC to greenhouse temperature control

45m

Diagram of Systems and Control



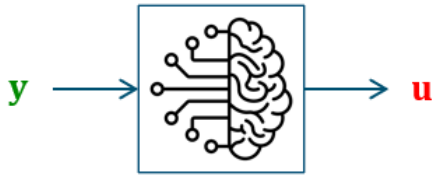
Domain Knowledge

Measured Data

$$\frac{d}{dt}\mathbf{x} = \mathbf{f}(\mathbf{x}, t, \mathbf{u}, \mathbf{d};\ \boldsymbol{\beta})$$
$$\mathbf{y} = \mathbf{g}(\mathbf{x}, t) + \mathbf{n}$$

System Model

Decision/Control

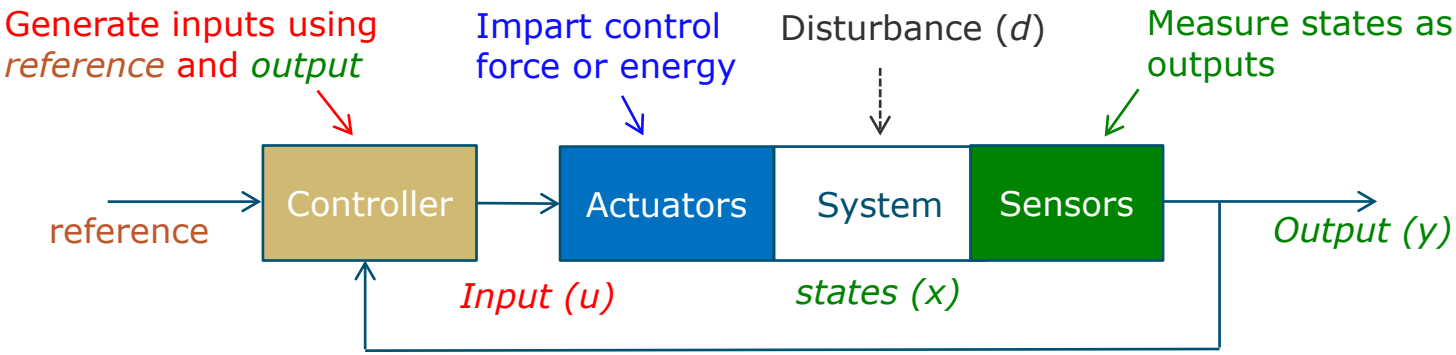


Optimalization/AI

Research Topics:

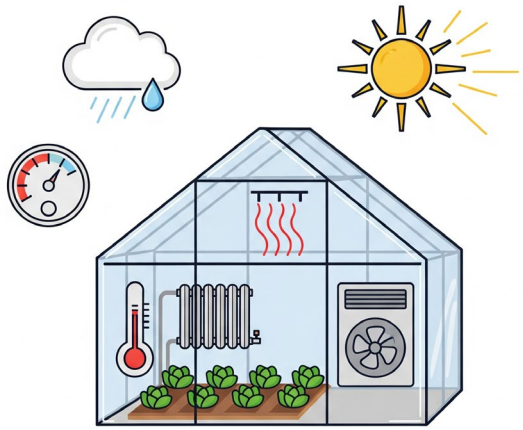
- Model identification: build mathematical input-output predictive models of dynamical systems
- Control optimization: sensor/actuator allocation, state estimation (soft sensing), design control signals

Control Systems

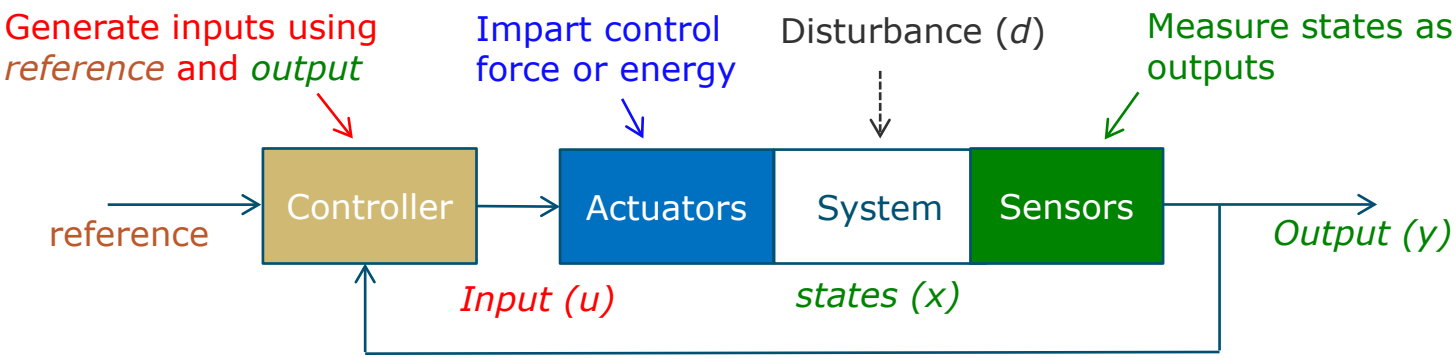


In a simple greenhouse:

- ***u**: heating, ventilation*
- ***y = x**: indoor temperature, indoor humidity*
- ***d**: radiation, outdoor temperature, outdoor humidity*



Control Systems



State-space model (continuous-time)

dynamics
(vector field)

disturbance

$$\frac{d}{dt}\mathbf{x} = \mathbf{f}(\mathbf{x}, \mathbf{u}, \mathbf{d}; \mathbf{p}) \qquad \mathbf{y} = \mathbf{g}(\mathbf{x}) + \mathbf{n}$$

states

inputs
(control)

parameters

noise

measurements

Terminology and notations

Systems and control	Sequential decision making
State-space model (deterministic/stochastic)	Transition function (stochastic)
State x (often continuous)	State: S (discontinuous)
Output y (often continuous)	Observation: O (discontinuous)
Input u (continuous/discontinuous)	Decision x (discontinuous)
Objective/cost function V	Objective/cost function C
Controller (control strategy)	Policy (strategy)

Key theories in control:

- *Stability (Lyapunov), Optimality (Bellman, Pontryagin)*
- *Controllability (valid controller)*
- *Observability (building state observer, Kalman filter)*

State-Space Models

continuous-time model

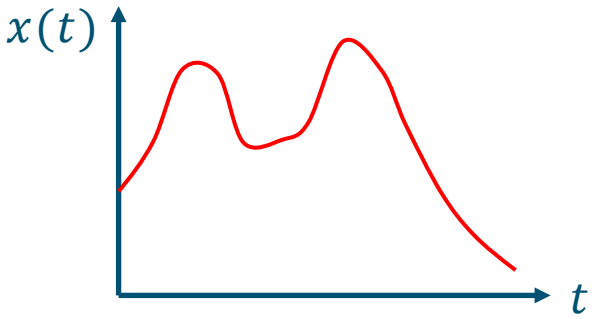
$$\frac{d}{dt}\mathbf{x} = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), \mathbf{d}(t); \mathbf{p})$$

$$\mathbf{y}(t) = \mathbf{g}(\mathbf{x}(t)) + \mathbf{n}(t)$$

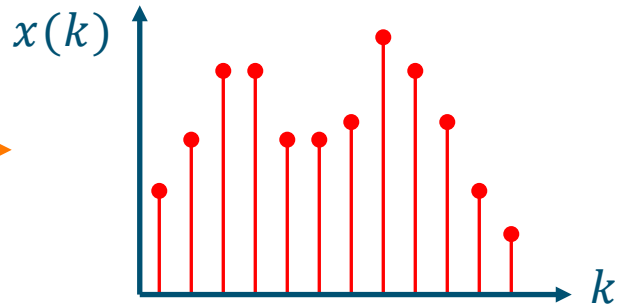
discrete-time model

$$\mathbf{x}(k+1) = \mathbf{f}(\mathbf{x}(k), \mathbf{u}(k), \mathbf{d}(k); \mathbf{p})$$

$$\mathbf{y}(k) = \mathbf{g}(\mathbf{x}(k)) + \mathbf{n}(k)$$

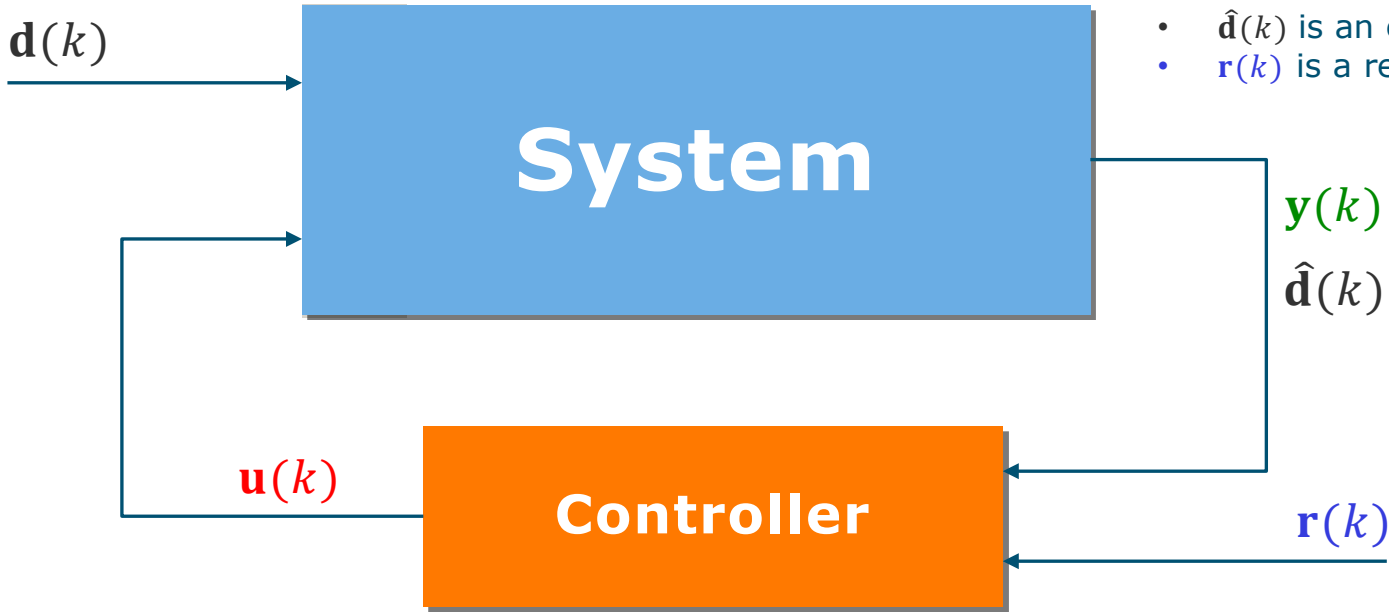


Sampling period



Most modern control systems are implemented on digital computers or embedded processors, which operate in discrete time

Problem Statement



- $\hat{\mathbf{d}}(k)$ is an estimation of $\mathbf{d}(k)$
- $\mathbf{r}(k)$ is a reference (setpoint)

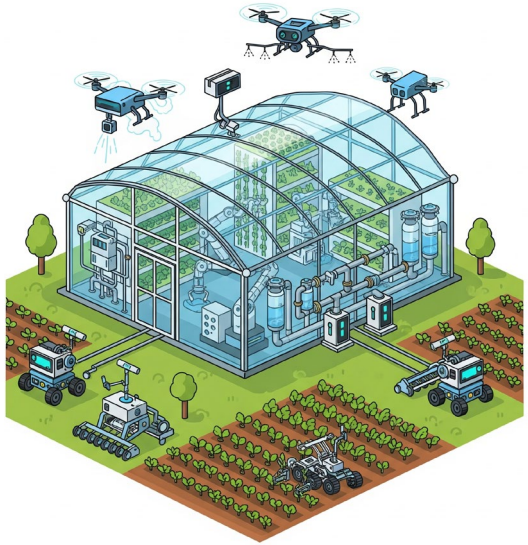
Problem: Given state space model of the system, and signals $\mathbf{y}(k)$, $\mathbf{r}(k)$, $\hat{\mathbf{d}}(k)$, compute the control input $\mathbf{u}(k)$

Problem Statement

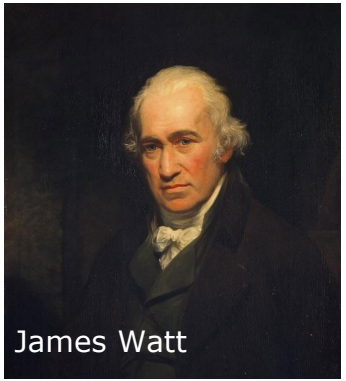
Problem: Given state space model of the system, and signals $\mathbf{y}(k)$, $\mathbf{r}(k)$, $\hat{\mathbf{d}}(k)$, compute the control input $\mathbf{u}(k)$

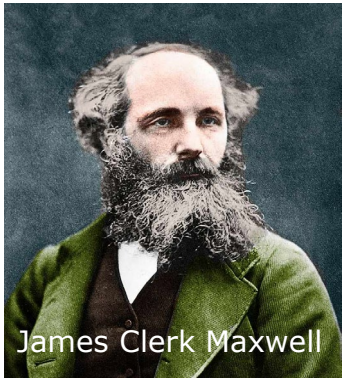
Objectives in agriculture:

- maximize crop yield
- maximize milk production
- tracking of a reference signal (setpoint)
- minimize reaching time
- ensure a bandwidth/maximum overshoot
- ensure well-being animal
- minimize pesticide
- noise/vibration reduction
- ....



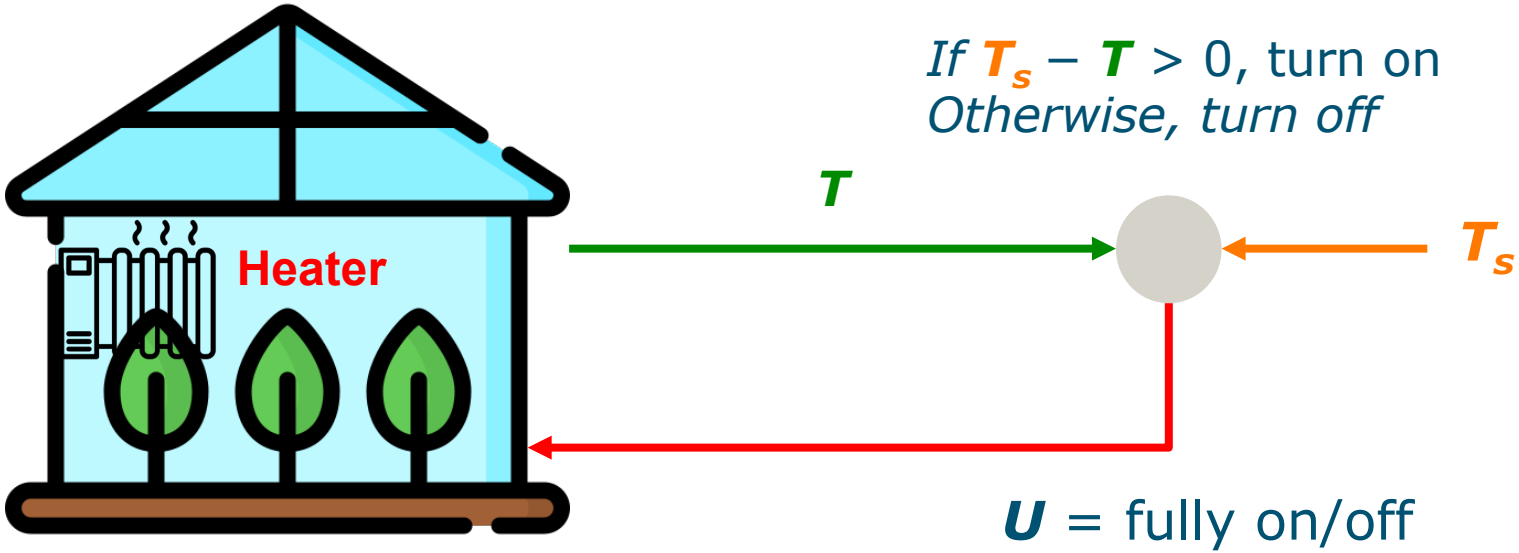
Starting Point of Control Theory


James Watt

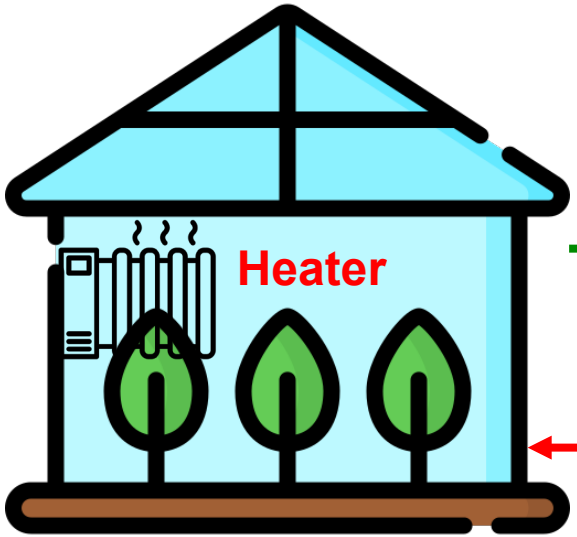

James Clerk Maxwell

- Steam regulator (Watt, 1788):
  Controls the speed of running engines by changing steam feed pressure
- Theoretical foundations **J.C. Maxwell**:
  His paper "On governors" (1868) mathematically describes the behavior of Watt regulator
  —thereby establishing the theoretical basis of control engineering

Principal of feedback control



Principal of feedback control



$e = T - T_s$

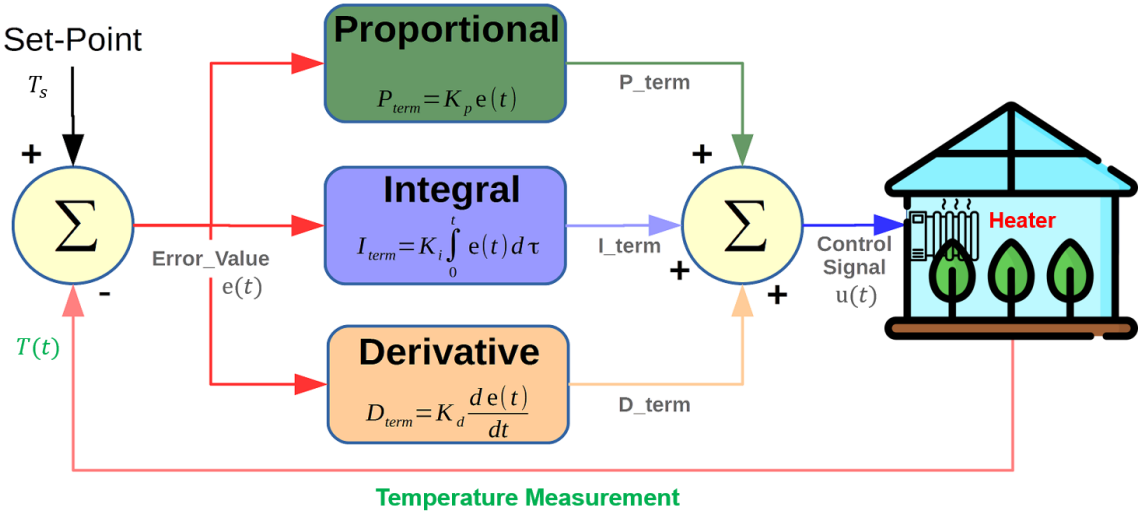
T

T_s

$U = \max\{ K_p * e, 0 \}$

P control!

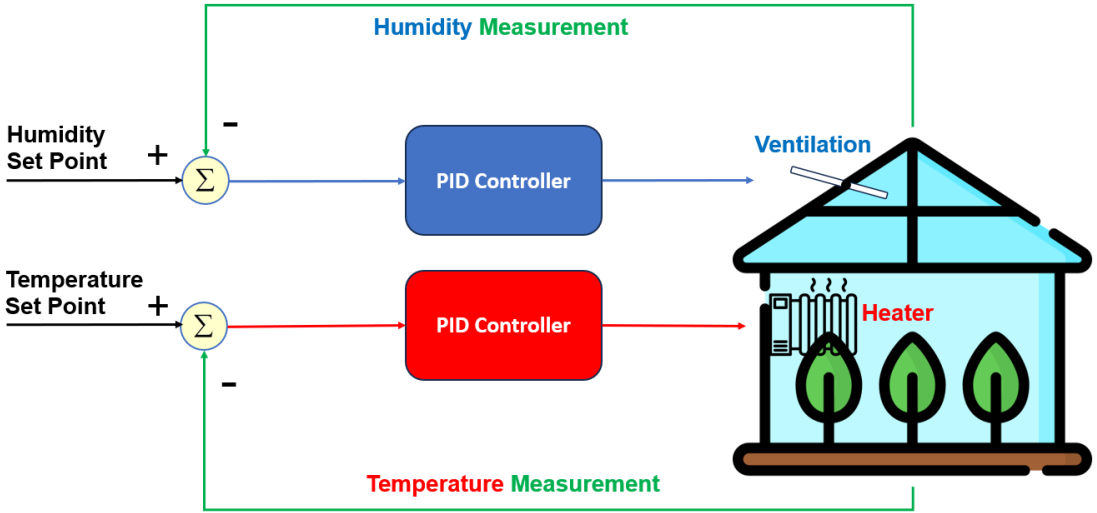
PID control - Classical control method



PID temperature control in greenhouse

- The variable we want to control (temperature) must be measured
- Set point is required to produce the error
- Mathematical model of the system (heater and greenhouse environment) describing its behavior is not required

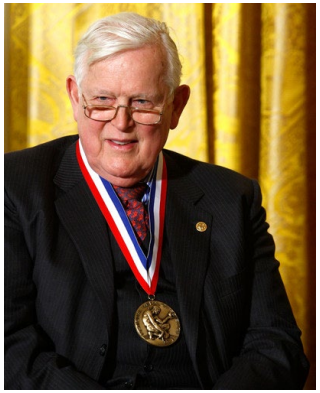
PID control - Classical control method

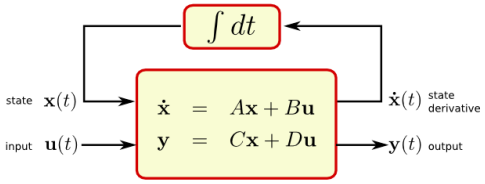


PID control of multiple variables (multi-loop PID control) and its complications

- humidity and temperature are not independent variables
- Coupling between the control loops: ventilation affects both humidity and temperature
- Design one control loop will also disturb the other control loop

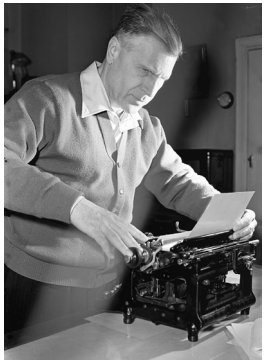
Modern Control Theory





Rudolf E. Kálmán (1930 –2016)

Richard E. Bellman
(1920 – 1984)



Lev Semenovich Pontryagin
(1908 – 1988)

Model-based control
Time domain analysis

- Kalman:
  controllability and observability, Kalman filter…
- Bellman, Pontryagin
  Foundation of optimal control, continuous-time MPC

Controller Design Methods

Model-Based	Model-Free
Robust Control	PID Tuning
Adaptive Control	Extremum Seeking
Optimal Control (LQR)	Neural Network Control
Model Predictive Control	Reinforcement Learning

Key Characters:

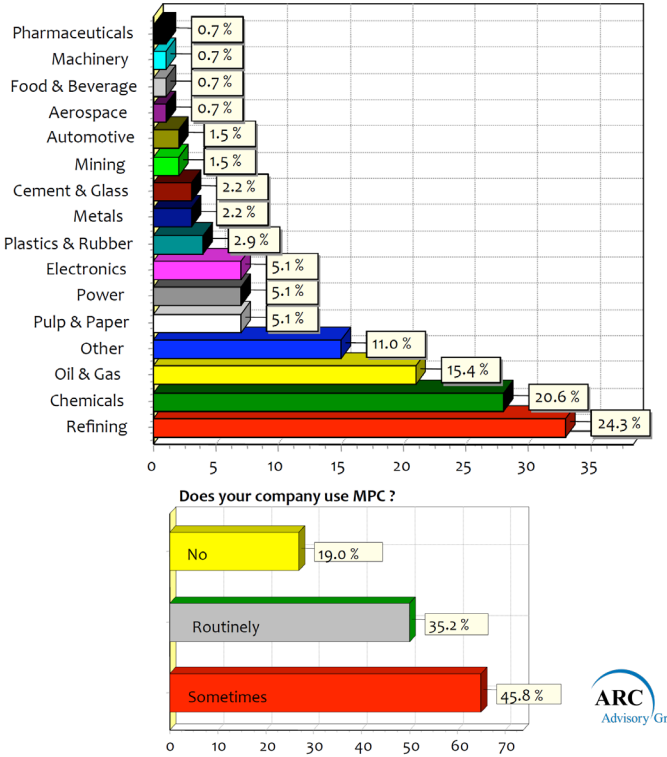
- Model-Based: reasonably accurate model, high transparency, reliable
- Model-Free: data-driven, unknown dynamics, adapt in real time

Hybrid Approaches: Best of Both Worlds:

- Adaptive MPC: Online update of model using data
- Model-Based RL: Uses a learned model to improve sample efficiency
- Neural-Augmented Control: Embeds NNs inside conventional controllers to estimate unknown dynamics

MPC History

- **Origins**: 1979 **Dynamic Matrix Control** (DMC) by **Shell** Oil Company (Motivation: *multivariable*, *constrained*)
- **Core Idea**: **predict future outcomes** using a **mathematical model** of a process to **optimize** control over a **finite horizon** (*Theory lagging behind 10-20 years*)
- Also known as **Receding Horizon Control (RHC)**
- **Impact**: MPC revolutionized industrial control by allowing systems to handle **multivariable processes** and **constraints** effectively



2005 Survey about US industries

Emerging MPC Applications

Many industrial applications: autonomous driving, robotics, energy systems…

Honeywell | INDUSTRIAL AUTOMATION

Industries Products Services Solutions Support News & Events

Products > Control & Supervisory Systems > Quality Control Systems (QCS) > Experion® MX > Control Applications > Machine Direction (MD)

CONTROL APPLICATIONS

Machine Direction (MD)

Honeywell's Experion MX Machine Direction Controls utilize industry proven multivariable model predictive control technology to improve product quality and paper machine efficiency, while minimizing material and energy usage.

Overview

CONTACT US SUPPORT

Overview

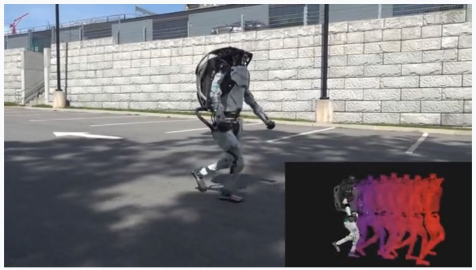
The foundation of the Experion MX machine direction controls is Honeywell's unique Alpha MPC controller. Alpha MPC has been designed to control the most difficult of paper making processes, with the least amount of configuration and maintenance effort. It minimizes the future predicted variation of all controlled variables (CVs) in the papermaking process while honoring both their quality constraints and any physical constraints on manipulated variables (MVs). For processes with extra degrees of freedom (more MVs than CVs), Alpha MPC can be configured to optimize any CVs or MVs.

Experion MX MD control is highly scalable from very basic weight and moisture control up to including total energy- and chemical ion optimization.

Boston Dynamics

Products Solutions Industries Atlas & Innovation Company Resources

At the heart of Atlas's controller is a technique called Model Predictive Control (MPC). Our model is a description of how the robot's actions will affect its state, and we use that model to predict how the robot's state will evolve over a short period of time. To control the robot, we use optimization: given the robot's measured state, MPC searches over possible actions it can take now and in the near future to best achieve the set task.

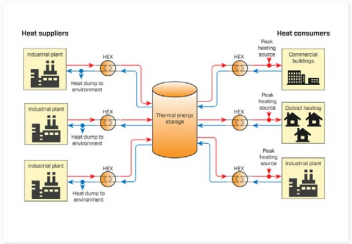


Atlas walking using model predictive control. The robot constantly updates its prediction of its future state, shown in the bottom right, and it uses that prediction to choose its actions in real time.

HighEFF

Home About Research and Results Education Laboratory facilities Events Outreach Final report

Robust Model Predictive Control in Energy Systems: Dealing with Uncertainty



Safe Vehicle Motion Control:
COMBINED LONGITUDINAL AND LATERAL PATH TRACKING

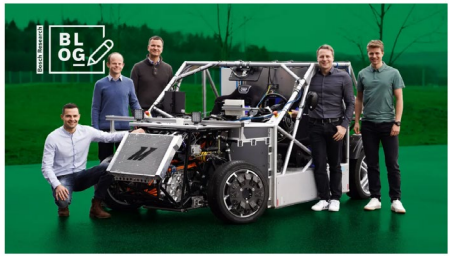
Model Predictive Control (MPC) for autonomous path tracking up to limits of handling

BOSCH

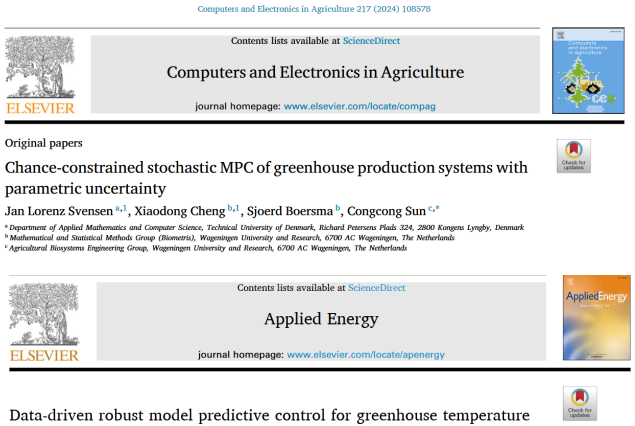
Products and services Company Careers

Make motion control functions even safer

BLOG



Emerging MPC Applications in Agriculture


Computers and Electronics in Agriculture 217 (2024) 108578

Contents lists available at ScienceDirect

Computers and Electronics in Agriculture

journal homepage: www.elsevier.com/locate/compag

Original papers

Chance-constrained stochastic MPC of greenhouse production systems with parametric uncertainty

Jan Lorenz Svensen [a,1], Xiaodong Cheng [b,1], Sjoerd Boersma [b], Congcong Sun [c,*]

[a] Department of Applied Mathematics and Computer Science, Technical University of Denmark, Richard Petersens Plads 324, 2800 Kongens Lyngby, Denmark
[b] Mathematical and Statistical Methods Group (Biometris), Wageningen University and Research, 6700 AC Wageningen, The Netherlands
[c] Agricultural Biosystems Engineering Group, Wageningen University and Research, 6700 AC Wageningen, The Netherlands

Contents lists available at ScienceDirect

Applied Energy

journal homepage: www.elsevier.com/locate/apenergy

Data-driven robust model predictive control for greenhouse temperature control and energy utilisation assessment

Farhat Mahmood, Rajesh Govindan, Amine Bermak, David Yang, Tareq Al-Ansari *

College of Science and Engineering, Hamad Bin Khalifa University, Qatar Foundation, Doha, Qatar

Greenhouse


Contents lists available at ScienceDirect

Smart Agricultural Technology

journal homepage: www.journals.elsevier.com/smart-agricultural-technology

Data-driven model predictive control for precision irrigation management

Erion Bwambale [a,b,c,*], Felix K. Abagale [a,b], Geophrey K. Anornu [d]

[a] West African Center for Water, Irrigation and Sustainable Agriculture (WACWISA), University for Development Studies, P. O. Box TL 1882, Tamale, Ghana
[b] Department of Agricultural Engineering, University for Development Studies, P. O. Box TL 1882, Tamale, Ghana
[c] Department of Agricultural and Biosystems Engineering, Makerere University, P. O. Box 7062, Kampala, Uganda
[d] Department of Civil Engineering, Regional Water and Environmental Sanitation Center Kumasi (RWESCK), Kwame Nkrumah University of Sciences and Technology, Kumasi, Ghana

Irrigation

Livestock


Proceedings of the European Control Conference 2007
Kos, Greece, July 2-5, 2007

Model Predictive Control of Thermal Comfort and Indoor Air Quality in Livestock Stable

Zhuang Wu and Murali R. Rajamani and James B. Rawlings and Jakob Stoustrup


ilable online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 52-30 (2019) 56–61

IFAC Papers Online
CONFERENCE PAPER ARCHIVE

Optimal irrigation management for large-scale arable farming using model predictive control *

L.P.A. Schoonen * A.T.J.R. Cobbenhagen * W.P.M.H. Heemels *

* Dept. of Mechanical Engineering, Eindhoven University of Technology, The Netherlands (e-mail: a.t.j.r.cobbenhagen@tue.nl)

Aquaculture


Contents lists available at ScienceDirect

Journal of Process Control

journal homepage: www.elsevier.com/locate/jprocont

Model predictive control paradigms for fish growth reference tracking in precision aquaculture ☆

Abderrazak Chahid [a], Ibrahima N'Doye [a], John E. Majoris [b], Michael L. Berumen [b], Taous Meriem Laleg-Kirati [a,*]

[a] Computer, Electrical and Mathematical Sciences and Engineering Division (CEMSE), King Abdullah University of Science and Technology (KAUST), Thuwal 23955-6900, Saudi Arabia
[b] Red Sea Research Center, Biological and Environmental Science and Engineering Division, King Abdullah University of Science and Technology (KAUST), Thuwal 23955-6900, Saudi Arabia

Poultry


Contents lists available at ScienceDirect

Smart Agricultural Technology

journal homepage: www.journals.elsevier.com/smart-agricultural-technology

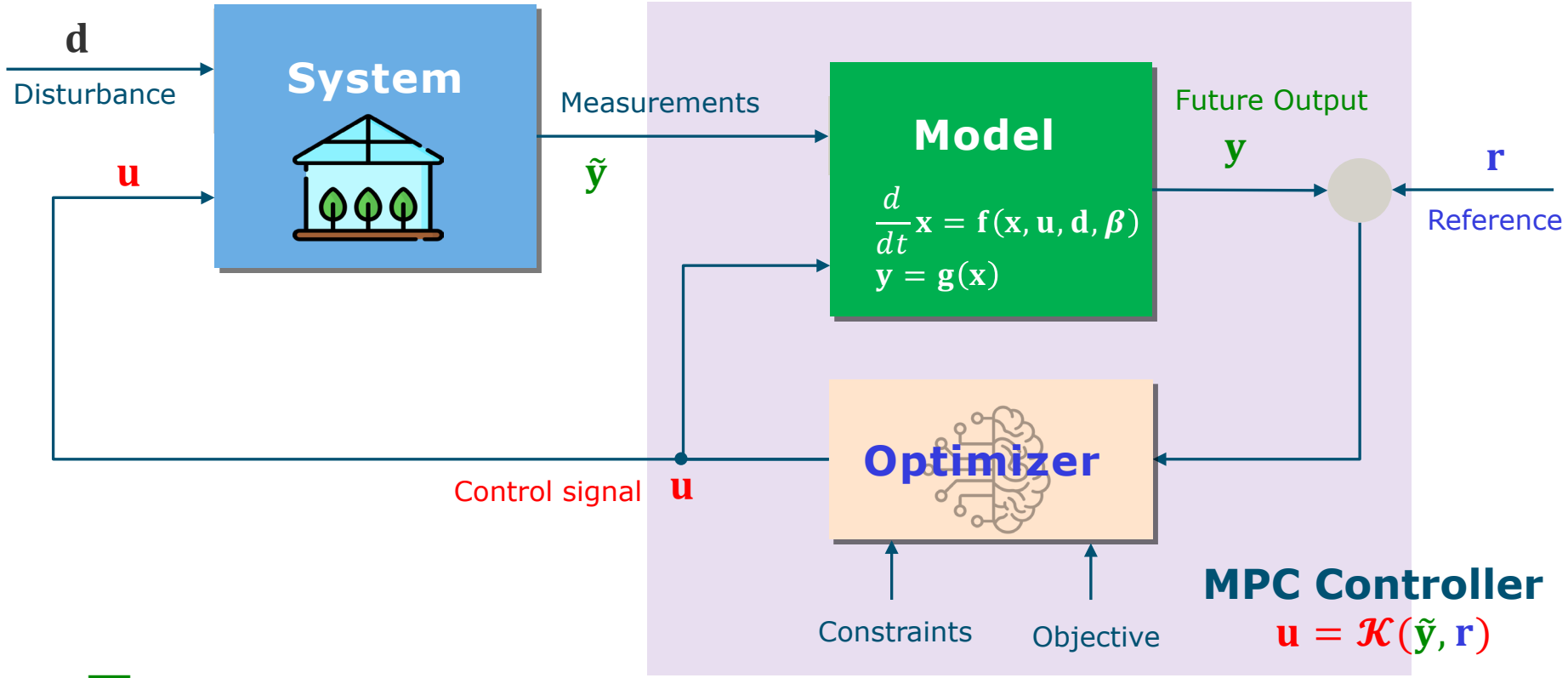
A predictive control strategy for optimal operation of a chicken coop integrated microgrid

Yasmine Achour [a,*], Ahmed Ouammi [b], Driss Zejli [a]

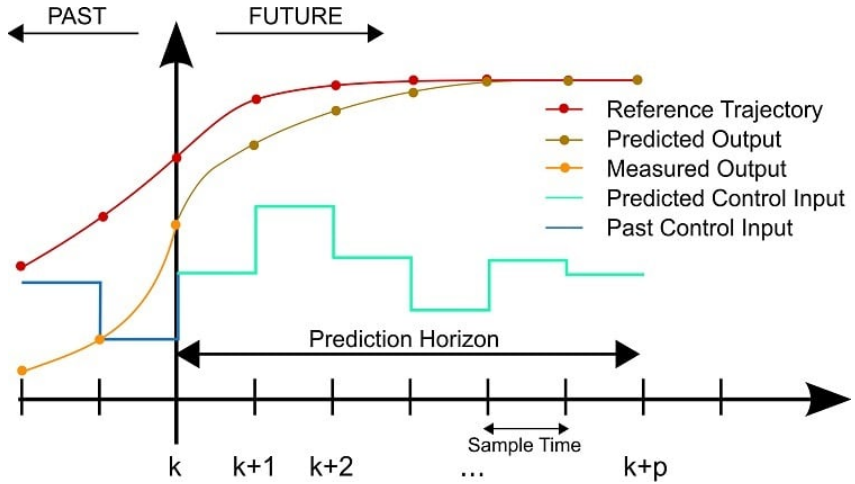
[a] Advanced Systems Engineering Laboratory, National School of Advanced Sciences, Kenitra, Morocco
[b] Electrical Engineering Department, ETS, Montreal

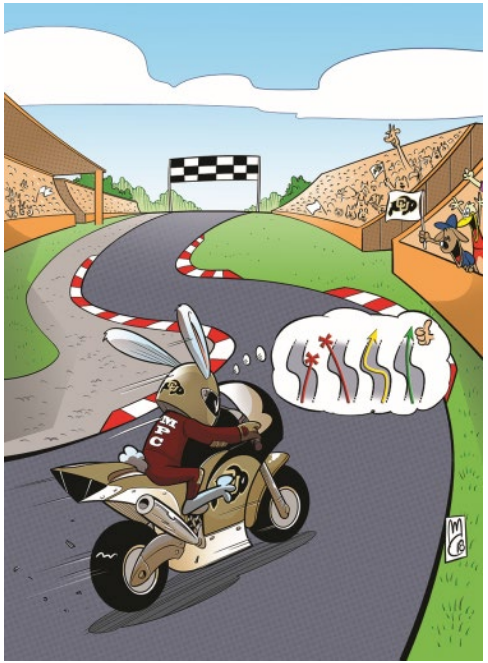
WAGENINGEN
UNIVERSITY & RESEARCH

Basic Structure of MPC



Basic Idea





MPC generates the control signal $\mathbf{u}(k)$ by solving an optimization problem at each sampling time k:

- Objective: optimal $\mathbf{u}(k)$, $\mathbf{u}(k+1)$, … $\mathbf{u}(k+p)$?
- Implement $\mathbf{u}(k)$ for step k, discard the remaining
- Repeat optimization at time $k+1$

Standard MPC

At time step k, solve

$$\min_{\mathbf{u}_k,\ldots,\mathbf{u}_{k+p}} J = \sum_{i=k}^{i=k+p} V(\mathbf{x}_i, \mathbf{u}_i)$$

subject to $\mathbf{x}_i = \mathbf{f}(\mathbf{x}_i, \mathbf{u}_i, \mathbf{d}_i; \mathbf{p})$

$\mathbf{u}_i \in \mathbb{U}$, $\mathbf{x}_i \in \mathbb{X}$, *for all* i

- **Model**: deterministic (prediction of **d**), stochastic (distribution on **d** or $\mathbf{p}$)
- **Objective (-reward + cost)**: deterministic/stochastic (expectation), *quadratic*, economic indicator, terminal cost, time
- **Constraints**: interval on control/state, chance constraints, integers, 0/1

At time step k+1, solve $\mathbf{u}_{k+1}, \ldots, \mathbf{u}_{k+p+1}$
using $\mathbf{u}_k^*, \ldots, \mathbf{u}_{k+p}^*$ as warm start

Standard MPC: continuous-time model

Linear time-invariant, deterministic
Reference signal $r_k = 0$

At time step *k*, solve

$$\min_{u_k,\ldots,u_{k+p}} J = \sum_{i=k}^{i=k+p} V(x_i, u_i)$$

subject to $x_{i+1} = Ax_i + Bu_i$
$u_i \in \mathbb{U}$, $x_i \in \mathbb{X}$, *for all i*

Linear quadratic regulation (LQR):
$V = x_{i+1}^T Q x_{i+1} + u_i^T R u_i$
$Q > 0$ *and* $R > 0$

Linear constraints:
$\mathbb{U} = \{G_u u \leq g_u\}$
$\mathbb{X} = \{G_x x \leq g_x\}$

Goal: Given x_k (measured at time step *k*), find the input sequence $u_k, \ldots, u_{k+p}$ that steers the state to the zero "optimally"

Linear MPC: A Simple Case

Finite horizon prediction:

$$x_{k+1} = Ax_k + Bu_k$$
$$x_{k+2} = Ax_{k+1} + Bu_{k+1} = A(Ax_k + Bu_k) + Bu_{k+1} = A^2x_k + ABu_k + Bu_{k+1}$$
$$x_{k+3} = A^3x_k + A^2Bu_k + ABu_{k+1} + Bu_{k+2}$$

$$\begin{bmatrix} x_{k+1} \\ x_{k+2} \\ \vdots \\ x_{k+p+1} \end{bmatrix} = \begin{bmatrix} A \\ A^2 \\ \vdots \\ A^p \end{bmatrix} x_k + \begin{bmatrix} B & 0 & \cdots & 0 \\ AB & B & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A^{p-1}B & A^{p-2}B & \cdots & B \end{bmatrix} \begin{bmatrix} u_k \\ u_{k+1} \\ \vdots \\ u_{k+p} \end{bmatrix}$$

$$\boldsymbol{X} = \boldsymbol{E}\, x_k + \boldsymbol{F}\, \boldsymbol{U}$$

Linear MPC: A Simple Case

Objective function:

$$J = \sum_{i=k}^{i=k+p} x_{i+1}^T Q x_{i+1} + u_i^T R u_i =$$

$$\begin{bmatrix} x_{k+1}^T & \cdots & x_{k+p+1}^T \end{bmatrix} \begin{bmatrix} Q & & \\ & \ddots & \\ & & Q \end{bmatrix} \begin{bmatrix} x_{k+1} \\ \vdots \\ x_{k+p+1} \end{bmatrix} + \begin{bmatrix} u_k^T & \cdots & u_{k+p}^T \end{bmatrix} \begin{bmatrix} R & & \\ & \ddots & \\ & & R \end{bmatrix} \begin{bmatrix} u_k \\ \vdots \\ u_{k+p} \end{bmatrix}$$

$$J = \quad X^T \quad \mathcal{Q} \quad X \quad + \quad U^T \quad \mathcal{R} \quad U$$

Constraints:

$$\begin{bmatrix} G_u & & \\ & \ddots & \\ & & G_u \end{bmatrix} \begin{bmatrix} u_k \\ \vdots \\ u_{k+p} \end{bmatrix} \leq \begin{bmatrix} g_u \\ \vdots \\ g_u \end{bmatrix} \quad \begin{bmatrix} G_x & & \\ & \ddots & \\ & & G_x \end{bmatrix} \begin{bmatrix} x_{k+1} \\ \vdots \\ x_{k+p+1} \end{bmatrix} \leq \begin{bmatrix} g_x \\ \vdots \\ g_x \end{bmatrix}$$

$$\mathcal{G}_u \quad U \quad \leq \quad \mathcal{g}_u \qquad \mathcal{G}_x \quad X \quad \leq \mathcal{g}_x$$

WAGENINGEN
UNIVERSITY & RESEARCH

Linear MPC: A Simple Case

$$\min_{u(k),\ldots,u(k+p)} \sum_{i=k}^{i=k+p} x_{i+1}^T Q x_{i+1} + u_i^T R u_i$$

subject to $x_{i+1} = Ax_i + Bu_i$

$u_i \in \mathbb{U}$, $x_i \in \mathbb{X}$, *for all i*

$$\boldsymbol{X} = \begin{bmatrix} x_{k+1} \\ x_{k+2} \\ \vdots \\ x_{k+p+1} \end{bmatrix}$$

$$\boldsymbol{U} = \begin{bmatrix} u_k \\ u_{k+1} \\ \vdots \\ u_{k+p} \end{bmatrix}$$

$$\min_{\boldsymbol{U}} \boldsymbol{X}^T \boldsymbol{Q} \boldsymbol{X} + \boldsymbol{U}^T \boldsymbol{\mathcal{R}} \boldsymbol{U}$$

subject to $\boldsymbol{X} = \boldsymbol{E} x_k + \boldsymbol{F} \boldsymbol{U}$

$\boldsymbol{\mathcal{G}_u} \boldsymbol{U} \leq \boldsymbol{g_u}$

$\boldsymbol{\mathcal{G}_x} \boldsymbol{X} \leq \boldsymbol{g_x}$

Define: $\boldsymbol{z} = \begin{bmatrix} \boldsymbol{X} \\ \boldsymbol{U} \end{bmatrix}, \boldsymbol{S} = \begin{bmatrix} \boldsymbol{Q} & 0 \\ 0 & \boldsymbol{\mathcal{R}} \end{bmatrix}, \boldsymbol{G} = \begin{bmatrix} \boldsymbol{\mathcal{G}_x} & 0 \\ 0 & \boldsymbol{\mathcal{G}_u} \end{bmatrix}, \boldsymbol{g} = \begin{bmatrix} \boldsymbol{g_x} \\ \boldsymbol{g_u} \end{bmatrix}, \boldsymbol{H} = [\boldsymbol{I} \quad -\boldsymbol{F}], \boldsymbol{h} = \boldsymbol{E} x_k$

quadratic programming: $\min_{\boldsymbol{z}} \boldsymbol{z}^T \boldsymbol{S} \boldsymbol{z}$ subject to $\boldsymbol{G} \boldsymbol{z} \leq \boldsymbol{g}, \boldsymbol{H} \boldsymbol{z} = \boldsymbol{h}$

Convex optimalization!

WAGENINGEN UNIVERSITY & RESEARCH

Convergence Result (based on Lyapunov theory)

Theorem: Consider the linear system $x_{k+1} = Ax_k + Bu_k$, and the MPC control law based on

$$\min_{u_k,\ldots,u_{k+p}} J = \sum_{i=k}^{i=k+p} \left(x_{i+1}^T Q x_{i+1} + u_i^T R u_i\right)$$

subject to $u_{\min} \leq u_k \leq u_{\max}$, $x_{\min} \leq x_k \leq x_{\max}$

Assume the optimalization problem is feasible at $k = 0$. Then, for all $Q > 0$ and $R > 0$,

$$\lim_{k\to\infty} x_k = 0, \qquad \lim_{k\to\infty} u_k = 0$$

And the constraints are satisfied at all time instants k

(Keerthi and Gilbert, 1988)(Bemporad *et al*., 1994)

Nonlinear MPC

Reference signal $r_k = 0$

At time step *k*, solve

$$\min_{u_k,\dots,u_{k+p}} J = \sum_{i=k}^{i=k+p} V(x_i, u_i)$$

subject to $x_{i+1} = f(x_i, u_i)$

$u_i \in \mathbb{U}$, $x_i \in \mathbb{X}$, *for all* $i = k, \dots, k+p$

Linear quadratic regulation (LQR):
$V = x_{i+1}^T Q x_{i+1} + u_i^T R u_i$
$Q > 0$ *and* $R > 0$

Linear constraints:
$\mathbb{U} = \{G_u u \leq g_u\}$
$\mathbb{X} = \{G_x x \leq g_x\}$

Goal: Given x_k, find the input sequence $u_k, \dots, u_{k+p}$ that steers the state to the zero "optimally"

Nonlinear MPC

$$\min_{u(k),\ldots,u(k+p)} \sum_{i=k}^{i=k+p} x_{i+1}^T Q x_{i+1} + u_i^T R u_i$$

subject to $x_{i+1} = f(x_i, u_i)$

$u_i \in \mathbb{U}$, $x_i \in \mathbb{X}$, *for all i*

$$\boldsymbol{X} = \begin{bmatrix} x_{k+1} \\ x_{k+2} \\ \vdots \\ x_{k+p+1} \end{bmatrix}$$

$$\boldsymbol{U} = \begin{bmatrix} u_k \\ u_{k+1} \\ \vdots \\ u_{k+p} \end{bmatrix}$$

$$\min_{\boldsymbol{U}} \boldsymbol{X}^T \boldsymbol{Q} \boldsymbol{X} + \boldsymbol{U}^T \boldsymbol{\mathcal{R}} \boldsymbol{U}$$

subject to $\boldsymbol{H}(\boldsymbol{X}, \boldsymbol{U}) = \boldsymbol{0}$

$\boldsymbol{\mathcal{G}_u} \boldsymbol{U} \leq \boldsymbol{g_u}$

$\boldsymbol{\mathcal{G}_x} \boldsymbol{X} \leq \boldsymbol{g_x}$

$$\boldsymbol{H}(\boldsymbol{X}, \boldsymbol{U}) = \begin{bmatrix} x_{k+1} - f(x_k, u_k) \\ \vdots \\ x_{k+p+1} - f(x_{k+p}, u_{k+p}) \end{bmatrix}$$

Define $\boldsymbol{z} = \begin{bmatrix} \boldsymbol{X} \\ \boldsymbol{U} \end{bmatrix}$, $\boldsymbol{S} = \begin{bmatrix} \boldsymbol{Q} & 0 \\ 0 & \boldsymbol{\mathcal{R}} \end{bmatrix}$, $\boldsymbol{G} = \begin{bmatrix} \boldsymbol{\mathcal{G}_x} & 0 \\ 0 & \boldsymbol{\mathcal{G}_u} \end{bmatrix}$, $\boldsymbol{g} = \begin{bmatrix} \boldsymbol{g_x} \\ \boldsymbol{g_u} \end{bmatrix}$

nonlinear programming:

$$\min_{\boldsymbol{z}} \boldsymbol{z}^T \boldsymbol{S} \boldsymbol{z} \quad \text{subject to} \quad \boldsymbol{G}\boldsymbol{z} \leq \boldsymbol{g}, \boldsymbol{H}(\boldsymbol{z}) = \boldsymbol{0}$$

WAGENINGEN UNIVERSITY & RESEARCH

Nonlinear MPC

$$\min_{\boldsymbol{z}} \boldsymbol{z}^T \boldsymbol{S}\boldsymbol{z} \quad \text{subject to} \quad \boldsymbol{G}\boldsymbol{z} \leq \boldsymbol{g}, \boldsymbol{H}(\boldsymbol{z}) = \boldsymbol{0} \qquad \boldsymbol{z} = \begin{bmatrix} \boldsymbol{X} \\ \boldsymbol{U} \end{bmatrix}$$

Step 1: initial guess for $\boldsymbol{z}^{(0)} = \begin{bmatrix} \boldsymbol{X}^{(0)} \\ \boldsymbol{U}^{(0)} \end{bmatrix}$,

where $\boldsymbol{X}^{(0)} = \begin{bmatrix} x_{k+1}^{(0)} \\ \vdots \\ x_{k+p+1}^{(0)} \end{bmatrix}$, $\boldsymbol{U}^{(0)} = \begin{bmatrix} u_k^{(0)} \\ \vdots \\ u_{k+p}^{(0)} \end{bmatrix}$

Use the previous time step's solution shifted forward as a **warm start**:

- At time step $k+1$, we compute $u_{k-1}^*, u_k^*, \dots, u_{k+p-1}^*$
- $u_i^{(0)} = u_{i-1}^*$ and $x_{i+1}^{(0)} = f(x_i^{(0)}, u_i^{(0)})$, for $i = k, k+1, \dots, k+p$

Nonlinear MPC

$$\min_{\boldsymbol{z}} \boldsymbol{z}^T \boldsymbol{S}\boldsymbol{z} \quad \text{subject to} \quad \boldsymbol{G}\boldsymbol{z} \leq \boldsymbol{g}, \boldsymbol{H}(\boldsymbol{z}) = \boldsymbol{0} \qquad \boldsymbol{z} = \begin{bmatrix} \boldsymbol{X} \\ \boldsymbol{U} \end{bmatrix}$$

Step 2: (SQP Iteration) For each iteration j:

- Linearize nonlinear constraints around $\boldsymbol{z}^{(j)}$:

$$\boldsymbol{H}(\boldsymbol{z}) \approx \boldsymbol{H}\left(\boldsymbol{z}^{(j)}\right) + \frac{\partial \boldsymbol{H}}{\partial \boldsymbol{z}}\left(\boldsymbol{z} - \boldsymbol{z}^{(j)}\right)$$

- Solve QP for $\boldsymbol{z}^{(j+1)}$:

$$\min_{\boldsymbol{z}} \boldsymbol{z}^T \boldsymbol{S}\boldsymbol{z} \quad \text{subject to} \quad \boldsymbol{G}\boldsymbol{z} \leq \boldsymbol{g}, \quad \boldsymbol{H}\left(\boldsymbol{z}^{(j)}\right) + \frac{\partial \boldsymbol{H}}{\partial \boldsymbol{z}}\left(\boldsymbol{z} - \boldsymbol{z}^{(j)}\right) = \boldsymbol{0}$$

IPOPT (Interior Point OPTimizer)

IPOPT: large-scale nonlinear programming (NLP) solver most commonly used for MPC. It solves optimization problems of the form:

$$\min_{\mathbf{x}} \mathbf{f}(\mathbf{x})$$

$$\text{s. t.} \quad \mathbf{g}(\mathbf{x}) = \mathbf{0} \quad \text{(equality constraints)}$$

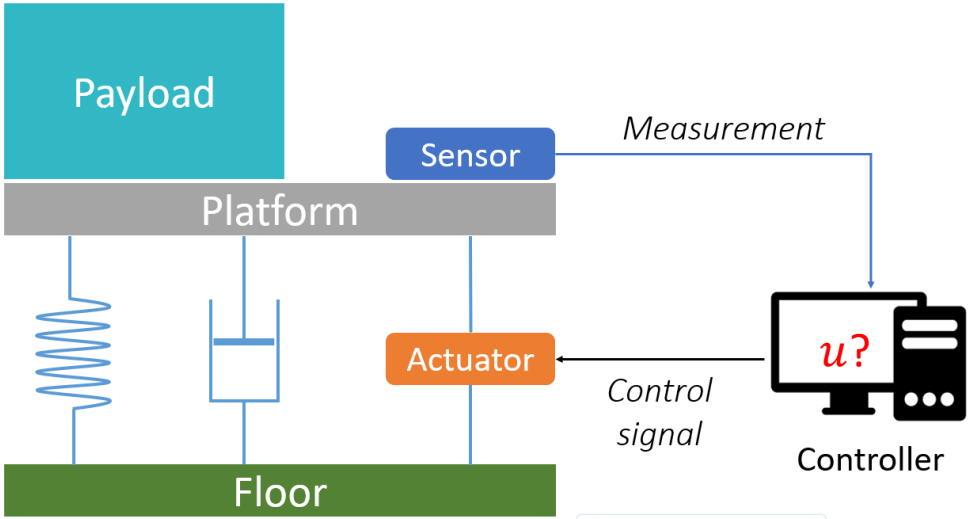
$$\mathbf{h}(\mathbf{x}) \leq \mathbf{0} \quad \text{(inequality constraints)}$$

- Inequality constraints are transformed using a **barrier function**:

$$\boldsymbol{\phi}_{\mu}(\mathbf{x}) = \mathbf{f}(\mathbf{x}) + \mu \sum_{i} \ln \mathbf{h}_i(\mathbf{x})$$

- "**barrier**": $\mathbf{h}_i(\mathbf{x})$ blows up near the boundary
- As $\mu \rightarrow 0$, solutions approach the boundary of feasible region.
- Searching in the **interior** of the **feasible region**

Example: Active Vibration Suppression



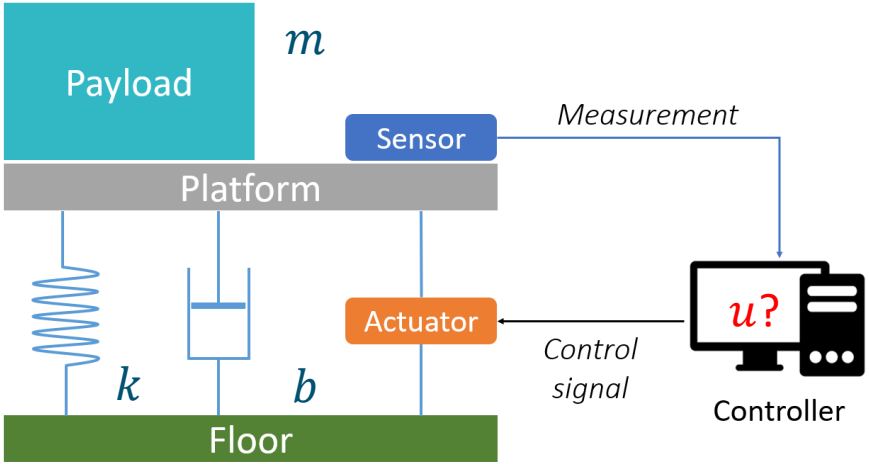
Questions: how to design control signal u to compens
undesirable vibration caused by disturbance?

Code

https://github.com/xcheng20/Greenhouse_Control_Exercise/

WAGENINGEN
UNIVERSITY & RESEARCH

Example: Active Vibration Suppression



According to Newton's law:

$$m\ddot{p}(t) = -kp(t) - b\dot{p}(t) + u(t)$$

p: displacement from the steady state

state vector $x(t) = \begin{bmatrix} p(t) \\ \dot{p}(t) \end{bmatrix}$

Continuous-time state space model

$$\dot{x}(t) = \underbrace{\begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{b}{m} \end{bmatrix}}_{A} x(t) + \underbrace{\begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix}}_{B} u$$

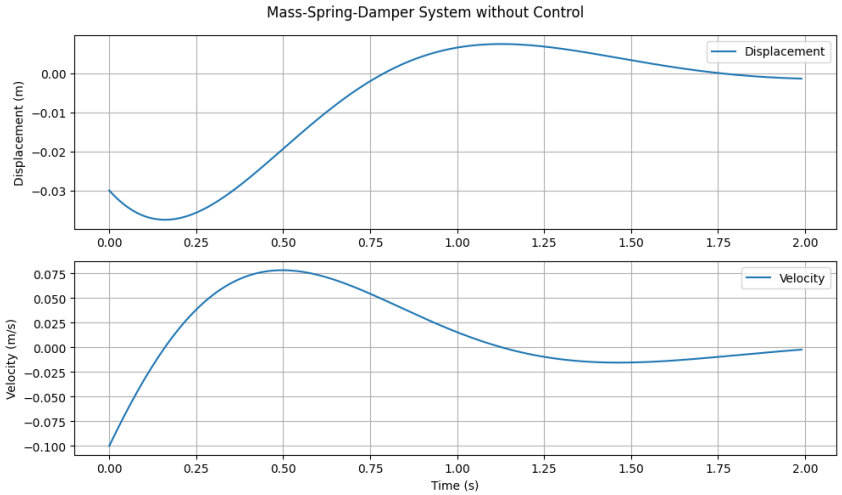
Example: Active Vibration Suppression

$$\dot{x}(t) = \underbrace{\begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{b}{m} \end{bmatrix}}_{A} x(t) + \underbrace{\begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix}}_{B} u$$

```python
def modelParameters():
    p = {}
    # parameter                 description                 [unit]
    p['m'] = 30                 # mass                      [kg]
    p['k'] = 4.0e2              # spring coefficient        [N/m]
    p['b'] = 1.0e2              # damping coefficient       [N s/m]
    return p
```

Setting: starts at a nonzero condition, e.g., $x_0 = \begin{bmatrix} -0.02 \\ -0.1 \end{bmatrix}$ (simulate the influence of a sudden disturbance)

Questions: how to design control signal u such that states can quickly go back to zero?



WAGENINGEN UNIVERSITY & RESEARCH

Example: Active Vibration Suppression

open-source symbolic framework for automatic differentiation and numerical optimization.

https://web.casadi.org/

```python
import casadi as ca
```

```python
def f(x, u, p, h):
    """
    Defines the differential equations of the mass-spring-damper model.
    This function calculates the rate of change of the state variables.
    """
    # Define state space matrices
    A = ca.DM([[0, 1], [-p['k']/p['m'], -p['b']/p['m']]])
    B = ca.DM([[0], [1/p['m']]])

    # State derivatives [dx1/dt, dx2/dt]
    return A @ x + B @ u
```

$$\dot{x}(t) = \underbrace{\begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{b}{m} \end{bmatrix}}_{A} x(t) + \underbrace{\begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix}}_{B} u$$

Example: Active Vibration Suppression

```
# Define a dictionary 'ops' to hold simulation options and parameters
ops = {}
ops['nx'] = 2      # Number of states
ops['ny'] = 2      # Number of outputs
ops['nu'] = 1      # Number of controllable inputs
ops['h']  = 0.01  # Sample period in seconds

ops['t'] = np.arange(0, 2, ops['h'])  # Time vector, simulate for 2 seconds
ops['N'] = len(ops['t'])  # Number of simulation steps
ops['Np'] = 20  # Prediction horizon (e.g., 20 steps, 0.2 seconds)

ops['x0'] = np.array([-0.03, -0.1])  # Initial state [position, velocity]

# systems constraints
PMin = -0.04  # Minimum displacement [m]
PMax =  0.04  # Maximum displacement [m]
VMin = -0.50  # Minimum velocity [m/s]
VMax =  0.50  # Maximum velocity [m/s]
CMin = -200   # Minimum control force [N]
CMax =  200   # Maximum control force [N]

# state constraints vector
ops['x_min']   = np.array([PMin, VMin])
ops['x_max']   = np.array([PMax, VMax])
# control input constraints vector
ops['u_min']   = np.array([CMin])
ops['u_max']   = np.array([CMax])
```

Define all the simulation parameters in a dictionary

Example: Active Vibration Suppression

```python
# Define symbolic variables within Opti environment
opti = ca.Opti() # Set up Opti environment for NMPC
X = opti.variable(ops['nx'], int(ops['Np']) + 1)  # State trajectory
U = opti.variable(ops['nu'], int(ops['Np']))      # Control inputs
x0_param = opti.parameter(ops['nx']) # Parameter for initial state

# Define objective function
obj = 0   # Initialize Objective
Q = ca.MX.eye(ops['nx']) * [1e8, 1e4]  # Higher weight on displacement
R = ca.MX.eye(ops['nu'])               # Lower weight on control effort
# Objective accumulation
for i in range(ops['Np']):
    x_next = fd(X[:, i], U[:, i], p, ops['h'])
    opti.subject_to(X[:, i+1] == x_next)
    obj += ca.mtimes([X[:, i].T, Q, X[:, i]]) + ca.mtimes([U[:, i].T, R, U[:, i]])
opti.minimize(obj) # Minimize objective

opti.subject_to(X[:, 0] == x0_param) # Initial state constraint
# Input constraints and state constraints
for i in range(int(ops['Np'])):
    opti.subject_to(U[:, i] >= ops['u_min'])
    opti.subject_to(U[:, i] <= ops['u_max'])
    opti.subject_to(X[:, i] >= ops['x_min'])
    opti.subject_to(X[:, i] <= ops['x_max'])

# Solver
opti.solver('ipopt', {'ipopt': {'print_level': 5, 'max_iter': 1000}})
```

Numerical integration

```python
def fd(x, u, p, h):
    """
    Discrete-time state transition function using 4th-order Runge-Kutta
    integration.
    h: sample period
    """
    k1 = f(x, u, p, h)
    k2 = f(x + h/2 * k1, u, p, h)
    k3 = f(x + h/2 * k2, u, p, h)
    k4 = f(x + h * k3, u, p, h)
    x_next = x + h/6 * (k1 + 2*k2 + 2*k3 + k4)
    return x_next
```

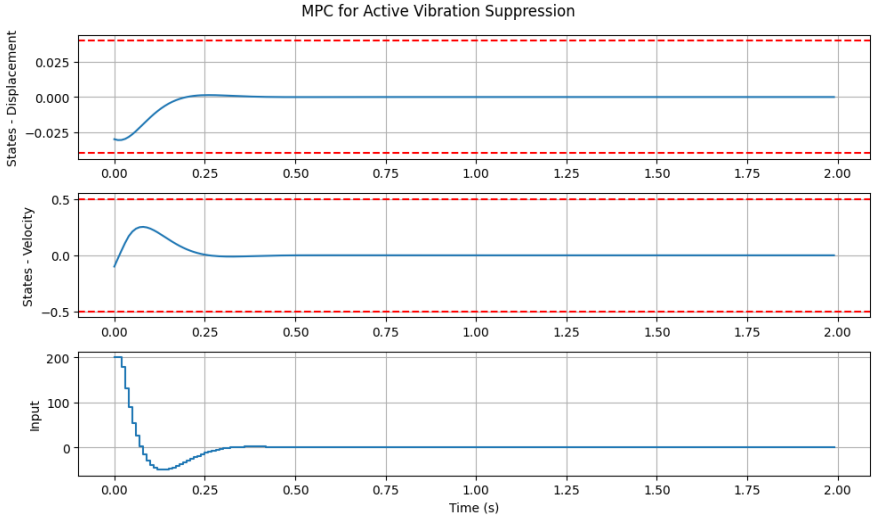
Define objective function

Constraints

Define Solver (**Interior Point OPTimizer**)

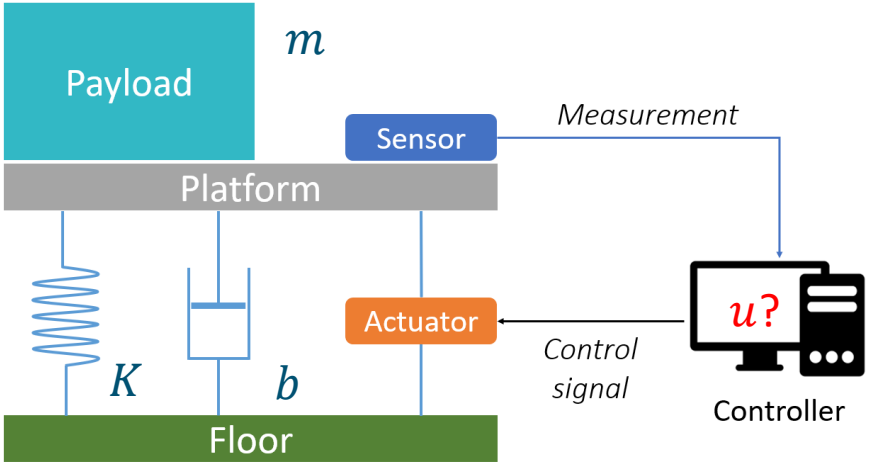
Example: Active Vibration Suppression

```
# Simulate with MPC
x_sim = np.zeros((ops['nx'], ops['N'] + 1))
u_sim = np.zeros((ops['nu'], ops['N']))
x_sim[:, 0] = ops['x0']  # Initial state

for k in range(ops['N']):
    # Set parameter (current state)
    opti.set_value(x0_param, x_sim[:, k])

    # Solve
    try:
        sol = opti.solve() # Solve the optimization problem
        u_sim[:, k] = sol.value(U[:, 0])  # Apply the first control input
        # Simulate next state
        x_next = fd(x_sim[:, k], u_sim[:, k], p, ops['h']).full().flatten()
        x_sim[:, k + 1] = x_next
```



- Average computation time: 0.007462s (< sampling time 0.01s)
- position and velocity are driven to the **zero equilibrium** much quicker

Example: Active Vibration Suppression



According to Newton's law:

$$m\ddot{p}(t) = -Kp(t)^3 - b\dot{p}(t) + u(t)$$

p: displacement from the steady state

state vector $x(t) = \begin{bmatrix} p(t) \\ \dot{p}(t) \end{bmatrix}$

Continuous-time nonlinear state space model:

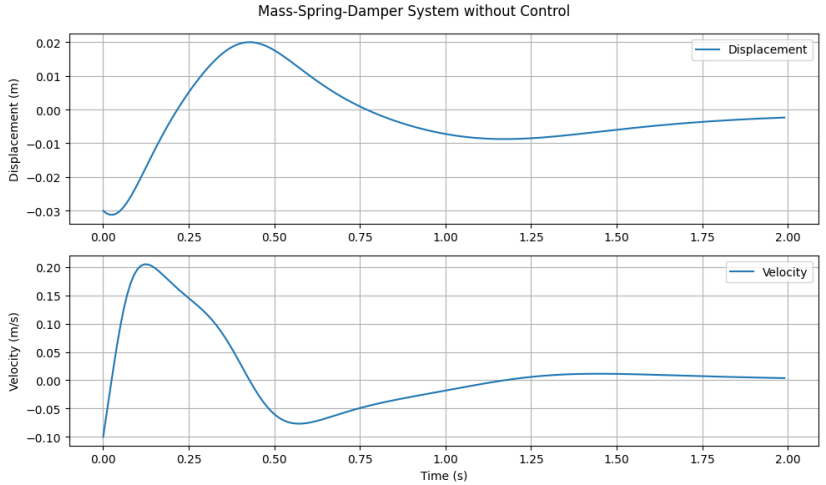
$$\dot{x}_1(t) = x_2(t)$$
$$\dot{x}_2(t) = -\frac{K}{m}x_1(t)^3 - \frac{b}{m}x_2 + \frac{1}{m}u(t)$$

Example: Active Vibration Suppression

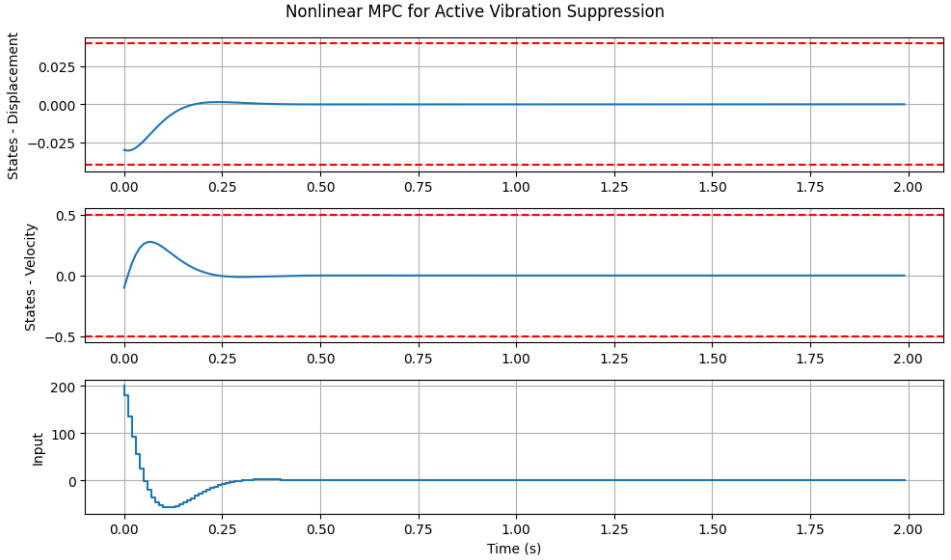
$$\dot{x}_1(t) = x_2(t)$$
$$\dot{x}_2(t) = -\frac{K}{m}x_1(t)^3 - \frac{b}{m}x_2 + \frac{1}{m}u(t)$$

```python
def f_nl(x, u, p, h):
    """
    Defines the differential equations of the mass-spring-damper model.
    This function calculates the rate of change of the state variables.
    """
    # State derivatives [dx1/dt, dx2/dt]
    ki = ca.vertcat(
        x[1],
        - p['k'] * 1e4 * x[0]**3 / p['m'] - p['b'] * x[1] / p['m'] + 1/p['m'] * u,
    )
    return ki
```

Control free simulation:
starts at a nonzero condition,
e.g., $x_0 = \begin{bmatrix} -0.02 \\ -0.1 \end{bmatrix}$

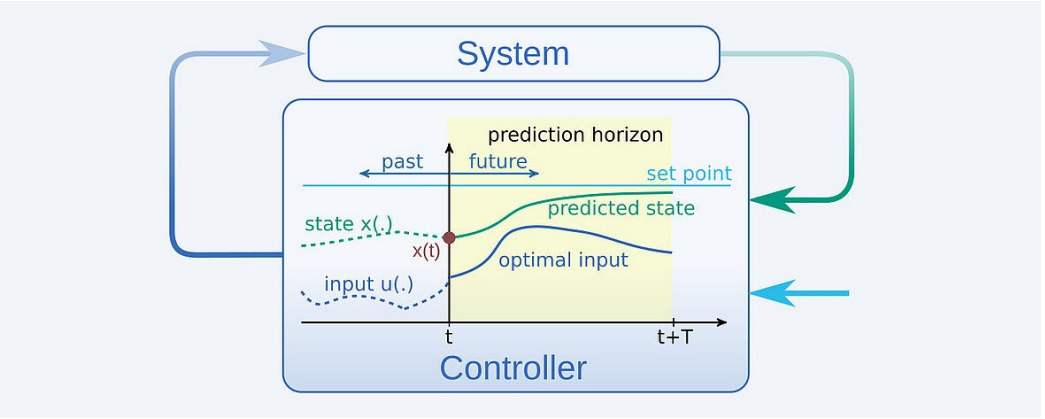


```python
# Define symbolic variables within Opti environment
opti = ca.Opti() # Set up Opti environment for NMPC
X = opti.variable(ops['nx'], int(ops['Np']) + 1)  # State trajectory
U = opti.variable(ops['nu'], int(ops['Np']))      # Control inputs
x0_param = opti.parameter(ops['nx']) # Parameter for initial state

# Define objective function
obj = 0   # Initialize Objective
Q = ca.MX.eye(ops['nx']) * [1e8, 1e4]  # Higher weight on displacement
R = ca.MX.eye(ops['nu'])               # Lower weight on control effort
# Objective accumulation
for i in range(ops['Np']):
    x_next = fd_nl(X[:, i], U[:, i], p, ops['h'])
    opti.subject_to(X[:, i+1] == x_next)
    obj += ca.mtimes([X[:, i].T, Q, X[:, i]]) + ca.mtimes([U[:, i].T, R, U[:, i]])
opti.minimize(obj) # Minimize objective

opti.subject_to(X[:, 0] == x0_param) # Initial state constraint
# Input constraints and state constraints
for i in range(int(ops['Np'])):
    opti.subject_to(U[:, i] >= ops['u_min'])
    opti.subject_to(U[:, i] <= ops['u_max'])
    opti.subject_to(X[:, i] >= ops['x_min'])
    opti.subject_to(X[:, i] <= ops['x_max'])

# Solver
opti.solver('ipopt', {'ipopt': {'print_level': 5, 'max_iter': 1000}})

# Simulate with MPC
x_sim = np.zeros((ops['nx'], ops['N'] + 1))
u_sim = np.zeros((ops['nu'], ops['N']))
x_sim[:, 0] = ops['x0']  # Initial state

for k in range(ops['N']):
    # Set parameter (current state)
    opti.set_value(x0_param, x_sim[:, k])

    # Solve
    try:
        sol = opti.solve() # Solve the optimization problem
        u_sim[:, k] = sol.value(U[:, 0])  # Apply the first control input
        # Simulate next state
        x_next = fd_nl(x_sim[:, k], u_sim[:, k], p, ops['h']).full().flatten()
        x_sim[:, k + 1] = x_next
```



- Average computation time: 0.009260s (< sampling time 0.01s > 24% higher than LMPC)
- position and velocity are driven to the **zero equilibrium** much quicker

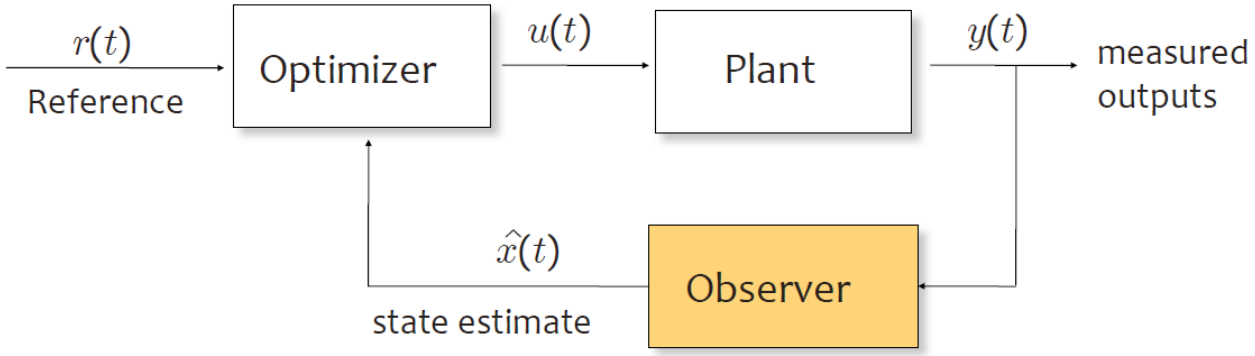
Variations of MPC



Standard MPC Methods:

- **Linear MPC**
  - Uses a linear state-space model.
  - Solves a quadratic program (QP) at each step.
  - Good for systems that are reasonably linear in the operating range
- **Nonlinear MPC (NMPC)**
  - Uses nonlinear dynamics in the prediction model.
  - Optimization becomes nonlinear programming (NLP).
  - More accurate for nonlinear systems, but computationally heavier.

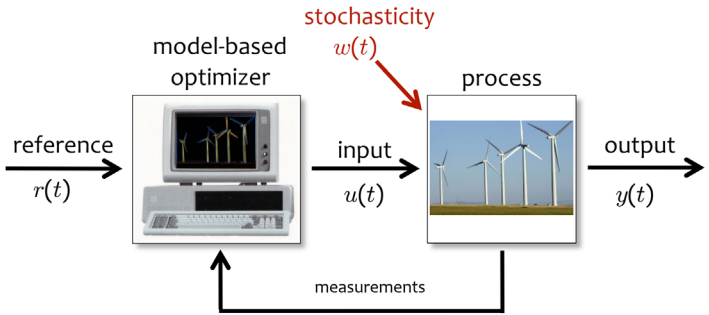
Variations of MPC



- Full state x(t) may not be available
- Even if available, noise should be filtered out
- State of prediction model may be different from plant model x(t) (e.g.: model reduction, identification)

Variations of MPC

- **Stochastic MPC (Robust MPC)**
  - Systems contains uncertainty (in process variables or parameters)
  - Objective is in expectation
  - Handles uncertainty via probabilistic constraints (chance constraints)



$$\min_{u(k)} \mathbb{E}\left[\sum_{k=k_0}^{k_0+N_p} V\left(\Delta u(k), \Delta y(k)\right)\right],$$

$$\text{s.t.} \quad \Delta x(k+1) = A_{k_0}\Delta x(k) + B_{k_0}\Delta u(k) + E_{k_0}\Delta d(k) + F_{k_0}\Delta p$$

$$\Delta y(k) = C_{k_0}\Delta x(k), \quad \Delta p_k \sim \mathcal{F}_b^a\left(\Delta\mu_p, \Sigma_p\right)$$

$$\Delta u_{\min} \leqslant \Delta u(k) \leqslant \Delta u_{\max},$$

$$|\Delta u(k) - \Delta u(k-1)| \leqslant \delta u,$$

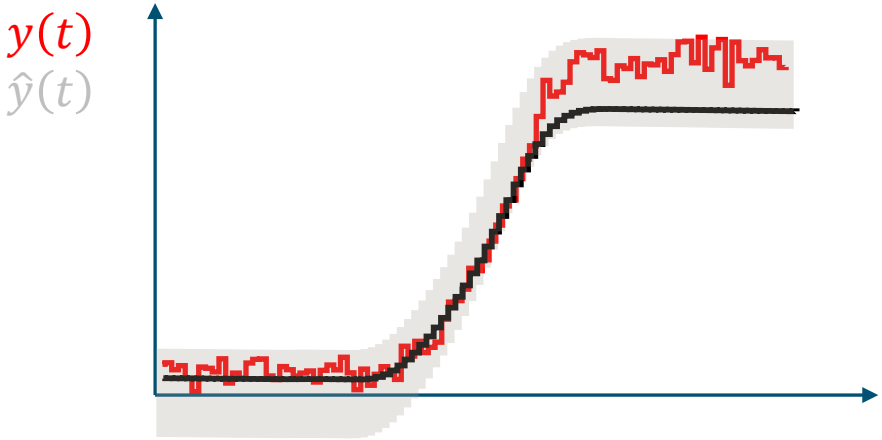
$$\mathcal{P}\left(\Delta y(k) \leqslant \Delta y_{\max}(k)\right) \geqslant \beta,$$

$$\mathcal{P}\left(\Delta y_{\min}(k) \leqslant \Delta y(k)\right) \geqslant \beta, \quad \text{for } k = k_0, \ldots, k_0 + N_p,$$

$$\Delta x(k_0) = x(k_0) - x^*(k_0),$$

Svensen, Jan Lorenz, et al. "Chance-constrained stochastic MPC of greenhouse production systems with parametric uncertainty." *Computers and Electronics in Agriculture* 217 (2024): 108578.

Modeling Uncertainty in Systems



we can 'enrich' the predictive model to predict the measurement's **distribution**

How to enrich our model? We can include:

- measurement noise (effect of sensor noise)
- parametric uncertainty (spring/damping coefficients)
- uncertain future disturbance (weather forecast)

Modeling Uncertainty in Systems

Uncertain discrete-time model

$$\mathbf{x}(k+1) = \mathbf{f}(\mathbf{x}(k), \mathbf{u}(k), \mathbf{d}(k), \mathbf{p}, \boldsymbol{w}(k))$$

$$\mathbf{y}(k) \quad = \mathbf{g}(\mathbf{x}(k), \mathbf{u}(k), \mathbf{d}(k), \mathbf{p}, \boldsymbol{v}(k))$$

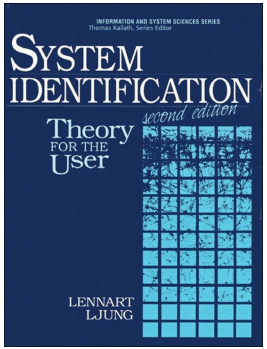
$\mathbf{p} \quad \sim \mathcal{N}(\mu_p, \Sigma_p)$

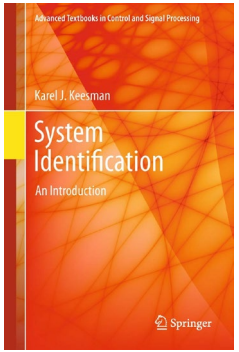
$\mathbf{d}(k) \sim \mathcal{N}(\mu_d, \Sigma_d)$ *(uncertain disturbance)*

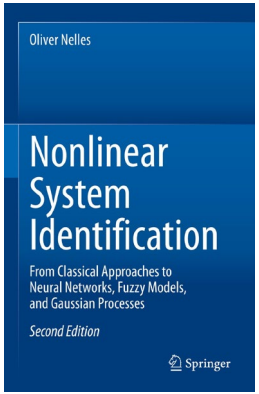
$\boldsymbol{w}(k) \sim \mathcal{N}(\mu_w, \Sigma_w)$ *(process noise)*

$\boldsymbol{v}(k) \sim \mathcal{N}(\mu_v, \Sigma_v)$ *(measurement noise)*

System identification







Stochastic MPC

At time step k, solve stochastic optimization problem:

$$\min_{\mathbf{u}_k,\ldots,\mathbf{u}_{k+p}} \mathbb{E}\left[\sum_{i=k}^{k+p} V\big(\mathbf{x}(i),\mathbf{u}(i),\mathbf{d}(i)\big)\right]$$

subject to: $\mathbf{x}(i+1) = \mathbf{f}(\mathbf{x}(i),\mathbf{u}(i),\mathbf{d}(i),\mathbf{p})$

$$\Pr\{\mathbf{g}(\mathbf{x}(i),\mathbf{u}(i),\mathbf{d}(i),\mathbf{p}) \geq 0\} \geq \beta$$

$$\mathbf{p} \sim \mathcal{N}(\boldsymbol{\mu},\boldsymbol{\Sigma})$$

- minimizing the expected value of the cost function
- chance constraints (probability that a constraint holds)

Stochastic MPC

How to solve stochastic optimization problem for MPC?

Taking samples:

$$\mathbf{p} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \xrightarrow[j = 1, \dots, N_s]{\mathbf{p}^{(j)}}$$

$$\mathbf{x}(i+1)^{(j)} = \mathbf{f}\left(\mathbf{x}(i)^{(j)}, \mathbf{u}(i), \mathbf{d}(i)^{(j)}, \mathbf{p}^{(j)}\right)$$

$$\mathbf{g}\left(\mathbf{x}(i)^{(j)}, \mathbf{u}(i), \mathbf{d}(i)^{(j)}, \mathbf{p}^{(j)}\right) \geq \mathbf{0}$$

$$\min_{\mathbf{u}_k, \dots, \mathbf{u}_{k+p}} \mathbb{E}\left[\sum_{i=k}^{k+p} V\left(\mathbf{x}(i), \mathbf{u}(i), \mathbf{d}(i)\right)\right] \quad \Rightarrow \quad \min_{\mathbf{u}_k, \dots, \mathbf{u}_{k+p}} \frac{1}{N_s} \sum_{j=1}^{N_s} \sum_{i=k}^{k+p} V\left(\mathbf{x}(i)^{(j)}, \mathbf{u}(i), \mathbf{d}(i)^{(j)}\right)$$

Note that the optimization problem becomes **large** if taking more samples, while more samples approximates better the original problem.

Robust MPC

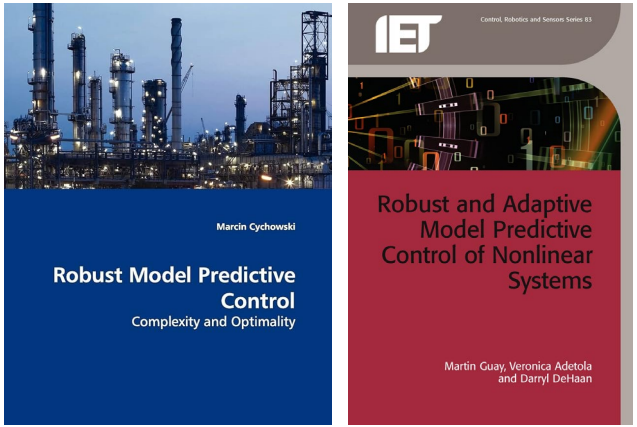
At time step k, solve optimization problem:

$$\min_{\mathbf{u}_k,\ldots,\mathbf{u}_{k+p}} \max_{\mathbf{p}} \sum_{i=k}^{k+p} V\big(\mathbf{x}(i), \mathbf{u}(i), \mathbf{d}(i)\big)$$

subject to: $\mathbf{x}(i+1) = \mathbf{f}(\mathbf{x}(i), \mathbf{u}(i), \mathbf{d}(i), \mathbf{p})$

$$\mathbf{g}(\mathbf{x}(i), \mathbf{u}(i), \mathbf{d}(i), \mathbf{p}) \geq 0, \ \forall \mathbf{p} \in \mathcal{P}$$

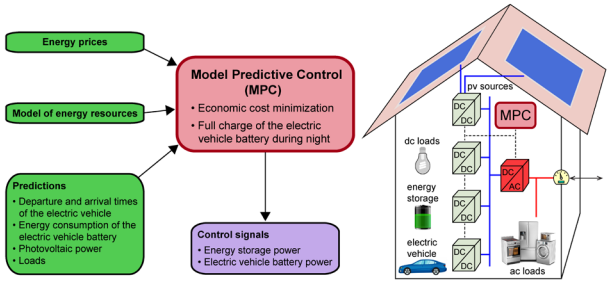
$$\mathbf{p} \in \mathcal{P}$$



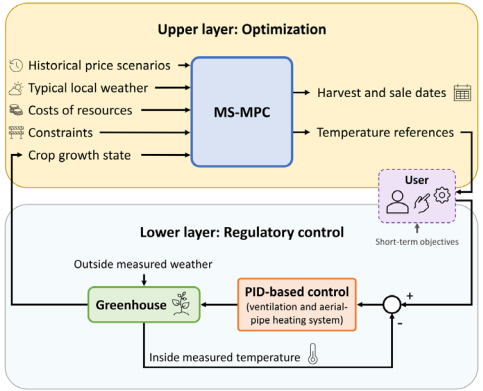
Robust MPC is in general conservative since it considers the worst case

Variations of MPC

- **Economic MPC**
  - Objective includes economic performance (e.g., cost, efficiency, profit)
  - Often steady-state may not be the goal (e.g., in energy systems, it optimizes operation cost subject to constraints)



Simmini, Francesco, et al. "Model predictive control for efficient management of energy resources in smart buildings." Energies 14.18 (2021): 5592



Harvest income Operational cost

$$\max_{\{n_{\text{days}};\, \boldsymbol{n};\, \boldsymbol{x}^{\text{r}}_{\text{T,a}}\}} \sum_{i=1}^{N_{\text{s}}} \left(\sum_{j=1}^{N_{\text{h}}} I_i(j) - \sum_{k=1}^{2n_{\text{days}}+1} f_{\text{hc}}(k) \right)$$

subject to:

$$n_{\text{days}} \in \mathbb{Z} : n_{\text{days}_{\min}} \leq n_{\text{days}} \leq n_{\text{days}_{\max}}$$

$$n_1 \in \mathbb{Z} : n_{\text{days}_{\min}} \leq n_1 \leq n_{\text{days}_{\min}} + 7$$

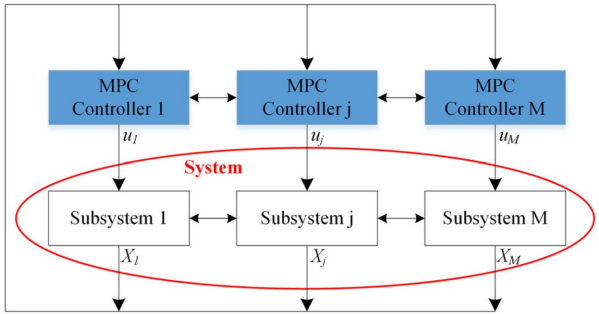
$$n_j \in \mathbb{Z} : n_{j-1} + 7 \leq n_j \leq n_{j-1} + 14, \qquad \text{for } j > 1$$

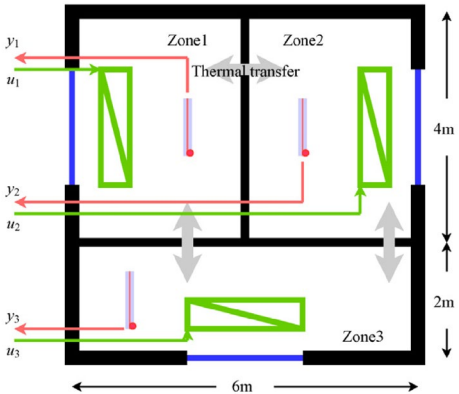
$$x^{\text{r}}_{\text{T,a}}(k) \in \mathbb{R} : x^{\text{r}}_{\text{T,a}_{\min}}(k) \leq x^{\text{r}}_{\text{T,a}}(k) \leq x^{\text{r}}_{\text{T,a}_{\max}}(k)$$

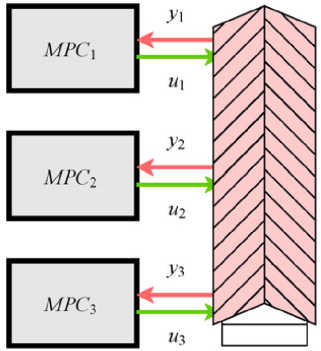
Garcia-Manas, Francisco, et al. "Multi-scenario model predictive control for greenhouse crop production considering market price uncertainty." IEEE Transactions on Automation Science and Engineering 21.3 (2023): 2936-2948.

Variations of MPC

- **Distributed MPC**
  - Each subsystem (agent) solves a local MPC problem, with coordination
  - Useful in large-scale interconnected systems (power grids, building)





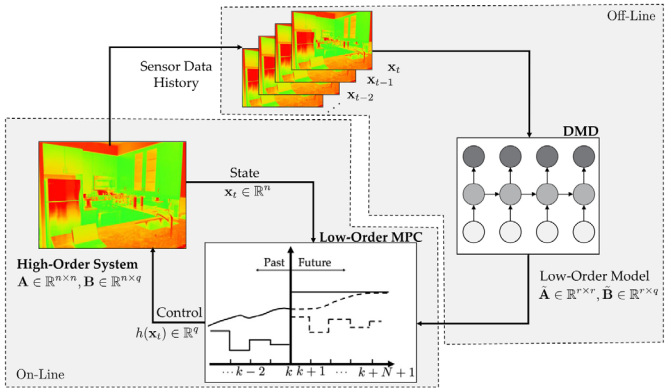


Paran, Sanaz, et al. "MPC-based distributed control for intelligent energy management of ac microgrids." Electric Power Components and Systems 47.16-17 (2019): 1437-1449.

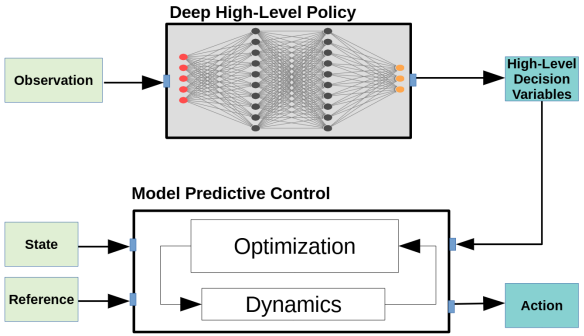
Moroşan, Petru-Daniel, et al. "Building temperature regulation using a distributed model predictive control." Energy and Buildings 42.9 (2010): 1445-1452.

Variations of MPC

- **Learning-based MPC**
  - Model parameters updated online (using data).
  - Incorporates machine learning (Gaussian Processes, Neural Networks) to improve the prediction model
  - MPC acts as a safe baseline; ML refines high-level policy



Lu, Qiugang, and Victor M. Zavala. "Image-based model predictive control via dynamic mode decomposition." Journal of Process Control 104 (2021): 146-157.



Song, Yunlong, and Davide Scaramuzza. "Learning high-level policies for model predictive control." 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2020.

Conclusion Remarks

- **Summary**
  - Background of Systems Control
  - Standard MPC: Linear MPC and Nonlinear MPC, with Case study
  - Variations of MPC (in particular, robust MPC)
- **Remarks**
  - powerful framework for modern control/decision making, unifying prediction, optimization, and feedback.
  - systematically handling constraints, multi-variables, and diverse objectives
  - Strengthen by machine learning techniques

Special thanks to
Sjoerd Boersma