Control of large-scale systems with applications to water distribution and road traffic networks

Bart De Schutter, Andreas Hegyi, Rudy Negenborn, Solomon Zegeye, Lakshmi Baskar, Anna Sadowska

Delft Center for Systems and Control
Delft University of Technology

www.dcsc.tudelft.nl

TUDelft

Outline

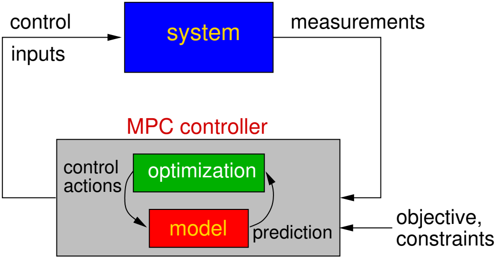
Model predictive control (MPC)

Features

- Very popular in process industry
- Model-based
- Easy to tune
- Multi-input multi-output (MIMO)
- Allows constraints on inputs and outputs
- Adaptive / receding horizon
- Uses off-line or on-line optimization

MPC: Principle of operation

- Performance/objective function (e.g., reference tracking versus input energy)
- Prediction model
- Constraints
- (On-line) optimization
- Receding horizon

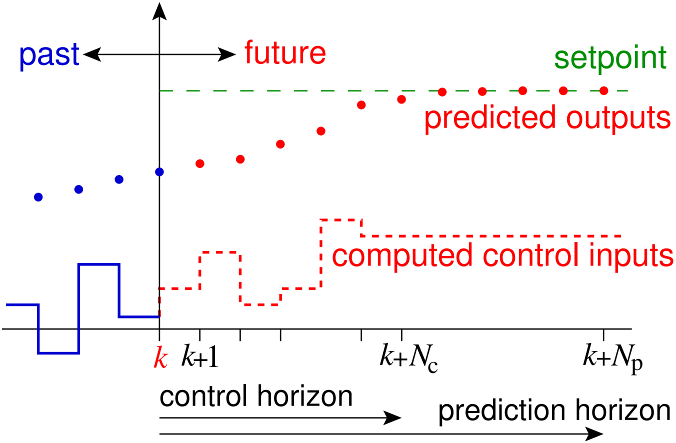


Nonlinear optimization problem: $\min_{\mathbf{u}_k} J_{k,N_\mathrm{p}}^{\mathrm{MPC}}(\mathbf{u}_k)$

subject to system dynamics, operational constraints

where $\mathbf{u}_k = [u^\mathrm{T}(k)\ u^\mathrm{T}(k+1)\ \cdots\ u^\mathrm{T}(k+N_\mathrm{p}-1)]^\mathrm{T}$

MPC: Receding horizon approach



Challenges in control of large-scale networks

- Large-scale nature of the system
- Distributed vs centralized control
- Optimality $\leftrightarrow$ computational efficiency/tractability
- Global $\leftrightarrow$ local
- Scalability
- Communication requirements (bandwidth)
- Robustness against failures

Challenges in MPC of large-scale networks

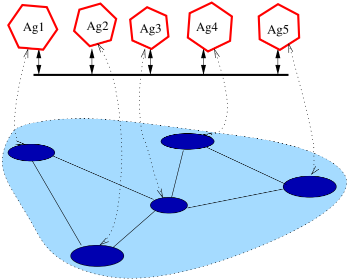
Major problem for MPC in practice:
In general: nonlinear, nonconvex optimization problem
$\rightarrow$ huge computation time, in particular for large-scale systems

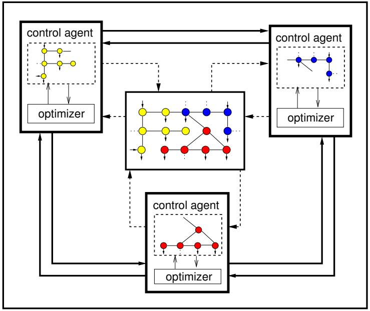
Solutions:

- Choice of the prediction model: accuracy versus computational complexity
- Use parametrized control laws
- Use distributed and/or multi-level approach
- Right optimization approach
  - parallel and/or distributed optimization
  - approximate original MPC optimization problem by another optimization problem that can be solved efficiently
- Include application-specific knowledge

Distributed MPC

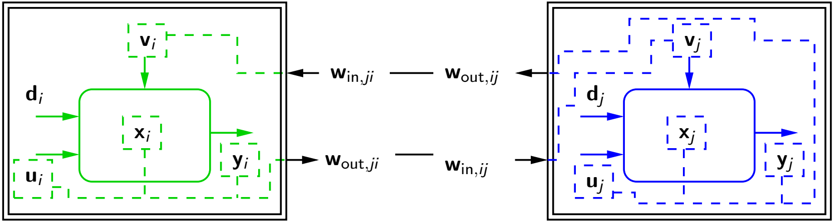
- Subsystems instead of overall system
- Single agent/controller for each subsystem
  - limited action capabilities
  - limited information gathering
- **Challenge**: agents should choose local inputs that are globally optimal





Distributed MPC

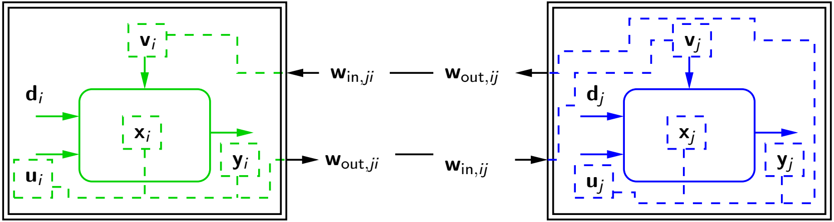
Interconnection between control agents



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

Distributed MPC

Interconnection between control agents



$$\mathbf{x}_i(k+1) = \mathbf{f}_i(\mathbf{x}_i(k), \mathbf{u}_i(k), \mathbf{d}_i(k), w_{\mathrm{in},j_1 i}(k), \ldots, w_{\mathrm{in},j_{m_i} i}(k))$$

$$\mathbf{w}_{\mathrm{out},ji}(k+1) = \mathbf{h}^{ji}_{\mathrm{out}}(\mathbf{u}_i(k), \mathbf{y}_i(k), \mathbf{x}_i(k+1)) \quad \text{for each neighbor } j \text{ of } i$$

Local MPC control problem of agent i at decision step k

$$\min_{\tilde{\mathbf{u}}_i(k),\tilde{\mathbf{x}}_i(k+1)} J_{\text{local},i}(\tilde{\mathbf{u}}_i(k), \tilde{\mathbf{x}}_i(k+1))$$

subject to

- subsystem dynamics: prediction model

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

$$\vdots$$

$$\mathbf{x}_i(k+N) = \mathbf{f}_i(\mathbf{x}_i(k+N-1), \mathbf{u}_i(k+N-1), \mathbf{d}_i(k+N-1), \dots)$$

- initial local state, disturbances, and additional constraints

Local MPC control problem of agent i at decision step k

$$\min_{\tilde{\mathbf{u}}_i(k),\tilde{\mathbf{x}}_i(k+1)} J_{\text{local},i}(\tilde{\mathbf{u}}_i(k),\tilde{\mathbf{x}}_i(k+1))$$

subject to

- subsystem dynamics: prediction model

$$\mathbf{x}_i(k+1) = \mathbf{f}_i(\mathbf{x}_i(k),\mathbf{u}_i(k),\mathbf{d}_i(k),\mathbf{w}_{\text{in},j_1 i}(k),\ldots,\mathbf{w}_{\text{in},j_{m_i} i}(k))$$

$$\mathbf{w}_{\text{out},ji}(k+1) = \mathbf{h}_{\text{out},ji}(\mathbf{u}_i(k),\mathbf{y}_i(k),\mathbf{x}_i(k+1)) \quad \text{for each neighbor } j \text{ of } i$$

$$\vdots$$

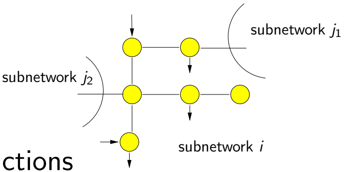
$$\mathbf{x}_i(k+N) = \mathbf{f}_i(\mathbf{x}_i(k+N-1),\mathbf{u}_i(k+N-1),\mathbf{d}_i(k+N-1), \mathbf{w}_{\text{in},j_1 i}(k+N-1),\ldots,\mathbf{w}_{\text{in},j_{m_i} i}(k+N-1))$$

$$\mathbf{w}_{\text{out},ji}(k+N) = \mathbf{h}_{\text{out},ji}(\mathbf{u}_i(k+N-1),\mathbf{y}_i(k+N-1),\mathbf{x}_i(k+N))$$

- initial local state, disturbances and additional constraints

Interconnecting constraints

- Constraints on interconnecting variables
- Imposed by dynamics of overall network
- *What goes in into i equals what goes out from j*
- Satisfaction necessary for accurate predictions



$$
\begin{aligned}
\mathbf{w}_{\text{in},ji}(k) &= \mathbf{w}_{\text{out},ij}(k)\\
\mathbf{w}_{\text{out},ji}(k) &= \mathbf{w}_{\text{in},ij}(k)\\
\vdots \quad & \quad \vdots\\
\mathbf{w}_{\text{in},ji}(k+N-1) &= \mathbf{w}_{\text{out},ij}(k+N-1)\\
\mathbf{w}_{\text{out},ji}(k+N-1) &= \mathbf{w}_{\text{in},ij}(k+N-1)
\end{aligned}
$$

For agent controlling subsystem i

- $\mathbf{w}_{\text{in},ij}$ and $\mathbf{w}_{\text{out},ij}$ of neighbor j unknown
- How to make accurate predictions?
  $\rightarrow$ via negotiations

Multiple-iterations scheme to agree on values of interconnecting variables

- Each agent
  - computes optimal local *and* interconnecting variables
  - communicates interconnecting variables to neighbors
  - updates parameters $\tilde{\lambda}_{\text{in}}^{ji}, \tilde{\lambda}_{\text{out}}^{ji}$ of additional cost term J_{inter}^{i}
- Iterations continue until stopping criterion satisfied
- Scheme converges to overall optimal solution under convexity assumptions

$$\min_{\tilde{\mathbf{u}}_i,\tilde{\mathbf{x}}_i,\tilde{\mathbf{w}}_{\text{in},li},\tilde{\mathbf{w}}_{\text{out},li}} J_{\text{local},i}(\tilde{\mathbf{u}}_i(k),\tilde{\mathbf{x}}_i(k+1)) + \sum_{j\in\text{neighbors}_i} J_{\text{inter},i}(\tilde{\mathbf{w}}_{\text{in},ji}(k),\tilde{\mathbf{w}}_{\text{out},ji}(k))$$

subject to

- dynamics of subsystem i over the horizon
- initial local state, disturbances, additional constraints

- Scheme based on augmented Lagrangian and block coordinate descent + serial implementation
- Additional objective function $J^{(s)}_{\text{inter},i}(\tilde{\mathbf{w}}_{\text{in},ji}(k), \tilde{\mathbf{w}}_{\text{out},ji}(k)) =$

$$\begin{bmatrix} \tilde{\boldsymbol{\lambda}}^{(s)}_{\text{in},ji}(k) \\ -\tilde{\boldsymbol{\lambda}}^{(s)}_{\text{out},ij}(k) \end{bmatrix}^{\mathsf{T}} \begin{bmatrix} \tilde{\mathbf{w}}_{\text{in},ji}(k) \\ \tilde{\mathbf{w}}_{\text{out},ji}(k) \end{bmatrix} + \frac{\gamma}{2} \left\| \begin{bmatrix} \tilde{\mathbf{w}}_{\text{in,prev},ij}(k) - \tilde{\mathbf{w}}_{\text{out},ji}(k) \\ \tilde{\mathbf{w}}_{\text{out,prev},ij}(k) - \tilde{\mathbf{w}}_{\text{in},ji}(k) \end{bmatrix} \right\|_2^2,$$

where for each j that is a neighbor that solved its problem before i in iteration s:

$$\tilde{\mathbf{w}}_{\text{in,prev},ij}(k) = \tilde{\mathbf{w}}^{(s)}_{\text{in},ij} \quad \text{and} \quad \tilde{\mathbf{w}}_{\text{out,prev},ij}(k) = \tilde{\mathbf{w}}^{(s)}_{\text{out},ij}$$

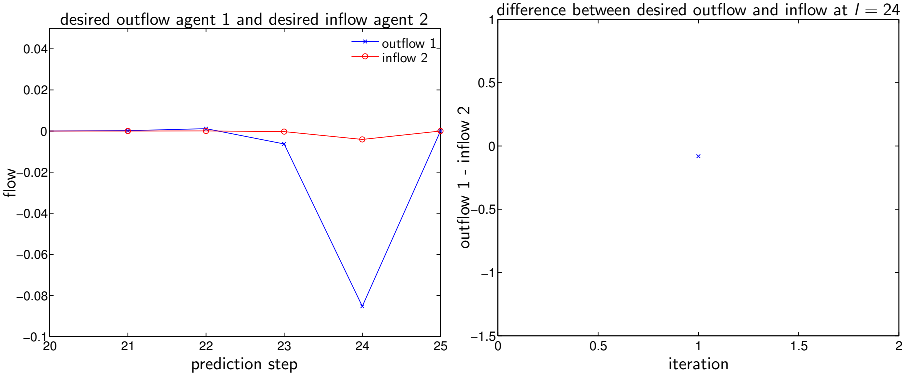
and where for each j that has not solved its problem in iteration s yet

$$\tilde{\mathbf{w}}_{\text{in,prev},ij}(k) = \tilde{\mathbf{w}}^{(s-1)}_{\text{in},ij} \quad \text{and} \quad \tilde{\mathbf{w}}_{\text{out,prev},ij}(k) = \tilde{\mathbf{w}}^{(s-1)}_{\text{out},ij}$$

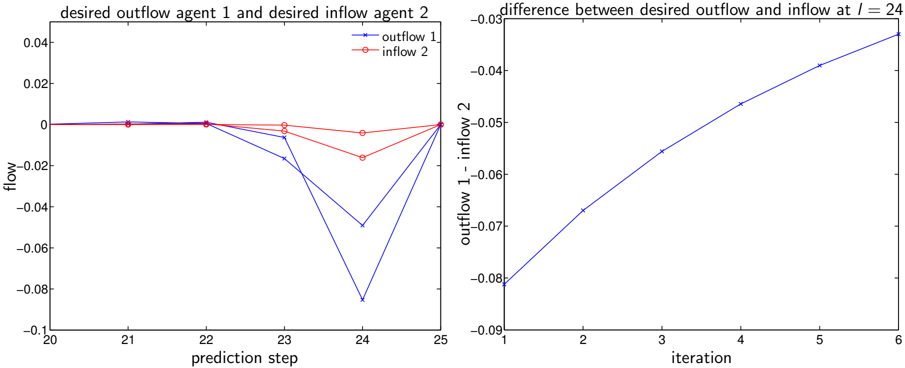
Multiple-iterations scheme (continued)

- Update of $\tilde{\boldsymbol{\lambda}}_{\text{in},ji}$:
  $\tilde{\boldsymbol{\lambda}}_{\text{in},ji}^{(s+1)}(k) = \tilde{\boldsymbol{\lambda}}_{\text{in},ji}^{(s)} + \gamma \left(\tilde{\mathbf{w}}_{\text{in},ji}^{(s)}(k) - \tilde{\mathbf{w}}_{\text{out},ij}^{(s)}(k) \right)$

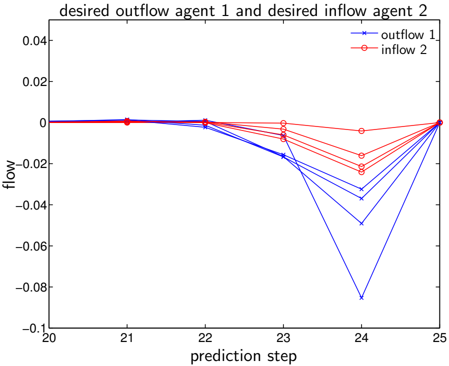
- Alternative: auxiliary problem principle with parallel implementation

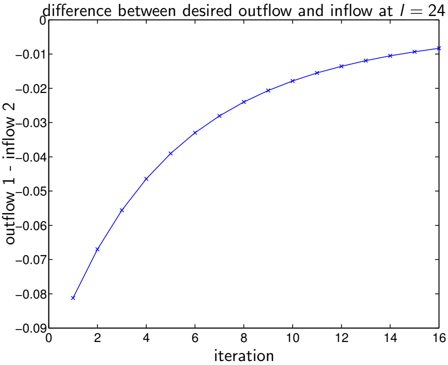


Obtaining agreement on flows between two subsystems

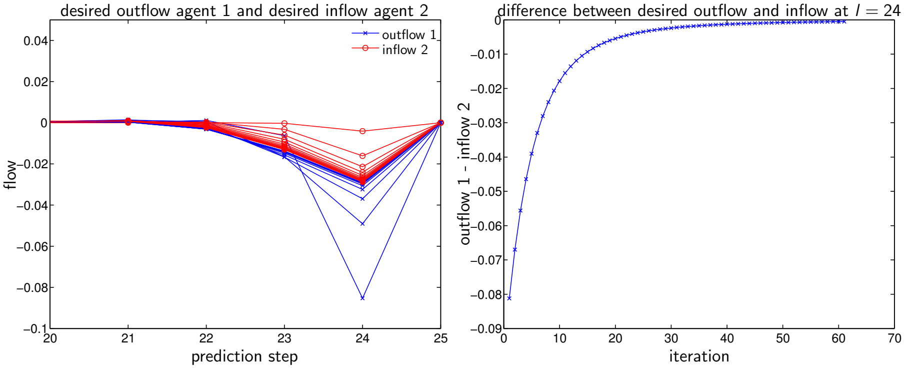


Obtaining agreement on flows between two subsystems





Obtaining agreement on flows between two subsystems



Obtaining agreement on flows between two subsystems

Multiple-iterations scheme

- Main problem with augmented Lagrangian approach + family:
  - convergence + convergence speed
  - feasibility issues in case of finite termination
  - extension to for nonlinear, nonconvex case

- Ongoing research in field is still very active and also explores alternative approaches:
  - agent-based coordination & consensus methods
  - game-based methods
  - swarm intelligence methods

Cooperative water control

Cooperative water control

Cooperative water control

Cooperation to
improve performance

Irrigation canals

Irrigation accounts for about 70% of global fresh water usage

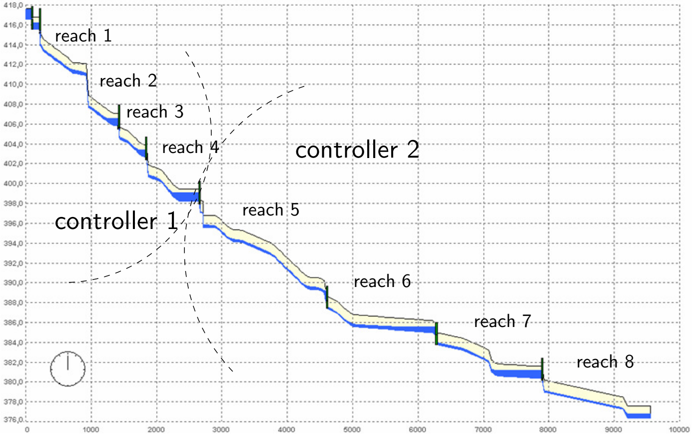
Irrigation canals should deliver water at the right time to the right location

Components:

- control structures
- off-takes
- canal reaches
- water users

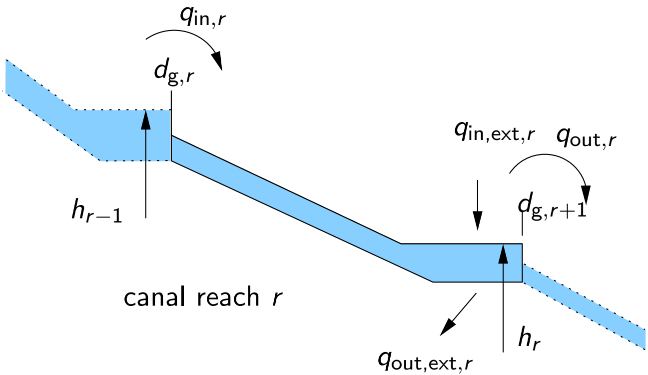
Irrigation canals – Case study

Case study: West-M canal, south of Phoenix, Arizona, 10 km long



Adjust gates to maintain water levels, while satisfying demand and actuator constraints.

Dynamics of a canal reach



Dynamics of a canal reach

Various ways to model canal reach: from accurate and slow to approximate and fast

- Saint Venant equations

$$\frac{\partial Q}{\partial x} + \frac{\partial A}{\partial t} = q_{\mathrm{lat}}$$
$$\frac{\partial Q}{\partial t} + \frac{\partial}{\partial x}\left(\frac{Q}{A}\right)^2 + gA\frac{\partial h}{\partial x} + \frac{gQ|Q|}{C^2 RA} = 0$$

  with Q flow, A cross-section area, q_{lat} lateral inflow, h water height
  $\rightarrow$ system of nonlinear differential equations
- Discretization of Saint Venant equations in time and space
  $\rightarrow$ system of nonlinear difference equations

Dynamics of a canal reach

Various ways to model canal reach: from accurate and slow to approximate and fast

- Saint Venant equations $\rightarrow$ system of nonlinear differential equations
- Discretization of Saint Venant equations in time and space $\rightarrow$ system of nonlinear difference equations
- Linearization
  $\rightarrow$ system of linear difference equations
- If spatial discretization step is equal to reach length, we get simple time-delay equation:
  inflow of reach influences water height at end after given constant delay

Dynamics of a canal reach

$$h_r(k+1) = h_r(k) + \frac{T_\mathrm{c}}{c_r} q_{\mathrm{in},r}(k - k_{\mathrm{d},r}) - \frac{T_\mathrm{c}}{c_r} q_{\mathrm{out},r}(k) + \frac{T_\mathrm{c}}{c_r} q_{\mathrm{ext,in},r}(k) - \frac{T_\mathrm{c}}{c_r} q_{\mathrm{ext,out},r}(k)$$

$$q_{\mathrm{in},r}(k) = q_{\mathrm{in},r}(k-1) + C_{e,r} \Delta h_{r-1}(k) + C_{u,r} \Delta d_{\mathrm{g},r}(k)$$

$$q_{\mathrm{out},r}(k) = q_{\mathrm{out},r}(k-1) + C_{e,r+1} \Delta h_r(k) + C_{u,r+1} \Delta d_{\mathrm{g},r+1}(k)$$

with constant

$$C_{e,r} = \frac{g c_{\mathrm{w},r} W_{\mathrm{s},r} \mu_r \underline{d}_{\mathrm{g},r}}{\sqrt{2g(\underline{h}_{r-1} - (z_{\mathrm{s},r} + \mu_r \underline{d}_{\mathrm{g},r}))}}$$

$$C_{u,r} = c_{\mathrm{w},r} W_{\mathrm{s},r} \mu_r \sqrt{2g(\underline{h}_{r-1} - (z_{\mathrm{s},r} + \mu_r \underline{d}_{\mathrm{g},r}))} - \frac{g c_{\mathrm{w},r} W_{\mathrm{s},r} \mu_r^2 \underline{d}_{\mathrm{g},r}}{\sqrt{2g(\underline{h}_{r-1} - (z_{\mathrm{s},r} + \mu_r \underline{d}_{\mathrm{g},r}))}},$$

where $\underline{h}, \underline{d}$ are given linearization points

Control of an irrigation canal

Control objectives

- Minimize deviations of water levels from set-points
- Minimize changes in gate positions

$$J_{\mathrm{local},i} = \sum_{l=0}^{N_{\mathrm{p}}-1} \sum_{r \in \mathcal{R}_i} \Big(\alpha_r \left(h_r(k+1+l) - h_{r,\mathrm{ref}} \right)^2 + \beta_r \left(d_{\mathrm{g},r}(k+l) - d_{\mathrm{g},r}(k+l-1) \right)^2 \Big)$$

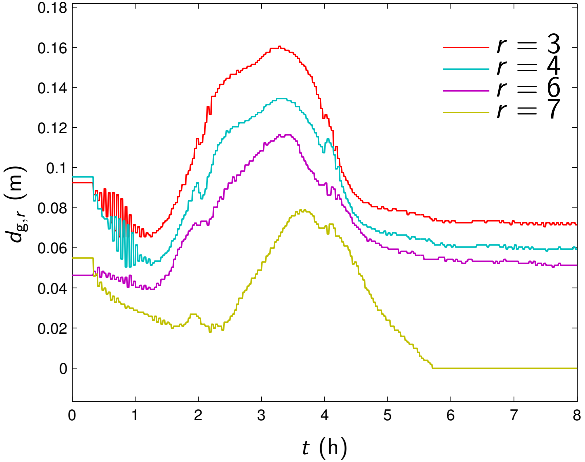
Constraints

- maximum on the change in the gate position, both upwards and downwards
- gate position should always be positive
- gate should not be lifted out of the water

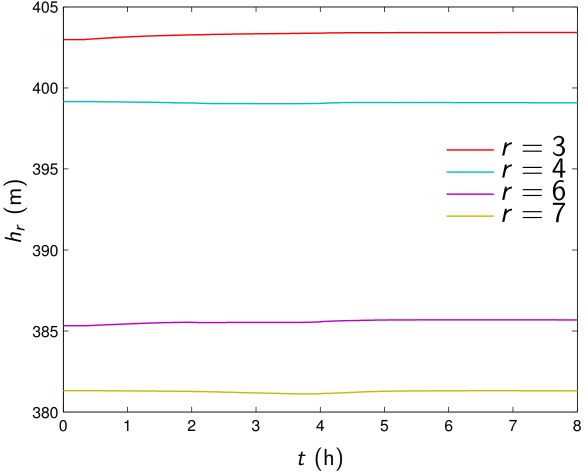
Setup

- Implementation
  - Nonlinear, validated model of the canal implemented in SOBEK
  - MPC controllers with linearized models implemented in Matlab
  - Optimization using CPLEX v10.0 through Tomlab 5.7 interface
- Parameters
  - $T_{\mathrm{c}} = 120\,\mathrm{s}$, $N = 30\,\mathrm{steps}$
  - Distributed MPC scheme parameters: $\gamma = 1000$, $\varepsilon = 1.10^{-4}$
  - Cost coefficients: $\alpha_r = 0.15$, $\beta_r = 0.0075$
- Scenario
  - 8 hour simulation
  - at $t = 2$: increase of 0.1 m^3/s in offtake of reach 3
  - at $t = 4$: decrease of 0.1 m^3/s in offtake of reach 3

Evolution of control actions over the full simulation

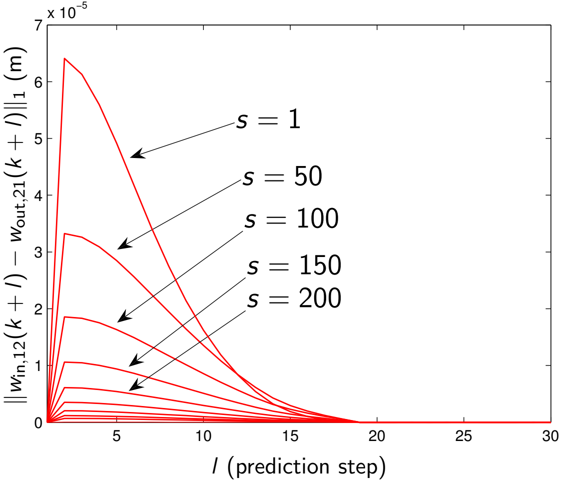


Evolution of water levels over the full simulation

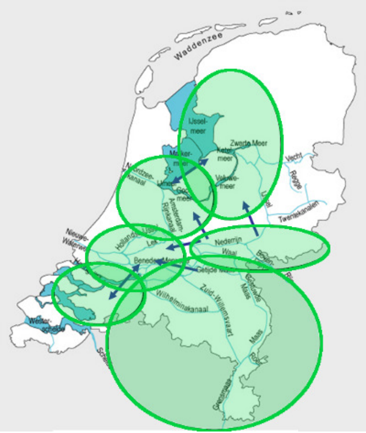


performance within 10% of centralized controller

Evolution of absolute error over the iterations at $t = 2.23$

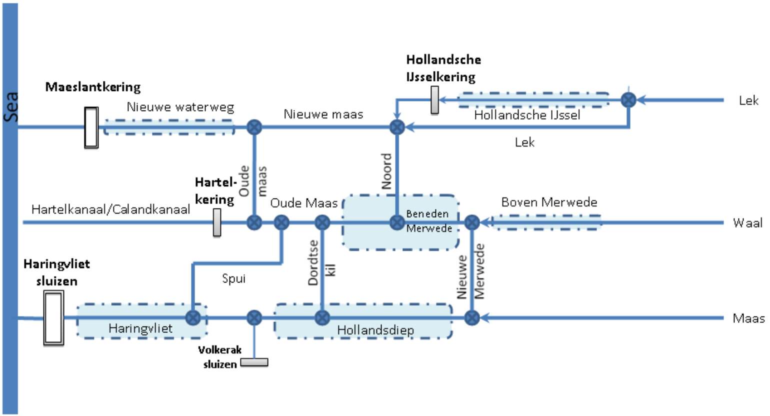


Dutch river system



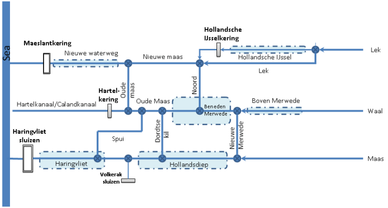
Dutch river system

Control of the Rijnmond area



Dutch river system

Control of the Rijnmond area

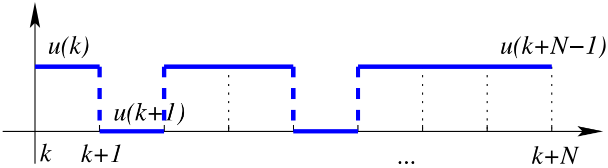


Maintain water levels in cities by controlling gates, subject to tidal sea water level, varying river inflows, safety and actuator constraints

Discrete (actuators) + continuous dynamics (partial differential equations)
→ hybrid MPC approach using mixed-integer nonlinear optimization

Time instant optimization MPC

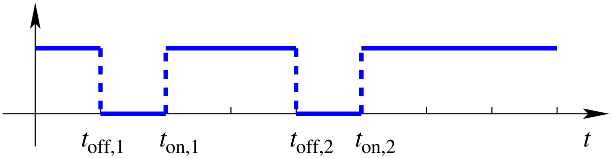
- Consider discrete on-off or open-closed actuator
- Two approaches to model control signal:
  - discrete-valued signal defined at each time step



$\rightarrow$ mixed integer optimization problem (often linear) with N binary variables per actuator

Time instant optimization MPC

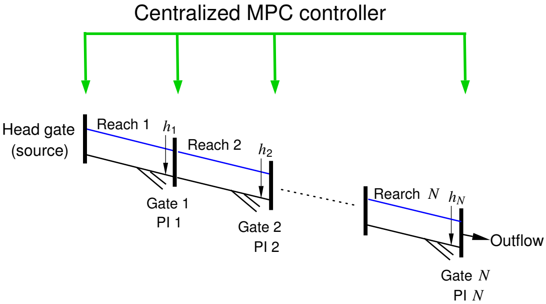
- Consider discrete on-off or open-closed actuator
- Two approaches to model control signal:
  - discrete-valued signal: N binary variables
  - different parametrization: **time instant optimization** assume limited number (M) of on-off switches



$\rightarrow$ real-valued nonlinear optimization problem with $2M$ real-valued variables per actuator

- Especially if horizon N is large, time instant optimization offers significant computational savings

Hierarchical MPC of water distribution canals



- Local PI controllers: 1 for each reach, controls water level by raising or lowering gate
- Set-points of local PI controllers as well as head gate are controlled by MPC controller
- Advantage:
  - robust control solution due to decentralized fast PI controllers
  - coordination via MPC controller (at slower time scale)

Multi-level control of large-scale networks

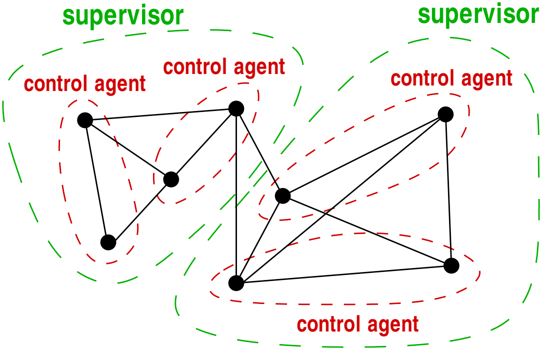
Challenges in control of large-scale networks:

- Large-scale networks
- Distributed vs centralized control
- Optimality $\leftrightarrow$ computational efficiency/tractability
- Global $\leftrightarrow$ local
- Scalability, communication requirements (bandwidth)
- Robustness against failures

$\rightarrow$ multi-level multi-agent approach

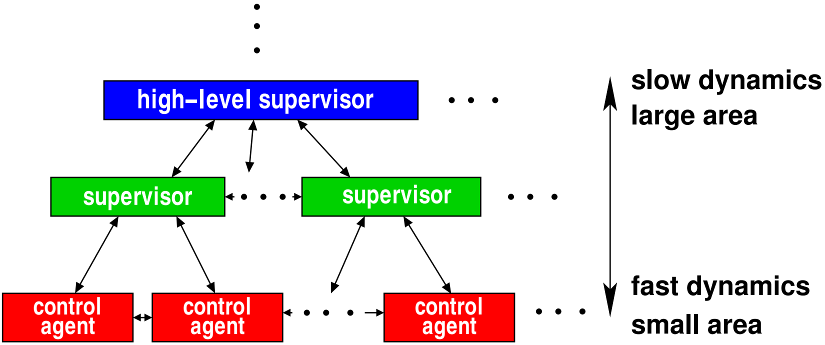
Multi-level multi-agent control

- Multi-level control with intelligent control agents & coordination
- Time-based and space-based separation into layers



Multi-level multi-agent control

- Multi-level control with intelligent control agents & coordination
- Time-based and space-based separation into layers



Multi-level control framework

- Lowest level:
  - local control agents
  - "fast" control
  - small region
  - operational control
- Higher levels:
  - supervisors
  - "slower" control
  - larger regions
  - operational, tactical, strategic control
- Multi-level, multi-objective control structure
- *Coordination at and across all levels*
- Combine with MPC

Main issues and topics in multi-level MPC

- How to obtain tractable prediction models?
- What is the best division into subnetworks?
- Selection of static/dynamic region boundaries?
- How to determine subgoals so as to optimize overall goal?
- How should the higher-level control layers be designed?
- How to effectuate interaction and coordination between agents and control regions?
- How to resolve conflicts & prevent counteracting?
- How can existing approaches be extended to hybrid systems?
- How can the computation/iteration time be reduced? (algorithms, properties, approximations, reductions, . . .)
- Analysis (stability, reliability, robustness, . . .)

Need for traffic control

- Traffic jams & congestion
  - cause time losses, extra costs, more incidents
  - have negative impact on economy, environment, society

- Several ways to reduce traffic jams and to improve traffic performance:
  - new infrastructure, missing links
  - pricing
  - modal shift
  - better use of available capacity through intelligent traffic control
    $\rightarrow$ model predictive traffic control

Traffic management using MPC

- Make use of roadside intelligence
  $\rightarrow$ traffic control center +
  current infrastructure
- Control measures: variable speed limits, ramp metering, traffic signals, lane closures, shoulder lane openings, tidal flow, ...
- Also include "soft" control measures: dynamic route information, travel time information, ...
- Performance criteria: total time spent, fuel consumption, emissions, ...
  $\rightarrow$ consider weighted sum

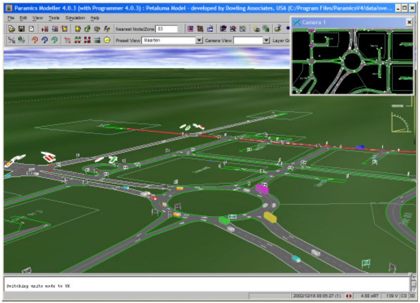
Traffic models

Two main classes of traffic models:

- Microscopic models $\rightarrow$ individual vehicles
- Macroscopic models $\rightarrow$ aggregated variables

Microscopic traffic flow models

- Consider individual vehicles
- Car following + lane changing + overtaking models
- Different driver classes (with different parameters settings)
- Simulation rather time-consuming for large networks
  $\rightarrow$ less suited as prediction model for MPC
  $\rightarrow$ better suited as simulation/validation model



Macroscopic traffic flow models

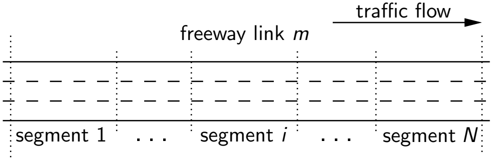
- Work with aggregated variables (average speed, density, flow)
- Examples:
  - fluid-like models: Lighthill-Whitham-Richards (LWR), Payne, *METANET*, ...
  - gas-kinetic models: Helbing model, ...
- Trade-off between computational speed and accuracy

  $\rightarrow$ well suited as prediction model for MPC

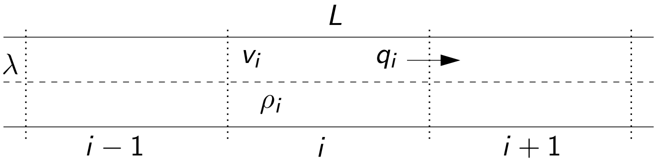
  $\rightarrow$ less suited as simulation/validation model
- In this lecture we use the macroscopic model METANET as prediction model for MPC

METANET

- Developed by Papageorgiou & Messmer
  + various extensions by Hegyi & De Schutter
- Network represented by directed graph
  - highway stretch with uniform characteristics → link divided into N segments of length L
  - on-ramp, off-ramp, change in geometry → node



METANET



- Density (conservation of vehicles):

$$\rho_i(k+1) = \rho_i(k) + \frac{T}{L\lambda}\big(q_{i-1}(k) - q_i(k)\big)$$

- Flow:

$$q_i(k) = \rho_i(k)\, v_i(k)\, \lambda$$

METANET

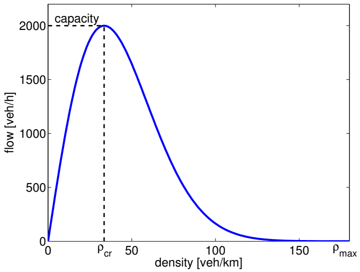
- Speed (relaxation + convection + anticipation):

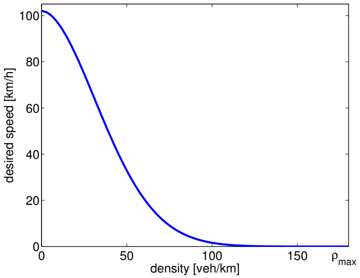
$$
\begin{aligned}
v_i(k+1) = v_i(k) &+ \frac{T}{\tau}\Big(V\big(\rho_i(k)\big) - v_i(k)\Big) \\
&+ \frac{T}{L} v_i(k)\big(v_{i-1}(k) - v_i(k)\big) \\
&- \frac{\nu T}{\tau L} \frac{\rho_{i+1}(k) - \rho_i(k)}{\rho_i(k) + \kappa}
\end{aligned}
$$

METANET

- Desired speed (cf. fundamental diagram):

$$V\left(\rho_i(k)\right) = v_{\mathrm{f}} \, \exp\left[-\frac{1}{a}\left(\frac{\rho_i(k)}{\rho_{\mathrm{cr}}}\right)^a\right]$$





METANET: Extensions

- Effect of speed limit:

$$V(\rho_i(k)) = \min\left(\underbrace{(1+\alpha)\, v_{\text{control},i}(k)}_{\text{speed limit}}, \underbrace{v_{\text{f}}\, \exp\left[-\frac{1}{a}\left(\frac{\rho_i(k)}{\rho_{\text{cr}}}\right)^{a}\right]}_{\text{desired speed}}\right)$$

  α: non-compliance

- Mainstream origin (vs on-ramp)
- Different reaction to higher vs lower downstream density

Shock waves in traffic flows

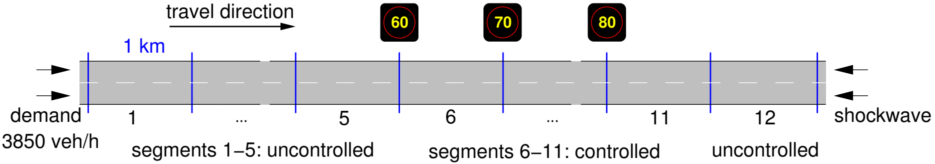
- "Moving" zones of traffic congestion
  arise due to bottlenecks, incidents, sudden braking, ...
  move upstream with approx. 15 km/h
- Cause extra travel time + unsafe situations
- Solution: impose variable speed limits upstream of shock wave

$\rightarrow$ reduce inflow of congested area
such that traffic congestion dissolves/attenuates
$\rightarrow$ create low density wave that propagates downstream
+ compensates (high density) show wave

Variable speed limits

- Goal: suppress/reduce effects of shock waves
- Prevent occurrence of new waves + negative impacts at other locations
- Requires coordination, prediction and optimization:
  - local control versus network control
  - take effects at other locations + future time instants into account
  - (feedback) control

Variable speed limits for reduction of shock waves

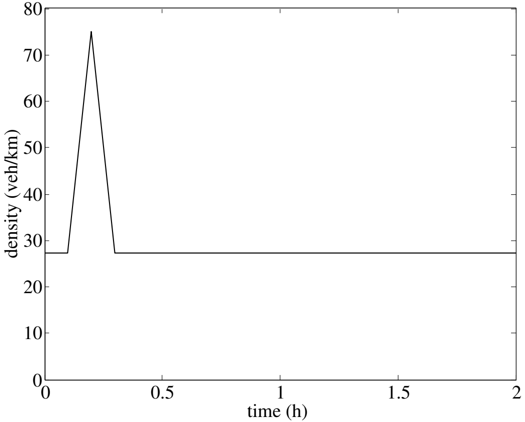


Set-up:

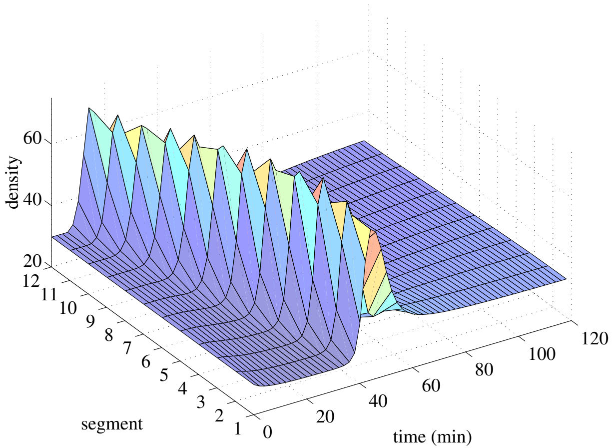
- 12 km freeway stretch, 12 segments of 1 km
- first 5 and last segment uncontrolled
- segment 6 up to 11: variable speed limits
- min. speed limit: 50 km/h
- max. speed limit difference: 10 km/h

Variable speed limits for reduction of shock waves

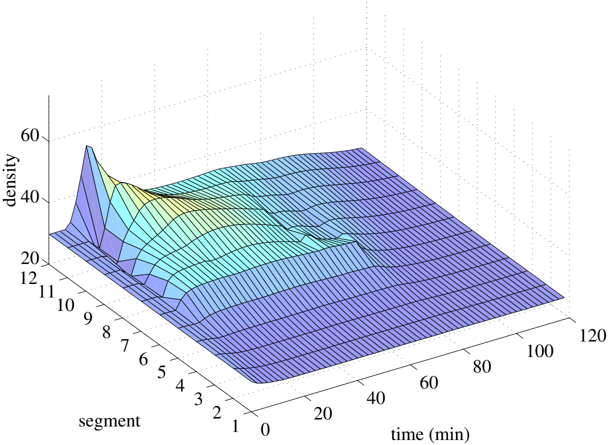
Shock wave enters freeway stretch (downstream density scenario)



No control → shock wave travels through entire stretch



Control (MPC) $\rightarrow$ shock wave disappears



Conventional versus parametrized MPC

Conventional MPC

- Optimizes control inputs

$$\min_{u} J(u)$$

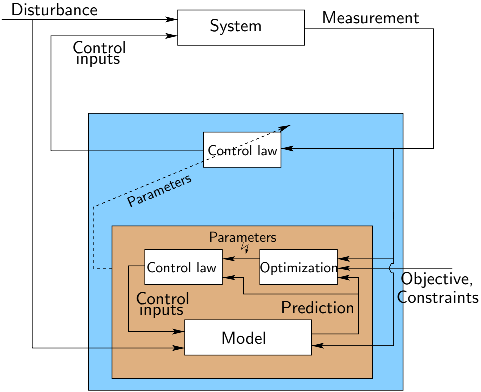
Parametrized MPC

- Optimizes parameter $\boldsymbol{\theta}$

$$\min_{\boldsymbol{\theta}} J\big(u(\boldsymbol{\theta})\big) \text{ with } u = f(\boldsymbol{\theta}, x)$$

- Effect: trade efficiency for optimality
- Note: for previous case study: much faster (up to 75-80 %) than conventional MPC while yielding comparable performance

Parametrized MPC



Define parametrization of control inputs

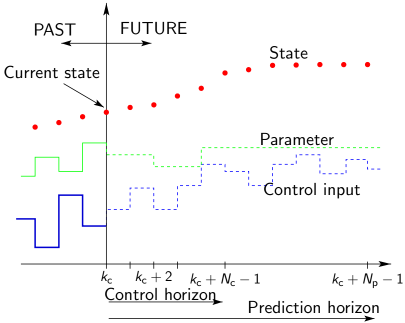
$$u = f(\boldsymbol{\theta}, x)$$

such that $\#(\boldsymbol{\theta}) \leq \#(u)$

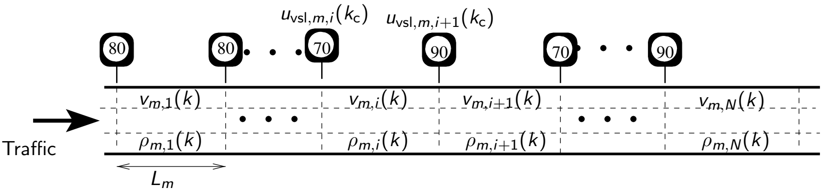
Control time steps can also be different

Parametrized MPC

- Due to state dependency of control law, control signal can still vary over full prediction horizon
- By introducing control horizon N_c or blocking, the number of optimization parameters can be reduced

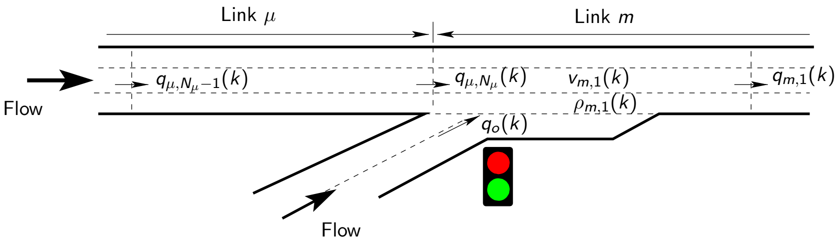


Parametrized variable speed limits



$$u_{\text{vsl},m,i}(k_c+1) = \theta_{0,m} v_{\text{free},m} + \theta_{1,m} \frac{v_{m,i+1}(k_c) - v_{m,i}(k_c)}{v_{m,i+1}(k_c) + \kappa_v} + \theta_{2,m} \frac{\rho_{m,i+1}(k_c) - \rho_{m,i}(k_c)}{\rho_{m,i+1}(k_c) + \kappa_\rho}$$

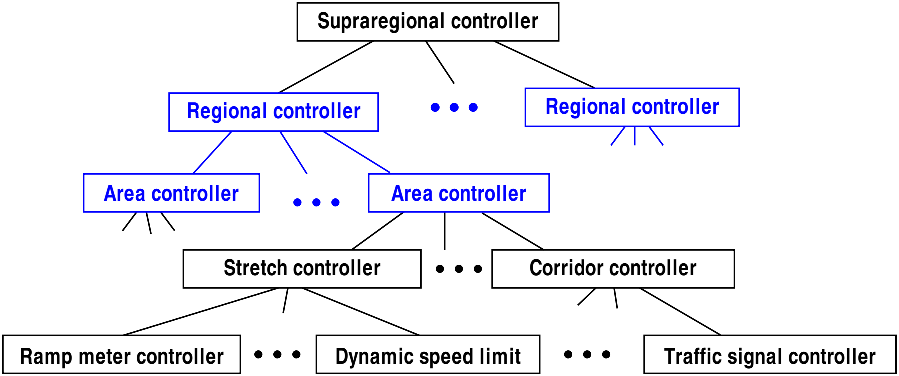
Parametrized on-ramp metering



$$u_{\mathrm{r},m,i}(k_\mathrm{c}+1) = u_{\mathrm{r},m,i}(k_\mathrm{c}) + \theta_{3,m} \frac{\rho_{\mathrm{cr},m} - \rho_{m,i}(k_\mathrm{c})}{\rho_{\mathrm{cr},m}}$$

- cf. ALINEA: $r(k+1) = r(k) + K_R[\hat{o} - o_\mathrm{out}(k)]$

Multi-level traffic control



Multi-level traffic control

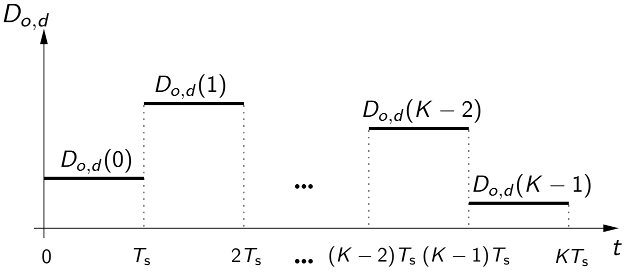
- Traffic signals, ramp metering: basic controllers (PID, logic)
- Freeway stretches, corridors: MPC → coordination + set-points for lower-level controllers
- Area controllers: MPC → routing
- Regional controllers: MPC → high-level routing
- MPC for stretches, corridors, areas, and regions:
  → medium-sized problems due to temporal & spatial division
  → still tractable
- Coordination (top-down) via performance criterion or constraints

Area controllers

- Aim: Route guidance (via tolling, dynamic route information panels, ...)
- Traffic network is represented by graph with nodes and links
- Due to computational complexity, optimal route choice control done via flows on links
- Optimal route guidance: in general, nonlinear integer optimization with high computational requirements → intractable
- Fast approach using Mixed-Integer Linear Programming (MILP)
  - transform nonlinear problem into system of linear equations using binary variables
  - can be solved efficiently using branch-and-bound; several efficient commercial and freeware solvers available

MILP approach – General set-up

- Only consider flows and queue lengths
- Each link has maximal allowed capacity constraint
- Piecewise constant time-varying demand - $[kT_\text{s}, (k+1)T_\text{s})$ for $k = 0, \ldots, K-1$ with K (simulation horizon)



- Main goal: assign optimal flows $x_{l,o,d}(k)$ to each link l

MILP approach – Model

- Inflow at origin:

$$\sum_{l \in L_o^{\text{out}} \cap L_{o,d}} x_{l,o,d}(k) \leqslant D_{o,d}(k) + \frac{q_{o,d}(k)}{T_\text{s}} \quad \text{for each } d \in \mathcal{D}$$

- Outflow from origin to destination:

$$F_{o,d}^{\text{out}}(k) = \sum_{l \in L_o^{\text{out}} \cap L_{o,d}} x_{l,o,d}(k)$$

- Assume constant delay κ between beginning and end of link
- Queue behavior at origin: Total demand − outflow
  i.e., $D_{o,d}(k) - F_{o,d}^{\text{out}}(k)$ in time interval $[kT_\text{s}, (k+1)T_\text{s})$

$$q_{o,d}(k+1) = \max\left(0,\ q_{o,d}(k) + (D_{o,d}(k) - F_{o,d}^{\text{out}}(k))T_\text{s}\right)$$

MILP approach – Equivalences

P1: $[f(x) \leqslant 0] \iff [\delta = 1]$ is true if and only if

$$\begin{cases} f(x) \leqslant M(1-\delta) \\ f(x) \geqslant \varepsilon + (m - \varepsilon)\delta \end{cases}$$

P2: $y = \delta f(x)$ is equivalent to $\begin{cases} y \leqslant M\delta \\ y \geqslant m\delta \\ y \leqslant f(x) - m(1-\delta) \\ y \geqslant f(x) - M(1-\delta) \end{cases}$

- f function with upper and lower bounds M and m
- δ is a binary variable
- y is a real-valued scalar variable
- ε is a small tolerance (machine precision)

$\rightarrow$ transform max equations into MILP equations

MILP approach – Transforming the queue model

$$q_{o,d}(k+1) = \max\left(0,\ q_{o,d}(k) + (D_{o,d}(k) - F^{\text{out}}_{o,d}(k))\,T_{\text{s}}\right)$$

Define

$$[\ \delta_{o,d}(k) = 1\ ] \iff [\ q_{o,d}(k) + (D_{o,d}(k) - F^{\text{out}}_{o,d}(k))\,T_{\text{s}} \geqslant 0\ ]$$

Can be transformed into MILP equations using equivalence P1

$$q_{o,d}(k+1) = \delta_{o,d}(k)\big(\underbrace{q_{o,d}(k) + (D_{o,d}(k) - F^{\text{out}}_{o,d}(k))\,T_{\text{s}}}_{f\ \text{(linear)}}\big)$$

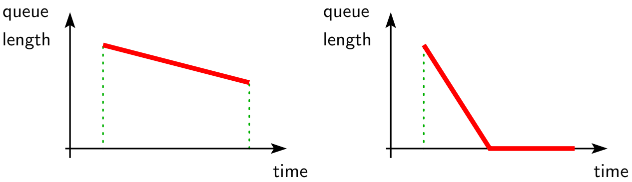
$$= z_{o,d}(k)$$

Product between $\delta_{o,d}(k)$ and f can be transformed into system of MILP equations using equivalence P2

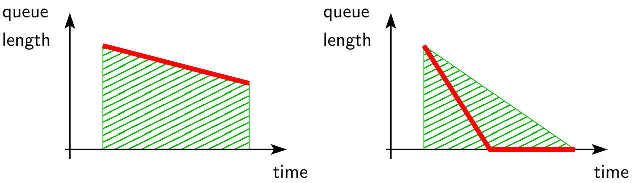
Queue model $\rightarrow$ system of MILP equations

MILP approach – Objective function for queues

Original objective function: time spent in queues (linear/quadratic):



Approximated objective function (linear):



MILP approach – Objective Functions

- Time spent in links:

$$J_{\text{links}} = \sum_{k=0}^{K_{\text{end}}-1} \sum_{(o,d)\in\mathcal{O}\times\mathcal{D}} \sum_{l\in L_{o,d}} x_{l,o,d}(k)\kappa_l T_{\text{s}}^2$$

- Time spent in queues:

$$J_{\text{queue}} = \sum_{k=0}^{K_{\text{end}}-1} \sum_{(o,d)\in\mathcal{O}\times\mathcal{D}} \frac{1}{2}\left(q_{o,d}(k) + q_{o,d}(k+1)\right) T_{\text{s}}$$

MILP approach – Overall area control problems

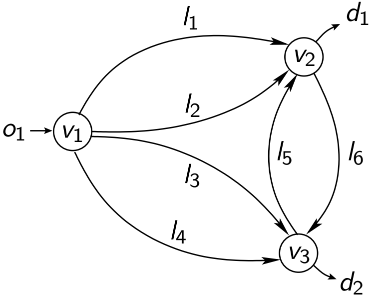
Nonlinear optimization problem:

$$
\begin{aligned}
&\min \left(\text{TTS links} + \text{TTS queues}\right)\\
&\text{subject to}\\
&\qquad \text{nonlinear model}\\
&\qquad \text{operational constraints}
\end{aligned}
$$

MILP optimization problem:

$$
\begin{aligned}
&\min \left(\text{TTS links} + \widehat{\text{TTS}} \text{ queues}\right)\\
&\text{subject to}\\
&\qquad \text{MILP model}\\
&\qquad \text{operational constraints}
\end{aligned}
$$

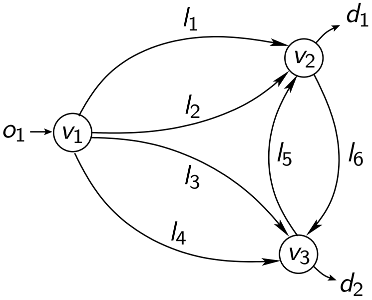
MILP approach – Case study – Set-up



- Dynamic demand case with queues only at origins of network

Period (min)	0–10	10–30	30–40	40–60
D_{o_1,d_1} (veh/h)	5000	8000	2500	0
D_{o_1,d_2} (veh/h)	1000	2000	1000	0

MILP approach – Case study – Set-up



- Scenario:
  - simulation period: 60 min, sampling time: 1 min
  - capacities: C_1=1900 veh/h, C_2=2000 veh/h, C_3=1800 veh/h, C_4=1600 veh/h, C_5=1000 veh/h, and C_6=1000 veh/h
  - delay factor: κ_1=10, κ_2=9, κ_3=6, κ_4=7, κ_5=2, and κ_6=2

MILP approach – Case study – Results

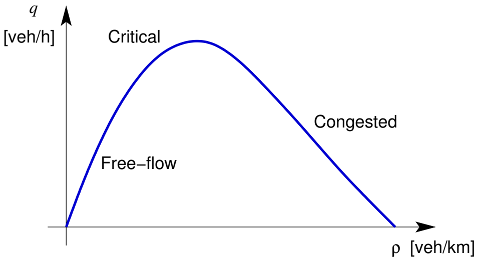
Case	TTS_{tot} (veh.h)	improvement	CPU time (s)
No control	1434	0 %	–
MILP	1081	24.6 %	0.27
SQP (5 initial points)	1067	25.6 %	90.0
SQP (50 initial points)	1064	25.8 %	983
SQP (with MILP solution as initial point)	1064	25.8 %	1.29

Regional controllers

- Control collection of areas
- Aim: Determine optimal flows of vehicles between areas
- Model: Aggregate model – Macroscopic Fundamental Diagram (MFD)
- Optimization: Nonlinear nonconvex programming problem
  $\rightarrow$ will be approximated using MILP

Macroscopic Fundamental Diagram (MFD)

- Introduced by Geroliminis and Daganzo
- Describes relation between space-mean flow and density in neighborhood-sized sections of cities (up to 10 km^2)
- Macroscopic fundamental diagram is independent of the demand
- Outflow of area is proportional to space-mean flow within area



Flow control between areas

- Represent traffic network by graph
  - links correspond to areas, with inflow $q_{\mathrm{in},a}(k)$, outflow $q_{\mathrm{out},a}(k)$, and density $\rho_a(k)$
  - nodes correspond to connections between areas, external origins (with inflow $q_{\mathrm{orig},o}(k)$), or external exits (with outflow $q_{\mathrm{exit},e}(k)$)

Model for regional controllers

- Network MFD results in static description of form

$$q_{\text{out},a}(k) = \mathcal{M}_a(\rho_a(k))$$

- Evolution of densities inside each area is described using simple conservation equation:

$$\rho_a(k+1) = \rho_a(k) + \frac{T}{L_a}(q_{\text{in},a}(k) - q_{\text{out},a}(k))$$

with T sample time step system and L_a measure for total length of highways and roads in area a

- For every node ν balance between inflows and outflows:

$$\sum_{a \in \mathcal{I}_\nu} q_{\text{out},a}(k) + \sum_{o \in \mathcal{I}_{\text{orig},\nu}} q_{\text{orig},o}(k) = \sum_{a \in \mathcal{O}_\nu} q_{\text{in},a}(k) + \sum_{e \in \mathcal{O}_{\text{exit},\nu}} q_{\text{exit},e}(k)$$

MPC for regional controllers

- Try to keep density in each region below critical density $\rho_{\text{crit},a}$:

$$J_{\text{pen}}(k) = \sum_{j=1}^{N_\text{p}} \sum_a \left[\max(0, \rho_a(k+j) - \rho_{\text{crit},a}) \right]^2$$

- Also minimize total time spent (TTS) by all vehicles in region:

$$J_{\text{TTS}}(k) = \sum_{j=1}^{N_\text{p}} \sum_a L_a \rho_a(k+j) T$$

- Total objective function:

$$J(k) = J_{\text{pen}}(k) + \gamma J_{\text{TTS}}(k)$$

- Constraints on maximal flows from one area to another,...
- Results in nonlinear, nonconvex optimization problem

Mixed integer linear programming (MILP) – Equivalences

P1: $[f(x) \leqslant 0] \iff [\delta = 1]$ is true if and only if

$$\begin{cases} f(x) \leqslant M(1-\delta) \\ f(x) \geqslant \varepsilon + (m - \varepsilon)\delta \end{cases}$$

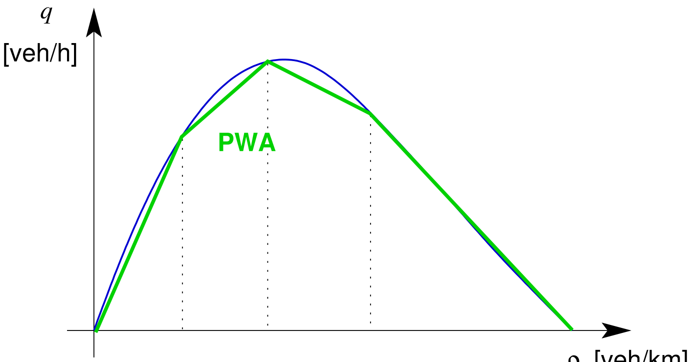
P2: $y = \delta f(x)$ is equivalent to $\begin{cases} y \leqslant M\delta \\ y \geqslant m\delta \\ y \leqslant f(x) - m(1-\delta) \\ y \geqslant f(x) - M(1-\delta) \end{cases}$

- f function with upper and lower bounds M and m
- δ is a binary variable
- y is a real-valued scalar variable
- ε is a small tolerance (machine precision)

Transformation into MILP problem

- Approximate MFD by piecewise affine function

$$q_{\mathrm{out},a}(k) = \alpha_{a,i}\rho_a(k) + \beta_{a,i} \quad \text{if } \rho_a(k) \in [\rho_{a,i}, \rho_{a,i+1}]$$



Transformation into MILP problem

- Approximate MFD by piecewise affine function

$$q_{\text{out},a}(k) = \alpha_{a,i}\rho_a(k) + \beta_{a,i} \quad \text{if } \rho_a(k) \in [\rho_{a,i}, \rho_{a,i+1}]$$

- Introduce binary variables $\delta_{a,i}(k)$ such that

$$\delta_{a,i}(k) = 1 \;\text{ if and only if }\; \rho_a(k) \leq \rho_{a,i+1}$$

Can be transformed into MILP equations using equivalence P1

- Now we have

$$q_{\text{out},a}(k) = \sum_{i=1}^{N_a} \big((\alpha_{a,i} - \alpha_{a,i-1})\rho_a(k) + (\beta_{a,i} - \beta_{a,i-1})\big)\delta_{a,i}(k)$$

- Introduce real-valued auxiliary variables $y_{a,i}(k) = \rho_a(k)\delta_{a,i}(k)$
Can be transformed into MILP equations using equivalence P2

Transformation into MILP problem

- Results in

$$q_{\mathrm{out},a}(k) = \sum_{i=1}^{N_a} (\alpha_{a,i} - \alpha_{a,i-1}) y_{a,i}(k) + (\beta_{a,i} - \beta_{a,i-1}) \delta_{a,i}(k)$$

- If we combine all equations and inequalities, we obtain a system of mixed-integer linear inequalities

Transformation into MILP problem

- Recall

$$J_{\text{pen}}(k) = \sum_j \sum_a \left[\max(0, \rho_a(k+j) - \rho_{\text{crit},a}) \right]^2 \quad \rightarrow \text{not linear}$$

$$J_{\text{TTS}}(k) = \sum_j \sum_a L_a \rho_a(k+j) T \quad \rightarrow \text{linear!}$$

- Removing square in $J_{\text{pen}}(k)$ results in piecewise affine objective function
  Can be transformed in MILP equations using P1 & P2
- Hence, we get MILP problem
- Solution of MILP problem can be directly applied or it can be used as good initial starting point for original nonlinear, nonconvex MPC optimization problem

Related work: Intelligent Vehicle Highway Systems (IVHS)

- Integrate various in-vehicle and roadside-based traffic control measures that support platoons of fully autonomous vehicles

platoon

cooperative adaptive cruise control

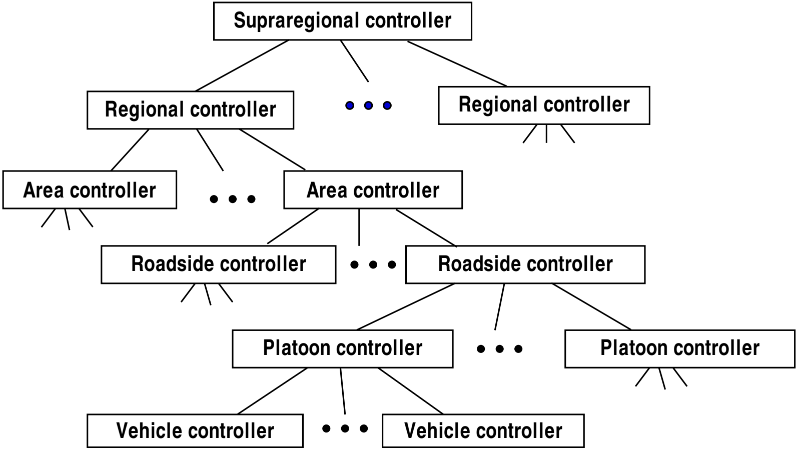
intelligent speed adaptation

dynamic route guidance

- Goal: improved traffic performance (safety, throughput, environment, ...) + constraints (robustness, reliability, ...)

A multi-scale HD-MPC approach for IVHS

→ multi-level multi-layer control approach (∼ California PATH)



Cooperative Vehicle Infrastructure Systems

- Intermediate step between current system and IVHS

Summary

- Model predictive control for large-scale systems $\rightarrow$ main issue: computational complexity
- Dealing with computational issues:
  - trade-off between accuracy and efficiency
  - use of macroscopic models
  - parametrized controllers
  - approximations
  - distributed control
  - multi-level control
- Applications: water distribution networks and road networks
- For more information: also see website of EU project HD-MPC (Hierarchical and Distributed MPC): `http://www.ict-hd-mpc.eu`