Input scheduling under constrained dynamics and receding-horizon control with uncertain preview

Adair Lang

Supervisors: Michael Cantoni and Iman Shames

Department of Electrical and Electronic Engineering, University of Melbourne

PhD completion seminar[1]

February 12, 2021

[1] Supported by Australian Research Council, Linkage Grant LP160100666 with Rubicon Water as Partner Organization

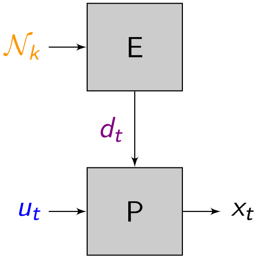
Outline

Consider a dynamical system P that operates in environment E.

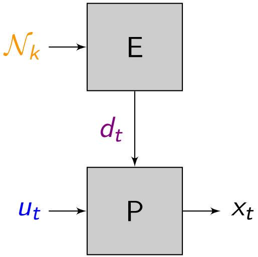
The dynamical system has two inputs:

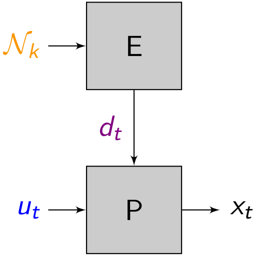
1. a control input u_t,
2. an input d_t, determined by the environment E.

The state x_t is the measured output



The operating environment E can be influenced through signal $\mathcal{N}_k$.





Primary objective: Constraint satisfaction

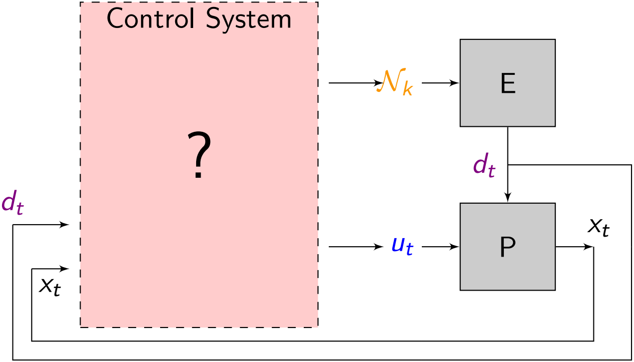
Given time horizon $\mathcal{T} \subset \mathbb{R}_{\geq 0}$ and constraint sets $(\mathcal{Y}_t)_{t \in \mathcal{T}}$ ensure

$$(\forall t \in \mathcal{T}) \; (x_t, u_t, d_t) \in \mathcal{Y}_t$$

Secondary objectives: Performance

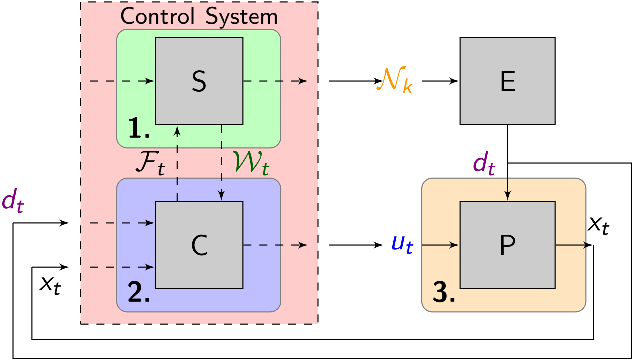
Secondary objectives include:

- minimizing running cost of dynamical system P;
- minimizing cost incurred by the operating environment E;



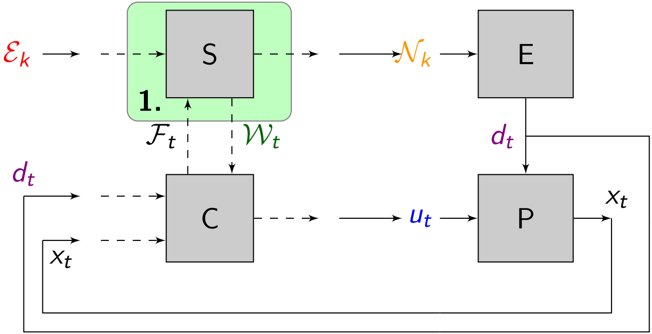
High-level problem

Given uncertain information about E, design a *control system* to meet the primary objective (constraint satisfaction) with consideration of secondary objectives (performance optimization).



One approach is to have a control hierarchy.

1. The top layer is a scheduler S that exerts influence over the operating environment E.
2. The middle layer is the controller C that steer dynamics of P through constraints using a model of scheduler influence over E
3. The bottom layer layer is the dynamical system P

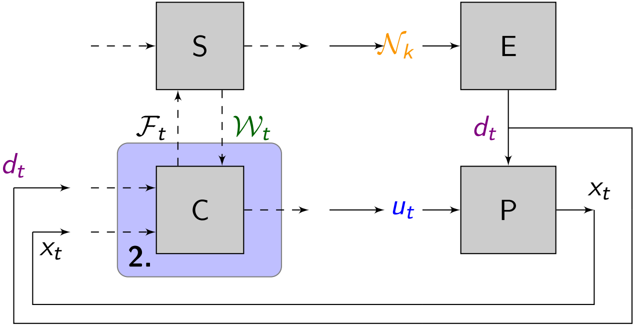


Scheduler Inputs:

- $\mathcal{E}_k$, constraints and preferences from the operating environment;
- $\mathcal{F}_t$, constraints from the controller regarding evolution of the uncertain model of future disturbances.

Scheduler Outputs:

- $\mathcal{N}_k$, to influence how E will produce disturbances
- $\mathcal{W}_t$, model of uncertain response of E to $\mathcal{N}_k$

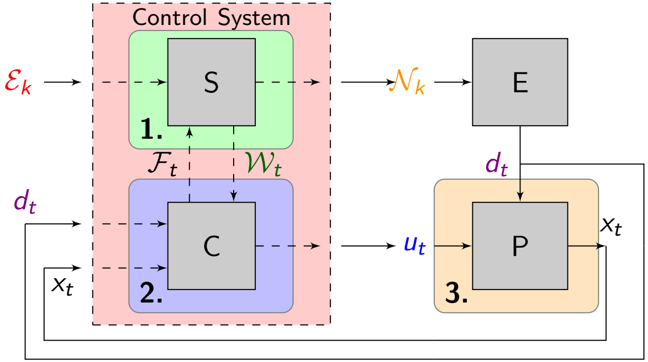


Controller Inputs:

- $\mathcal{W}_t$, uncertain model of future disturbances from scheduler S;
- d_t, disturbance input to the dynamical system P;
- x_t, state of dynamical system P.

Controller Outputs:

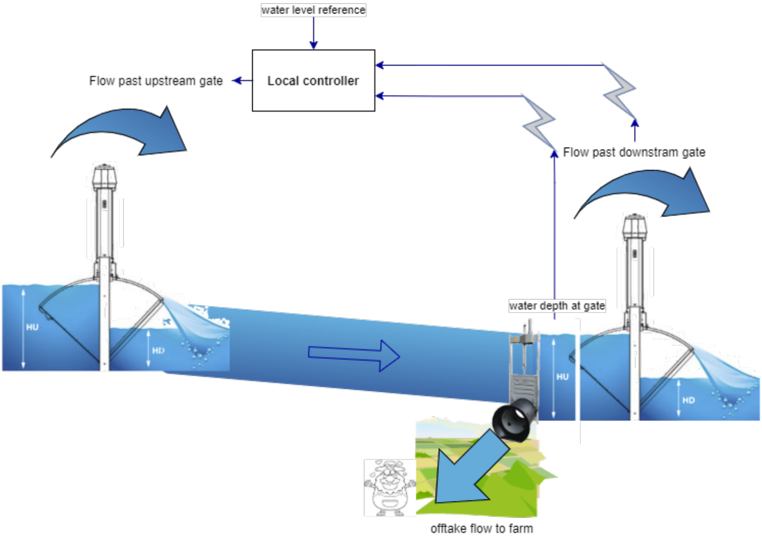
- u_t, control-input to dynamical system P
- $\mathcal{F}_t$, interface signal to scheduler.



Our focus: Optimization based approaches for designing aspects of the scheduler S and controller C.

Aiming for tractability

Motivating Application: Automated Irrigation Channels



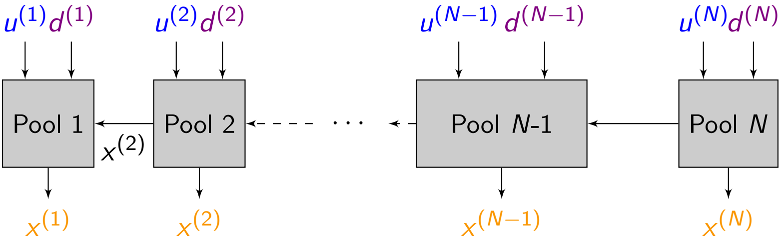
P

Images courtesy of Rubicon Water

Motivating Application: Automated Irrigation Channels

Dynamical System P

- Control input u: the water level references for each pool $u^{(i)}$, $i \in [1:N]$
- Disturbance input d: the offtake flows from each pool $d^{(i)}$, $i \in [1:N]$
- State x: the levels, flows and local controller states of all pools



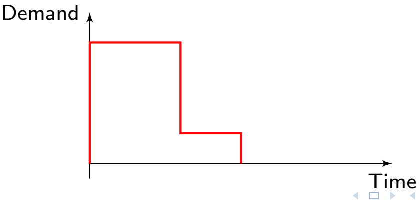
Motivating Application: Automated Irrigation Channels

;

Environment E

The users (e.g., farmers) of the system. The information $\mathcal{E}_k$ consists of:

- requested rigid-profile load inputs for each user;



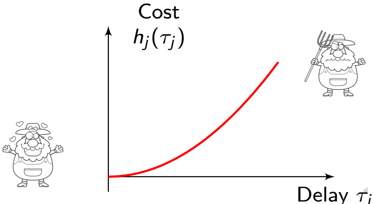
Motivating Application: Automated Irrigation Channels

;

Environment E

The users (e.g., farmers) of the system. The information $\mathcal{E}_k$ consists of:

- requested rigid-profile load inputs for each user;
- sensitivity of each user to shifting of its request;



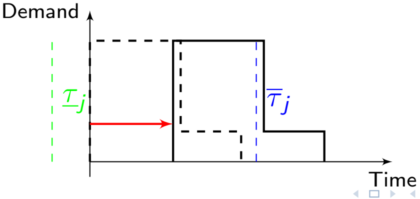
Motivating Application: Automated Irrigation Channels


;

Environment E

The users (e.g., farmers) of the system. The information $\mathcal{E}_k$ consists of:

- requested rigid-profile load inputs for each user;
- sensitivity of each user to shifting of its request;
- shift intervals for each user



Motivating Application: Automated Irrigation Channels

Scheduler S

Given $\mathcal{E}_k$ generate a nominal schedule $\mathcal{N}_k$ (set of shifts) to minimize a social measure of sensitivity to deviation of requested delivery time, while ensuring:

- response of P to scheduled offtakes is within constraints for entire scheduling horizon;
- scheduled shifts are within bounds.

Motivating Application: Automated Irrigation Channels

Scheduler S

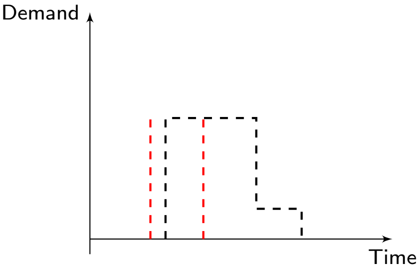
Given $\mathcal{E}_k$ generate a nominal schedule $\mathcal{N}_k$ (set of shifts) to minimize a social measure of sensitivity to deviation of requested delivery time, while ensuring:

- response of P to scheduled offtakes is within constraints for entire scheduling horizon;
- scheduled shifts are within bounds.

Problem 1 is about solving this scheduling problem

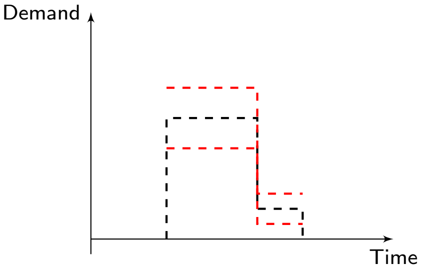
Automated Irrigation Channels

There may be uncertainty in how the operating environment responds to the nominal schedule $\mathcal{N}_k$ when generating actual offtakes flows d_t.
For example:



Uncertainty in start time

There may be uncertainty in how the operating environment responds to the nominal schedule $\mathcal{N}_k$ when generating actual offtakes flows d_t.
For example:



Uncertainty in magnitude

Automated Irrigation Channels

Controller C

Given at time $t \in \mathcal{T}$:

- uncertain model of scheduler influence over the environment $\mathcal{W}_t$;
- real-time measurement of the disturbance input d_t (i.e., outlet flows)
- real-time measurement of state x_t;

determine the water-level references u_t and an interface to the scheduler ($\mathcal{F}_t$) that ensures the system remains within constraints.

Automated Irrigation Channels

Controller C

Given at time $t \in \mathcal{T}$:

- uncertain model of scheduler influence over the environment $\mathcal{W}_t$;
- real-time measurement of the disturbance input d_t (i.e., outlet flows)
- real-time measurement of state x_t;

determine the water-level references u_t and an interface to the scheduler ($\mathcal{F}_t$) that ensures the system remains within constraints.

QUESTIONS:

- How can $\mathcal{W}_t$ be allowed to evolve?
- How to manage the balance between scheduler and control objectives?

Problems considered

Two problems considered

- Rigid-profile input scheduling; and
- Receding horizon control under uncertain preview.

Rigid-profile input scheduling

Rigid-profile input scheduling problem

Given:

- continuous-time LTI dynamical system P;
- finite scheduling horizon of length $T \in \mathbb{R}_{>0}$, i.e., $\mathcal{T} = [0, T]$;
- requested (future) rigid-profile disturbance inputs for P, sensitivity to deviation from requested timing, as encoded in $\mathcal{E}_k$;
- fixed nominal control input $\bar{u} \in \mathbb{R}^m$;

determine

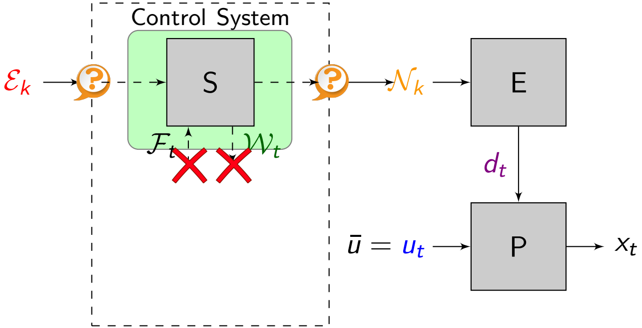
- the 'best' nominal schedule $\mathcal{N}_k$,

such that

- the dynamic response of P to scheduled disturbance inputs and fixed nominal control-input is inside the constraints for the entire continuous horizon.

Rigid-profile input scheduling

Problem relates to generating the nominal influence signal $\mathcal{N}_k$ only, not to the interaction with C through $\mathcal{F}_t$ and $\mathcal{W}_t$.



Optimization problem

Formulation as a non-convex semi-infinite program

$$f^* := \min_{(\tau_j)_{j=1}^m} f(\tau_1, \ldots, \tau_m)$$

$$\text{s.t.} \quad Cx(t; (v_j, \ \tau_j \ )_{j=1}^m) \leq c \text{ for } t \in \ \mathcal{T} \ ,$$

$$\tau_j \in [\underline{\tau}_j, \overline{\tau}_j] \text{ for } j \in [1 : m],$$

where $f(\tau_1, \ldots, \tau_m) := \sum_{j=1}^m h_j(\tau_j)$.

- $x(t, (v_j, \tau_j)_{j=1}^m)$ is the evolution of state of dynamical system

Optimization problem

Formulation as a non-convex semi-infinite program

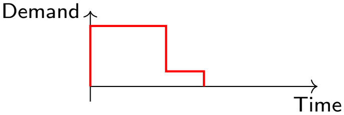
$$f^* := \min_{(\tau_j)_{j=1}^m} f(\tau_1, \dots, \tau_m)$$

$$\text{s.t.} \quad Cx(t; (v_j, \tau_j)_{j=1}^m) \leq c \text{ for } t \in \mathcal{T},$$

$$\tau_j \in [\underline{\tau}_j, \overline{\tau}_j] \text{ for } j \in [1:m],$$

where $f(\tau_1, \dots, \tau_m) := \sum_{j=1}^m h_j(\tau_j)$.

- $x(t, (v_j, \tau_j)_{j=1}^m)$ is the evolution of state of dynamical system
- $(v_j)_{j=1}^m$ are the requested rigid profiles;



Optimization problem

Formulation as a non-convex semi-infinite program

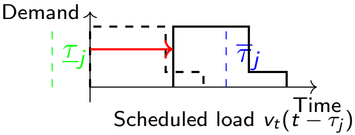
$$f^* := \min_{(\tau_j)_{j=1}^m} f(\tau_1, \ldots, \tau_m)$$

$$\text{s.t.} \quad Cx(t; (v_j, \tau_j)_{j=1}^m) \leq c \text{ for } t \in \mathcal{T},$$

$$\tau_j \in [\underline{\tau}_j, \overline{\tau}_j] \text{ for } j \in [1:m],$$

where $f(\tau_1, \ldots, \tau_m) := \sum_{j=1}^m h_j(\tau_j)$.

- $x(t, (v_j, \tau_j)_{j=1}^m)$ is the evolution of state of dynamical system
- $(v_j)_{j=1}^m$ are the requested rigid profiles;
- $(\tau_j)_{j=1}^m$ are the decision variables (shifts).



Optimization problem

Formulation as a non-convex semi-infinite program

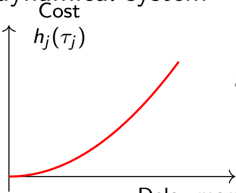
$$f^* := \min_{(\tau_j)_{j=1}^m} f(\tau_1, \ldots, \tau_m)$$

$$\text{s.t.} \quad Cx(t; (v_j, \tau_j)_{j=1}^m) \leq c \text{ for } t \in \mathcal{T},$$

$$\tau_j \in [\underline{\tau}_j, \overline{\tau}_j] \text{ for } j \in [1:m],$$

where $f(\tau_1, \ldots, \tau_m) := \sum_{j=1}^m h_j(\tau_j)$.

- $x(t, (v_j, \tau_j)_{j=1}^m)$ is the evolution of state of dynamical system
- $(v_j)_{j=1}^m$ are the requested rigid profiles;
- $(\tau_j)_{j=1}^m$ are the decision variables (shifts).
- $(h_j)_{j=1}^m$ user sensitivity to shifts.



Challenges

Formulation as a non-convex semi-infinite program

$$f^* := \min_{(\tau_j)_{j=1}^m} f(\tau_1, \ldots, \tau_m)$$

$$\text{s.t.} \quad Cx(t; (v_j, \tau_j)_{j=1}^m) \leq c \text{ for } t \in \mathcal{T},$$

$$\tau_j \in [\underline{\tau}_j, \overline{\tau}_j] \text{ for } j \in [1:m],$$

where $f(\tau_1, \ldots, \tau_m) := \sum_{j=1}^m h_j(\tau_j)$.

- Constraints are non-convex in shift variables

Challenges

Formulation as a non-convex semi-infinite program

$$f^* := \min_{(\tau_j)_{j=1}^m} f(\tau_1, \ldots, \tau_m)$$

$$\text{s.t.} \quad Cx(t; (v_j, \tau_j)_{j=1}^m) \le c \text{ for } t \in \mathcal{T},$$

$$\tau_j \in [\underline{\tau}_j, \overline{\tau}_j] \text{ for } j \in [1:m],$$

where $f(\tau_1, \ldots, \tau_m) := \sum_{j=1}^m h_j(\tau_j)$.

- Constraints are non-convex in shift variables
- Infinite number of constraints

Approach

We consider a two-stage approach to computing feasible solutions

Approach

A two-stage approach to computing feasible solutions

First Stage

- Discretization of the decision variables
  $\hat{\mathcal{D}}_j := \{\tau_j^{(1)}, \ldots, \tau_j^{(N_j)}\} \subset [\underline{\tau_j}, \overline{\tau_j}], j \in [1:m] \rightarrow$ integer programs
- Discretization of the constraints
  $\hat{\mathcal{T}}_i := \{t_i^{(1)}, \ldots, t_i^{(T_i)}\} \subset \mathcal{T},\ i \in [1:n_c], \rightarrow$ finite number of constraints
- Exploit linearity of constraints to reformulate discretized problems as binary linear programs

Approach

A two-stage approach to computing feasible solutions

First Stage

- Discretization of the decision variables
  $\hat{\mathcal{D}}_j := \{\tau_j^{(1)}, \ldots, \tau_j^{(N_j)}\} \subset [\underline{\tau_j}, \overline{\tau_j}], j \in [1:m] \rightarrow$ integer programs
- Discretization of the constraints
  $\hat{\mathcal{T}}_i := \{t_i^{(1)}, \ldots, t_i^{(T_i)}\} \subset \mathcal{T},\ i \in [1:n_c], \rightarrow$ finite number of constraints
- Exploit linearity of constraints to reformulate discretized problems as binary linear programs

Challenge is to

- manage problem size (don't want dense discretizations); and
- ensure discretizations are sufficiently rich to ensure outcome is continuous time feasible.

Approach

A two-stage approach to computing feasible solutions

Second Stage

- Restore the decision spaces to the continuous intervals $\tau_j \in [\underline{\tau}_j, \overline{\tau}_j]$;
- Use a sequential quadratic programming (SQP) method to locally improve cost;

Approach

A two-stage approach to computing feasible solutions

Second Stage

- Restore the decision spaces to the continuous intervals $\tau_j \in [\underline{\tau}_j, \overline{\tau}_j]$;
- Use a sequential quadratic programming (SQP) method to locally improve cost;

Challenge is to

- ensure continuous-time feasibility;
- ensure algorithm terminates finitely; and
- the final schedule is “better” (or no worse) than initial schedule

Outcome

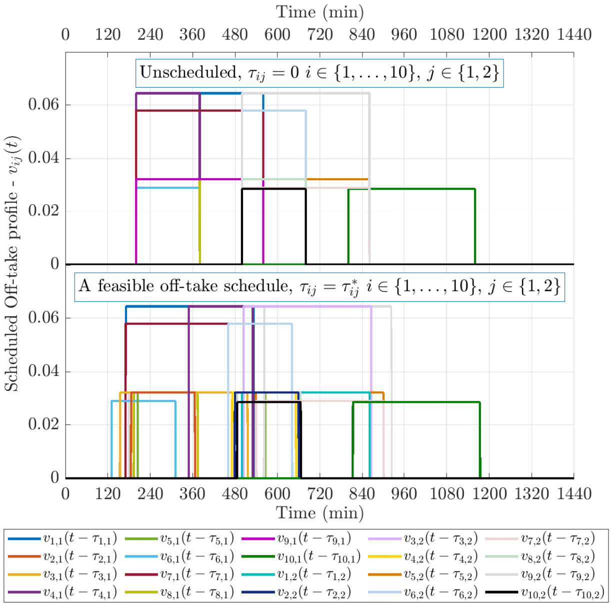
Main Result

Algorithm terminates after a finite number of steps with schedule that

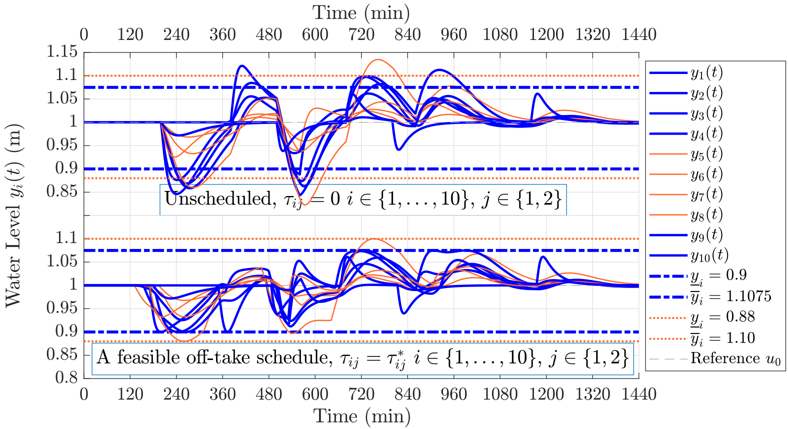
- is feasible for original problem
- within a tolerance of optimal for the restricted discretized decision space problem

Method appears to be tractable for realistic size problems.

Numerical Example

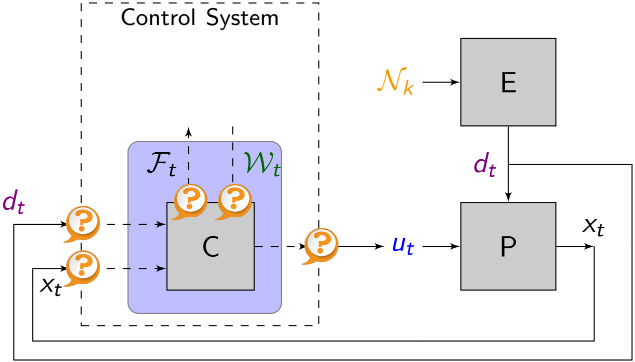


- 10 pools, 2 users per pool
- 24 hour horizon, 3hr scheduling intervals
- 24^{20} possible combinations in discretized version
- $\approx$ 20 minutes for algorithm termination



A. Lang, M. Cantoni, F. Farokhi, and I. Shames, Rigid-profile input scheduling under constrained dynamics with a water network application In *IEEE Transactions on Control Systems Technology*, [Early release pp.1-16, 2020].

Recall control hierarchy



For the synthesis of controller C the dynamical system P is modelled in discrete-time, i.e., $t \in \mathbb{N}$.

Problem statement

Constrained control with uncertain disturbance preview problem

Synthesize a controller C that has:

Inputs:

- $x_t \in \mathbb{R}^n$ current state;
- $d_t \in \mathbb{R}^M$ current disturbance input;
- $\mathcal{W}_t$ model of future disturbances;

Outputs:

- $u_t \in \mathbb{R}^m$ current control input;
- $\mathcal{F}_t$ signal to scheduler S (to be determined).

The controller should achieve the following:

- state and input constraint satisfaction over infinite horizon $\mathcal{T} = \mathbb{N}$
- "good" performance

Approach

Use robust receding horizon optimal control (a.k.a. robust MPC)

Aim: To achieve constraint satisfaction for dynamics of P over infinite horizon, i.e., recursive feasibility of the robust MPC problem to solve at each time.

Challenge: How to restrict what the scheduler can provide as uncertain preview of future disturbances to achieve recursive feasibility?

Approach

Use robust receding horizon optimal control (a.k.a. robust MPC)

Aim: To achieve constraint satisfaction for dynamics of P over infinite horizon, i.e., recursive feasibility of the robust MPC problem to solve at each time.

Challenge: How to restrict what the scheduler can provide as uncertain preview of future disturbances to achieve recursive feasibility?

This is the role of the signal $\mathcal{F}_t$*!*

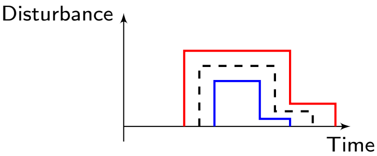
What is the preview model?

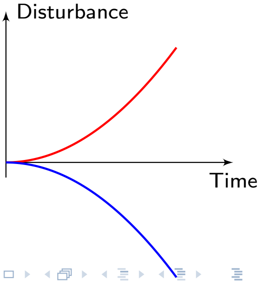
- Exact value of current disturbance d_t (e.g., as measured by meter at supply point);
- $\mathcal{W}_t$, a $T-1$ cartesian product of disturbance sets across the prediction horizon:

$$d_{t:t+T-1} \in \{d_t\} \times \mathcal{W}_t \subset \mathbb{R}^M \times (\mathbb{R}^M)^{(T-1)}$$

Preview can be structured, time-varying, with time correlation across the prediction horizon

Examples:



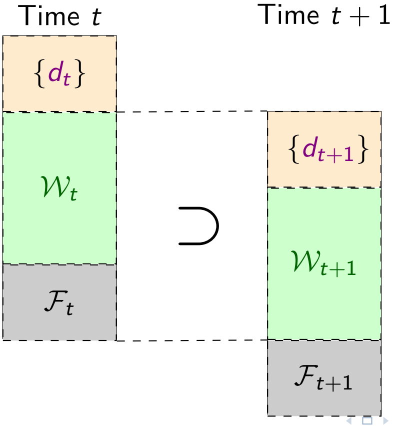


How to coordinate preview generation and control?

Uncertainty can only increase at end of horizon via $\mathcal{F}_t$, i.e.,

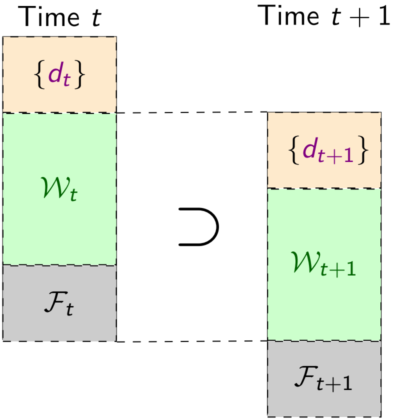
$$d_{t+T} \in \mathcal{F}_t \subset \mathbb{R}^M$$

Preview consistency condition:



Scheduler Assumption

Given end-of-preview uncertainty set $\mathcal{F}_t$, at each time $t \in \mathbb{N}$, the scheduler S determines a preview update $\mathcal{W}_{t+1}$ such that the following *consistency condition* holds.



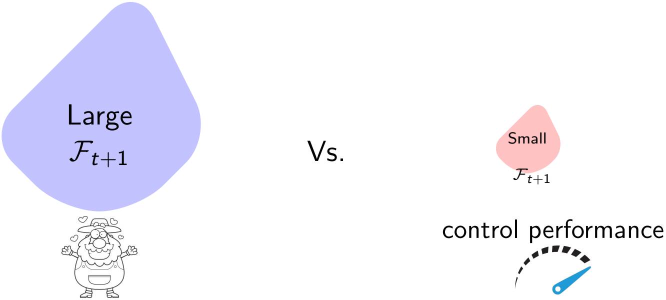
Challenges and questions

End-of-preview set $\mathcal{F}_{t+1}$ unconstrained in the *preview consistency condition*.

What is a desirable end-of-preview set?

End-of-preview set $\mathcal{F}_{t+1}$ unconstrained in the *preview consistency condition*.

What is a desirable end-of-preview set?



Conflicting objectives $\Rightarrow$ *a trade-off*

How can this trade-off be managed?

Model for dynamical system and form of controller

Dynamics of P modeled by

$$(\forall t \in \mathbb{N})\ x_{t+1} = Ax_t + Bu_t + Gd_t,$$

given initial state $x_0 = \xi \in \mathbb{R}^n$, model data $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$ and $G \in \mathbb{R}^{n\times M}$.

Model for dynamical system and form of controller

Dynamics of P modeled by

$$(\forall t \in \mathbb{N})\ x_{t+1} = Ax_t + Bu_t + Gd_t,$$

given initial state $x_0 = \xi \in \mathbb{R}^n$, model data $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$ and $G \in \mathbb{R}^{n\times M}$.

Controller asserts

$$u_t = \Bbbk_t(x_t, \{d_t\} \times \mathcal{W}_t) \text{ for } \ t \in \mathbb{N},$$

where $\Bbbk = (\Bbbk_t)_{t\in\mathbb{N}}$ is a control law sequence to be determined, with $\Bbbk_t : \mathrm{dom}(\Bbbk_t) \subset \mathbb{R}^n \times 2^{(\mathbb{R}^M)^T} \to \mathbb{R}^m$.

Constraints

Given constraint data $((C_t, D_t, H_t, c_t))_{t\in\mathbb{N}}$ with

- $C_t \in \mathbb{R}^{p\times n}$
- $D_t \in \mathbb{R}^{p\times m}$
- $H_t \in \mathbb{R}^{p\times M}$, and
- $c_t \in \mathbb{R}^{p}$.

the constraint set at time $t \in \mathbb{N}$ is defined as

$$\mathcal{Y}_t := \left\{(\chi, \upsilon, \delta) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^M \;\middle|\; C_t\chi + D_t\upsilon + H_t\delta \le c_t\right\}.$$

Constraints can be time-varying!

Problem

Control synthesis problem

The recursive synthesis of control law sequence $\mathbb{k}$ and end-of-preview sets $(\mathcal{F}_t)_{t\in\mathbb{N}}$ such that under the scheduler assumption

$$(\forall t \in \mathcal{T} = \mathbb{N})\ (x_t, \mathbb{k}_t(x_t, \{d_t\} \times \mathcal{W}_t), d_t) \in \mathcal{Y}_t,$$

where

$$(\forall t \in \mathbb{N})\ x_t = A^t\xi + \sum_{k\in[0:t-1]} A^{t-1-k}(B\mathbb{k}_k(x_k, \{d_k\} \times \mathcal{W}_k) + Gd_k).$$

Building MPC problem

Abstractly, the optimal control problem to solve at time $t \in \mathbb{N}$ is given by

$$\mathfrak{P}_t(x_t, \{d_t\} \times \mathcal{W}_t): \quad \inf \left\{ J_t(\mathbb{p}, \mathcal{Z}_t) \;\middle|\; (\mathbb{p}, \mathcal{Z}_t) \in \mathcal{P}_t(x_t, \{d_t\} \times \mathcal{W}_t) \right\},$$

where

- $\mathbb{p} = (\mathbb{p}_k)_{k \in [0:T-1]}$ are feedback policies, with $\mathbb{p}_k : \mathrm{dom}(\mathbb{p}_k) \subset \mathbb{R}^n \times (\mathbb{R}^M)^{k+1} \to \mathbb{R}^m$ i.e., function of initial state and past and current disturbance inputs.
- $\mathcal{Z}_t \subset \mathbb{R}^n$ is a terminal constraint for the state to be selected/adjusted *online*.
- $\mathcal{P}_t(x_t, \{d_t\} \times \mathcal{W}_t)$ encodes the state dynamics and constraints for the given set of possible disturbances across the prediction horizon and the admissible terminal constraints.
- J_t is a cost.

Building MPC problem

Abstractly, the optimal control problem to solve at time $t \in \mathbb{N}$ is given by

$$\mathfrak{P}_t(x_t, \{d_t\} \times \mathcal{W}_t) : \quad \inf \left\{ J_t(\mathbb{p}, \mathcal{Z}_t) \;\middle|\; (\mathbb{p}, \mathcal{Z}_t) \in \mathcal{P}_t(x_t, \{d_t\} \times \mathcal{W}_t) \right\},$$

where

- $\mathbb{p} = (\mathbb{p}_k)_{k \in [0:T-1]}$ are feedback policies, with $\mathbb{p}_k : \mathrm{dom}(\mathbb{p}_k) \subset \mathbb{R}^n \times (\mathbb{R}^M)^{k+1} \to \mathbb{R}^m$ i.e., function of initial state and past and current disturbance inputs.
- $\mathcal{Z}_t \subset \mathbb{R}^n$ is a terminal constraint for the state to be selected/adjusted *online*.
- $\mathcal{P}_t(x_t, \{d_t\} \times \mathcal{W}_t)$ encodes the state dynamics and constraints for the given set of possible disturbances across the prediction horizon and the admissible terminal constraints.
- J_t is a cost. *Not the focus of this work.*

MPC scheme

How does the controller work?

1. When $\mathcal{P}_t(x_t, \{d_t\} \times \mathcal{W}_t) \neq \emptyset$, select any $(\mathbb{p}^*, \mathcal{Z}_t^*) \in \mathcal{P}_t(x_t, \{d_t\} \times \mathcal{W}_t)$

MPC scheme

How does the controller work?

1. When $\mathcal{P}_t(x_t, \{d_t\} \times \mathcal{W}_t) \neq \emptyset$, select any $(\mathbb{p}^*, \mathcal{Z}_t^*) \in \mathcal{P}_t(x_t, \{d_t\} \times \mathcal{W}_t)$
2. Assert control input as

$$u_t = \mathbb{k}_t(x_t, \{d_t\} \times \mathcal{W}_t) = \mathbb{p}_0^*(x_t, d_t).$$

Since $\mathbb{p}^*$ is feasible then

$$(x_t, u_t, d_t) \in \mathcal{Y}_t$$

MPC scheme

How does the controller work?

1. When $\mathcal{P}_t(x_t, \{d_t\} \times \mathcal{W}_t) \neq \emptyset$, select any $(\mathbb{p}^*, \mathcal{Z}_t^*) \in \mathcal{P}_t(x_t, \{d_t\} \times \mathcal{W}_t)$
2. Assert control input as
$$u_t = \mathbb{k}_t(x_t, \{d_t\} \times \mathcal{W}_t) = \mathbb{p}_0^*(x_t, d_t).$$
Since $\mathbb{p}^*$ is feasible then
$$(x_t, u_t, d_t) \in \mathcal{Y}_t$$
3. Use $\mathcal{Z}_t^*$ to inform selection of the end-of-preview set $\mathcal{F}_t$

MPC scheme

How does the controller work?

1. When $\mathcal{P}_t(x_t, \{d_t\} \times \mathcal{W}_t) \neq \emptyset$, select any $(\mathbb{p}^*, \mathcal{Z}_t^*) \in \mathcal{P}_t(x_t, \{d_t\} \times \mathcal{W}_t)$

2. Assert control input as

$$u_t = \mathbb{k}_t(x_t, \{d_t\} \times \mathcal{W}_t) = \mathbb{p}_0^*(x_t, d_t).$$

Since $\mathbb{p}^*$ is feasible then

$$(x_t, u_t, d_t) \in \mathcal{Y}_t$$

3. Use $\mathcal{Z}_t^*$ to inform selection of the end-of-preview set $\mathcal{F}_t$

4. Repeat at each time $t \in \mathbb{N}$

MPC scheme

How does the controller work?

1. When $\mathcal{P}_t(x_t, \{d_t\} \times \mathcal{W}_t) \neq \emptyset$, select any $(\mathbb{p}^*, \mathcal{Z}_t^*) \in \mathcal{P}_t(x_t, \{d_t\} \times \mathcal{W}_t)$
2. Assert control input as
$$u_t = \mathbb{k}_t(x_t, \{d_t\} \times \mathcal{W}_t) = \mathbb{p}_0^*(x_t, d_t).$$
Since $\mathbb{p}^*$ is feasible then
$$(x_t, u_t, d_t) \in \mathcal{Y}_t$$
3. Use $\mathcal{Z}_t^*$ to inform selection of the end-of-preview set $\mathcal{F}_t$
4. Repeat at each time $t \in \mathbb{N}$

QUESTION: How can we ensure that $\mathcal{P}_{t+1}(x_{t+1}, \{d_{t+1}\} \times \mathcal{W}_{t+1}) \neq \emptyset$)?

MPC scheme

Goal: Recursive feasibility $\Rightarrow$ closed-loop constraint satisfaction for infinite horizon provided $\mathcal{P}_0(x_0, \{d_0\} \times \mathcal{W}_0) \neq \emptyset$, i.e.,

$$\underbrace{(\mathcal{P}_t \neq \emptyset \Rightarrow \mathcal{P}_{t+1} \neq \emptyset)}_{\text{recursive feasibility}} \Rightarrow \left(\mathcal{P}_0 \neq \emptyset \Rightarrow \underbrace{\left((\forall t \in \mathbb{N})\ (x_t, u_t, d_t) \in \mathcal{Y}_t \right)}_{\text{constraint satisfaction}} \right)$$

QUESTIONS:

- How to make the problem recursively feasible?
- How to select $\mathcal{F}_t$ using $\mathcal{Z}_t^*$?

MPC scheme

Goal: Recursive feasibility $\Rightarrow$ closed-loop constraint satisfaction for infinite horizon provided $\mathcal{P}_0(x_0, \{d_0\} \times \mathcal{W}_0) \neq \emptyset$, i.e.,

$$\underbrace{(\mathcal{P}_t \neq \emptyset \Rightarrow \mathcal{P}_{t+1} \neq \emptyset)}_{\text{recursive feasibility}} \Rightarrow \left(\mathcal{P}_0 \neq \emptyset \Rightarrow \underbrace{\left((\forall t \in \mathbb{N})\ (x_t, u_t, d_t) \in \mathcal{Y}_t \right)}_{\text{constraint satisfaction}}\ \right)$$

QUESTIONS:

- How to make the problem recursively feasible? *Restrict the choice of terminal set using admissible set of terminal sets $\mathcal{T}_t$, i.e., $\mathcal{Z}_t \in \mathcal{T}_t$*
- How to select $\mathcal{F}_t$ using $\mathcal{Z}_t^*$? *Link the $\mathcal{F}_t$ to the admissible terminal set as part of ensuring recursive feasibility*

Recursive feasibility condition

QUESTION: What property should $\mathcal{T}_t$ satisfy jointly with $\mathcal{F}_t$ such that problem is recursively feasible?

Recursive feasibility for abstract MPC problem

Given $(\mathbb{p}, \mathcal{Z}_t) \in \mathcal{P}_t(x_t, \{d_t\} \times \mathcal{W}_t)$, $\mathcal{F}_t \subset \mathbb{R}^M$, and $\mathcal{T}_{t+1} \subset 2^{\mathbb{R}^n}$, if

$$\Big(\exists \mathcal{Z}_{t+1} \in \mathcal{T}_{t+1} \;\; (\forall (\zeta, \omega) \in \mathcal{Z}_t \times \mathcal{F}_t)(\exists \alpha \in \mathbb{R}^m) \\ (\zeta, \alpha, \omega) \in \mathcal{Y}_{t+T} \quad \text{and} \quad (A\zeta + B\alpha + G\omega) \in \mathcal{Z}_{t+1},$$

then provided scheduler assumption holds for given $\mathcal{F}_t$ the MPC problem is recursive feasible, i.e., $\mathcal{P}_{t+1}(x_{t+1}, \{d_{t+1}\} \times \mathcal{W}_{t+1}) \neq \emptyset$ for $x_{t+1} = Ax_t + B\mathbb{p}_0(x_t, d_t) + Gd_t$.

Recursive feasibility condition

Recursive feasibility for abstract MPC problem

Given $(\mathbb{p}, \mathcal{Z}_t) \in \mathcal{P}_t(x_t, \{d_t\} \times \mathcal{W}_t)$, $\mathcal{F}_t \subset \mathbb{R}^M$, and $\mathcal{T}_{t+1} \subset 2^{\mathbb{R}^n}$, if

$$\Big(\exists \mathcal{Z}_{t+1} \in \mathcal{T}_{t+1} \ (\forall (\zeta, \omega) \in \mathcal{Z}_t \times \mathcal{F}_t)(\exists \alpha \in \mathbb{R}^m) \\ (\zeta, \alpha, \omega) \in \mathcal{Y}_{t+T} \quad \text{and} \quad (A\zeta + B\alpha + G\omega) \in \mathcal{Z}_{t+1},$$

then provided scheduler assumption holds for given $\mathcal{F}_t$ the MPC problem is recursive feasible, i.e., $\mathcal{P}_{t+1}(x_{t+1}, \{d_{t+1}\} \times \mathcal{W}_{t+1}) \neq \emptyset$ for $x_{t+1} = Ax_t + B\mathbb{p}_0(x_t, d_t) + Gd_t$.

QUESTION: How to design $(\mathcal{T}_t)_{t \in \mathbb{N}}$ so that each $\mathcal{F}_t$ can be constructed to meet condition above?

Static set approach

Fix $\mathcal{T}_t = \{\bar{\mathcal{Z}}\}$ and $\mathcal{F}_t = \bar{\mathcal{F}}$ for all $t \in \mathbb{N}$. If

$$(\forall(\zeta,\omega) \in \bar{\mathcal{Z}} \times \bar{\mathcal{F}})(\exists\alpha \in \mathbb{R}^m)\ (\zeta,\alpha,\omega) \in \bigcap_{k=0}^{\infty} \mathcal{Y}_k \text{ and } (A\zeta + B\alpha + G\omega) \in \bar{\mathcal{Z}},$$

then recursive feasibility condition is satisfied.

Static set approach

Fix $\mathcal{T}_t = \{\bar{\mathcal{Z}}\}$ and $\mathcal{F}_t = \bar{\mathcal{F}}$ for all $t \in \mathbb{N}$. If

$$(\forall(\zeta,\omega) \in \bar{\mathcal{Z}} \times \bar{\mathcal{F}})(\exists \alpha \in \mathbb{R}^m)\ (\zeta,\alpha,\omega) \in \bigcap_{k=0}^{\infty} \mathcal{Y}_k \text{ and } (A\zeta + B\alpha + G\omega) \in \bar{\mathcal{Z}},$$

then recursive feasibility condition is satisfied.

Advantages:

- finding $\bar{\mathcal{Z}}$ and $\bar{\mathcal{F}}$ is almost a standard control-invariance problem
- $\bar{\mathcal{Z}}$ and $\bar{\mathcal{F}}$ can be calculated offline meaning online-optimization problem may be simpler.

Limitations:

- strict restrictions on how constraints can change, e.g., requires $\bigcap_{k=0}^{\infty} \mathcal{Y}_k \neq \emptyset$
- may lead to very conservative performance both for control and scheduler

An adjustable terminal ingredient approach

Idea: Allow terminal set to be adjusted online by parametrizing the set of admissible terminal sets $\mathcal{T}_t$ in terms of additional decision variables. Builds on ideas from literature on MPC reference tracking.

Approach:

1. Determine some fixed sets $\mathring{\mathcal{Z}} \subset \mathbb{R}^n$ and $\mathring{\mathcal{F}} \subset \mathbb{R}^M$ that satisfy a control-invariance condition.

An adjustable terminal ingredient approach

Idea: Allow terminal set to be adjusted online by parametrizing the set of admissible terminal sets $\mathcal{T}_t$ in terms of additional decision variables. Builds on ideas from literature on MPC reference tracking.

Approach:

1. Determine some fixed sets $\mathring{\mathcal{Z}} \subset \mathbb{R}^n$ and $\mathring{\mathcal{F}} \subset \mathbb{R}^M$ that satisfy a control-invariance condition.
2. Introduce additional decision variables:
   - scaling parameter $\sigma \in \mathbb{R}_{\geq 0}$;
   - artificial reference trajectory; and
   - constraint restriction level.

An adjustable terminal ingredient approach

Idea: Allow terminal set to be adjusted online by parametrizing the set of admissible terminal sets $\mathcal{T}_t$ in terms of additional decision variables. Builds on ideas from literature on MPC reference tracking.

Approach:

1. Determine some fixed sets $\mathring{\mathcal{Z}} \subset \mathbb{R}^n$ and $\mathring{\mathcal{F}} \subset \mathbb{R}^M$ that satisfy a control-invariance condition.

2. Introduce additional decision variables:
   - scaling parameter $\sigma \in \mathbb{R}_{\geq 0}$;
   - artificial reference trajectory; and
   - constraint restriction level.

3. Parameterize $\mathcal{T}_t$ as scaled and shifted versions of the set $\mathring{\mathcal{Z}}$.

An adjustable terminal ingredient approach

Idea: Allow terminal set to be adjusted online by parametrizing the set of admissible terminal sets $\mathcal{T}_t$ in terms of additional decision variables. Builds on ideas from literature on MPC reference tracking.

Approach:

1. Determine some fixed sets $\mathring{\mathcal{Z}} \subset \mathbb{R}^n$ and $\mathring{\mathcal{F}} \subset \mathbb{R}^M$ that satisfy a control-invariance condition.
2. Introduce additional decision variables:
   - scaling parameter $\sigma \in \mathbb{R}_{\geq 0}$;
   - artificial reference trajectory; and
   - constraint restriction level.
3. Parameterize $\mathcal{T}_t$ as scaled and shifted versions of the set $\mathring{\mathcal{Z}}$.
4. Select $\mathcal{F}_t$ as the scaled and shifted $\mathring{\mathcal{F}}$ corresponding to the selection of $\mathcal{Z}_t^* \in \mathcal{T}_t$, i.e., $\mathcal{F}_t = \{\omega + \sigma\mathring{\omega} \mid \mathring{\omega} \in \mathring{\mathcal{F}}\}$

How to make it recursively feasible?

Ensure the artificial reference trajectory satisfies the following:

- it is consistent with the model of the dynamics;
- the trajectory is eventually periodic;
- the periodic part of trajectory is "sufficiently inside the constraints"

How to make it recursively feasible?

Ensure the artificial reference trajectory satisfies the following:

- it is consistent with the model of the dynamics;
- the trajectory is eventually periodic;
- the periodic part of trajectory is "sufficiently inside the constraints"

Recursive feasibility result

Assume the scheduling assumption holds with $\mathcal{F}_t$ equal to scaled and shifted static set $\mathring{\mathcal{F}}$ and the constraint set from one time instant to next does not "change too much". Then the adjustable terminal ingredient approach is recursively feasible.

Adjustable terminal ingredient approach

Advantages:

- less restrictions on the static sets $\mathring{\mathcal{Z}}$ and $\mathring{\mathcal{F}}$
- constraint restriction does not require $\bigcap_{k=0}^{\infty} \mathcal{Y}_k \neq \emptyset$
- additional decision variables can be used in cost to balance secondary objectives

Remaining challenge: Tractability

- $\mathbb{p}$ is an infinite dimensional decision variable; and
- infinite number of constraints
  - robust constraint condition, i.e., $\forall w \in \{d_t\} \times \mathcal{W}_t$ in the definition of $\mathcal{P}_t$
  - the "sufficiently inside the constraints" condition for introduced artificial reference variable

Towards a tractable problem

Approach:

1. Parameterize polices $\mathbb{p}$ to be affine

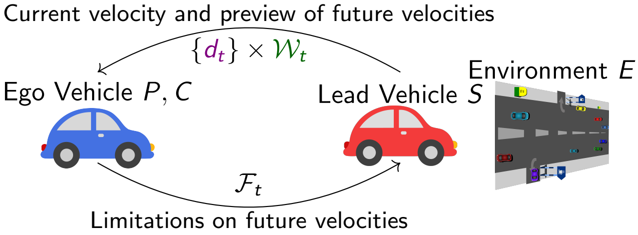
$$(\forall k \in [0:T-1])\ \mathbb{p}_k(x_t, w_{0:k}) = \mathscr{k}_k + \sum_{j \in [1:k]} \mathscr{K}_{k,j} w_j.$$

2. Restrict $\mathcal{W}_t$ to be from class of closed-convex cones, and static set $\mathring{\mathscr{Z}}$ be polytopic
3. Use a duality based robust-optimization trick (a la Ben-Tal, Ghaoui, Nemirovski) to replace infinite constraints with equivalent set of finite-dimensional ones.

Result: A finite-dimensional set of convex (conic) constraints that is equivalent to restricted version of original problem.

A Two-vehicle platoon numerical example

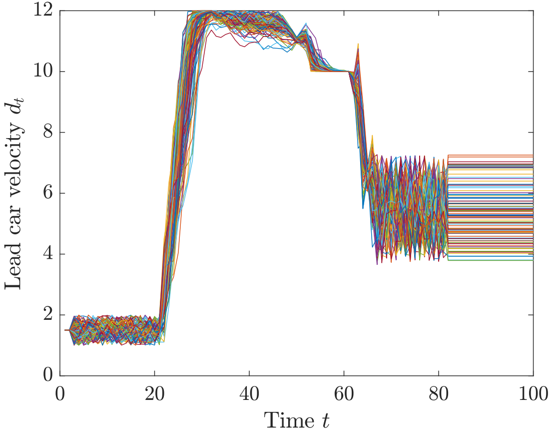
Two vehicle platoon



- control input u_t is acceleration to ego vehicle
- constraints on acceleration, relative distance, and relative velocity
- performance "goal" is to minimize distance between vehicles and provide lead vehicle flexibility to adapt to environment

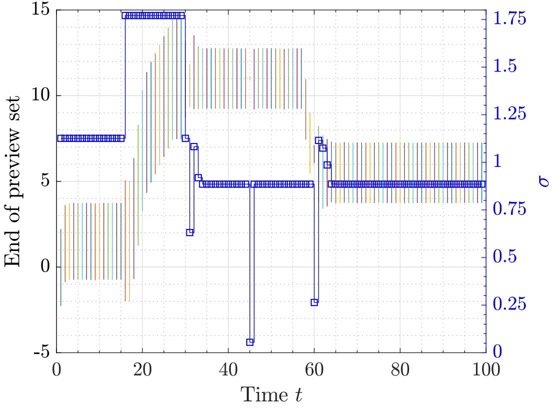
Results

100 realizations of the lead vehicles velocities.



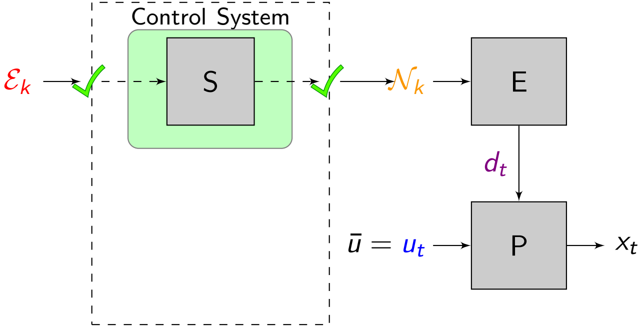
Flexibility provided to lead vehicle.

Low flexibility when constraints are restricted.



Conclusion

Considered two problems associated with synthesis of scheduler and controller in a control hierarchy.

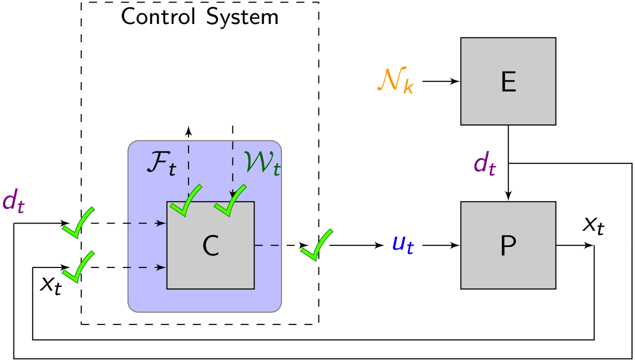


Problem 1: Rigid-profile input scheduling

Difficult scheduling problem (rigid-profiles and continuous dynamics). Two-stage method proposed using discretization to yeild "good" feasible solutions.

Conclusion

Considered two problems associated with synthesis of scheduler and controller in a control hierarchy.



Problem 2: Receding horizon control under uncertain preview.

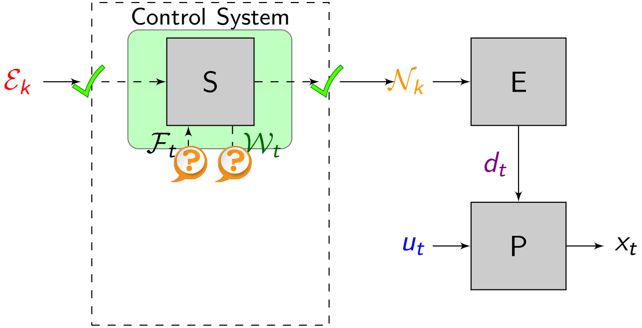
Difficult to guarantee feasibility with structured uncertain preview and time-varying constraints. Receding horizon control method with adjustable terminal ingredients proposed. It yields feasibility with flexibility in the choice of cost.

Future work

- Problem 1: explore improved methods for chosing restricted decision spaces to achieve better and faster results.
- Problem 2: explore how the cost can be chosen and updated across time to meet certain objectives.
- Problem 2: how to exploit structure within optimization solvers to enable scalability to large systems (especially cascade structure)
- Extend results to more general system dynamics

Future work

Explore problem of generating preview uncertainty sets within the context of problem 2



Thank-you for listening!