Hierarchical MPC with applications in transportation and infrastructure networks

Milan, Italy, August 28, 2011

HD–MPC

SEVENTH FRAMEWORK PROGRAMME
1. HD-MPC for large-scale systems
2. Traffic management and automated highway systems
3. Multi-level multi-scale HD-MPC for AHS
4. Related work
5. Conclusions and future work
Challenges in control of large-scale networks:

- Large-scale networks
- Distributed vs centralized control
- Optimality ↔ computational efficiency/tractability
- Global ↔ local
- Scalability
- Communication requirements (bandwidth)
- Robustness against failures

→ 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

Hierarchical MPC for transportation networks — Bart De Schutter

HD-MPC for large-scale systems

Supervisor

Control agent

High-level supervisor

Supervisor

Control agent

Slow dynamics
Large region

Fast dynamics
Small region
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 model predictive control (MPC)
Major problem for MPC in practice: Required computation time for large-scale systems

- Use distributed and/or hierarchical control approach
- Choice of the prediction model: accuracy versus computational complexity
- 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
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**
Intelligent traffic control

Next generation traffic control and management system

- Use in-car telematics (navigation, telecommunication, information, ...) systems
- Vehicle-vehicle + vehicle-roadside communication
- Use intelligent vehicles (IVs)
  - control system senses environment using sensors
  - enhances either performance of driver or vehicle itself
    - assisting (advisory/warning)
    - taking partial or complete control (full automation)
- Two variants of traffic management using IVs:
  - cooperative vehicle-infrastructure systems (CVIS): drivers are still in charge of their vehicles
  - Automated Highway Systems (AHS): autonomous vehicles organized in platoons
Automated highway systems (AHS)

- Platoons of intelligent, autonomous vehicles
- Small inter-vehicle distance inside distances + high speeds → higher throughput
- Larger inter-platoon distance for safety
- Problems:
  - transition
  - psychological & legal aspects → long-term, trucks
Automated highway systems (AHS)

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

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

Hierarchical MPC for transportation networks — Bart De Schutter
HD-MPC
Additional advantage of platoons: No capacity drop

- Capacity drop for human drivers: If traffic flow breaks down, then afterwards outflow from congested area is less than previous higher flow

- Reason: Human drivers tend to accelerate more slowly when they are coming out of congestion
- This effect plays less or even not with autonomous vehicles
Traffic flow models

Two main classes:

- 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
  → less suited as prediction model for MPC
  → 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 versus accuracy
  → well suited as prediction model for MPC
  → less suited as simulation/validation model

- In this presentation we use macroscopic models for automated highway systems as prediction model for MPC
A multi-level multi-scale HD-MPC approach for AHS → hierarchical multi-layer control approach (≈ California PATH)
## Multi-level multi-scale HD-MPC for AHS

<table>
<thead>
<tr>
<th>Controller</th>
<th>Unit</th>
<th>Control</th>
<th>Time scale</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vehicle</td>
<td>vehicle</td>
<td>throttle, brake, steering</td>
<td>$\ll s$</td>
</tr>
<tr>
<td>Platoon</td>
<td>vehicles</td>
<td>distances &amp; speeds, trajectories</td>
<td>$&lt; s$</td>
</tr>
<tr>
<td>Roadside</td>
<td>platoons</td>
<td>lanes &amp; speeds, split &amp; merge</td>
<td>s–min</td>
</tr>
<tr>
<td>Area</td>
<td>flows of platoons</td>
<td>routing</td>
<td>$&gt; \text{min}$</td>
</tr>
<tr>
<td>Regional</td>
<td>flows</td>
<td>routing</td>
<td>$&gt; 15–30 \text{ min}$</td>
</tr>
</tbody>
</table>
Control strategies

- Vehicle controllers: (adaptive) PID + logic (for safety)
- Platoon controllers: rule-based control, hybrid control
- Roadside, area, regional controllers: MPC

\[
\min_{u(k), \ldots, u(k+N_c-1)} J(k)
\]

s.t. model of system

operational constraints

→ medium-sized problems due to temporal & spatial division
→ still tractable

- Coordination (top-down) via performance criterion $J$ or constraints
Roadside controllers

- Control highway or stretch of highway
- Measurements: position, speed, lanes of platoon leaders
- Control inputs: platoon speeds, lane allocations, on-ramp release times
- Objectives:
  - track speed and splitting rate profiles imposed by area controllers
  - minimize total time spent (TTS) in network and queues, ...
- Constraints: min. headway, min. and max. speeds
MPC for roadside controllers

- Model: “big-car” model
  platoon = vehicle with speed-dependent length

\[ L_{\text{platoon},p}(k) = (n_p - 1)S_0 + \sum_{i=1}^{n_p-1} T_{\text{gap},i}v_{n_p}(k) + \sum_{i=1}^{n_p} L_i \]

with \( S_0 \) minimum safe distance at zero speed and \( T_{\text{gap},i} \) the desired time gap

- Nonlinear optimization problem:
  \[ \min (\text{TTS links} + \text{TTS queues}) \]
  subject to nonlinear model
  operational constraints

- Optimization: mixed-integer nonlinear programming
  Simplify by bi-level approach in which first lane allocation is determined (via heuristics, optimized, slower rate, . . . )
Case study – Problem statement

Two-lane highway with an incident causing traffic

Scenario:
- **Demand**: 2500 veh/h (mainstream) and 350 veh/h (on-ramp)
- **Incident** at 4-5 km, start of simulation (10 minutes)
- **Queues** at start: empty
- **Simulation period**: 10 min, **controller sampling time**: 1 min
- **Simulation sampling time**: 1 s
Case study – Cases

Cases considered:

- Uncontrolled human drivers
- Controlled human drivers (current situation)
- Platoon approach – our approach
## Case study – Results

<table>
<thead>
<tr>
<th>Case</th>
<th>TTS (veh·h)</th>
<th>Relative improvement (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Uncontrolled</td>
<td>71.80</td>
<td>0 %</td>
</tr>
<tr>
<td>Controlled (human drivers)</td>
<td>63.38</td>
<td>10.96 %</td>
</tr>
<tr>
<td>Controlled (platoons)</td>
<td>57.75</td>
<td>18.86 %</td>
</tr>
</tbody>
</table>

Reduced TTS → decreased travel times, increased trips, . . .
Area controllers

- Route guidance + provide set-points for roadside controllers
- 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: nonlinear integer optimization with high computational requirements $\rightarrow$ intractable
Area controllers (contd.)

- Fast approaches based on
  - 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
  - macroscopic METANET-like traffic flow model
    - for humans, splitting rates are determined by traffic assignment
    - in AHS, splitting rates considered as controllable input
    - will result in non-convex *real-valued* optimization
MILP approach – General set-up

- Only consider flows and queue lengths
- Each link has maximal allowed capacity constraint
- Piecewise constant time-varying demand - \([kT_s, (k + 1)T_s]\) for \(k = 0, \ldots, K - 1\) with \(K\) (simulation horizon)

Main goal: assign optimal flows \(x_{l,o,d}(k)\)
MILP approach – Model

- Inflow at origin:

\[
\sum_{l \in L_{o}^{\text{out}} \cap L_{o,d}} x_{l,o,d}(k) \leq D_{o,d}(k) + \frac{q_{o,d}(k)}{T_s} \quad \text{for each } d \in \mathcal{D}
\]

- 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 \( \kappa \) between beginning and end of link

- Queue behavior at origin: Total demand – outflow

- More specifically, \( D_{o,d}(k) - F_{o,d}^{\text{out}}(k) \) in time interval \([kT_s, (k + 1)T_s)\)

\[
q_{o,d}(k + 1) = \max\left(0, q_{o,d}(k) + (D_{o,d}(k) - F_{o,d}^{\text{out}}(k))T_s\right)
\]
MILP approach – Equivalences

\textbf{P1:} \quad [f(x) \leq 0] \iff [\delta = 1] \text{ is true if and only if}

\[
\begin{align*}
& f(x) \leq M(1 - \delta) \\
& f(x) \geq \epsilon + (m - \epsilon)\delta
\end{align*}
\]

\textbf{P2:} \quad y = \delta f(x) \text{ is equivalent to}

\[
\begin{align*}
& y \leq M\delta \\
& y \geq m\delta \\
& y \leq f(x) - m(1 - \delta) \\
& y \geq f(x) - M(1 - \delta)
\end{align*}
\]

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

\( \rightarrow \) transform max equations into MILP equations
MILP approach – Transforming the queue model

\[ q_{o,d}(k+1) = \max (0, q_{o,d}(k) + (D_{o,d}(k) - F_{o,d}^{\text{out}}(k)) T_s) \]

Define

\[ [ \delta_{o,d}(k) = 1 ] \iff [ q_{o,d}(k) + (D_{o,d}(k) - F_{o,d}^{\text{out}}(k)) T_s \geq 0 ] \]

Can be transformed into MILP equations using equivalence P1

\[ q_{o,d}(k+1) = \delta_{o,d}(k) \left( q_{o,d}(k) + (D_{o,d}(k) - F_{o,d}^{\text{out}}(k)) T_s \right) \]

\[ = z_{o,d}(k) \]

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:

\[
J_{\text{links}} = \sum_{k=0}^{K_{\text{end}}-1} \sum_{(o,d) \in \mathcal{O} \times \mathcal{D}} \sum_{l \in L_{o,d}} \chi_{l,o,d}(k) \kappa_l T_s^2
\]

- Time spent in queues:

\[
J_{\text{queue}} = \sum_{k=0}^{K_{\text{end}}-1} \sum_{(o,d) \in \mathcal{O} \times \mathcal{D}} \frac{1}{2} (q_{o,d}(k) + q_{o,d}(k+1)) T_s
\]
**MILP approach – Overall area control problem**

**Nonlinear optimization problem:**

\[
\min \left( \text{TTS links} + \text{TTS queues} \right)
\]
subject to
- nonlinear model
- operational constraints

**MILP optimization problem:**

\[
\min \left( \text{TTS links} + \hat{\text{TTS queues}} \right)
\]
subject to
- MILP model
- operational constraints
MILP approach – Case study

Figure: Set-up of case study network
MILP approach – Case study – Set-up

- Dynamic demand case with queues only at origins of network

<table>
<thead>
<tr>
<th>Period (min)</th>
<th>0–10</th>
<th>10–30</th>
<th>30–40</th>
<th>40–60</th>
</tr>
</thead>
<tbody>
<tr>
<td>$D_{o_1,d_1}$ (veh/h)</td>
<td>5000</td>
<td>8000</td>
<td>2500</td>
<td>0</td>
</tr>
<tr>
<td>$D_{o_1,d_2}$ (veh/h)</td>
<td>1000</td>
<td>2000</td>
<td>1000</td>
<td>0</td>
</tr>
</tbody>
</table>

- 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: $\kappa_1=10$, $\kappa_2=9$, $\kappa_3=6$, $\kappa_4=7$, $\kappa_5=2$, and $\kappa_6=2$
MILP approach – Case study – Cases

Cases considered

- Case A: no control
- Case B: controlled using the MILP solution
- Case C: controlled using the exact solution
## MILP approach – Case study – Results

<table>
<thead>
<tr>
<th>Case</th>
<th>$TTS_{\text{tot}}$ (veh.h)</th>
<th>Improvement</th>
<th>CPU time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>No control</td>
<td>1434</td>
<td>0 %</td>
<td>-</td>
</tr>
<tr>
<td>MILP</td>
<td>1081</td>
<td>24.6 %</td>
<td>0.27</td>
</tr>
<tr>
<td>SQP (5 initial points)</td>
<td>1067</td>
<td>25.6 %</td>
<td>90.0</td>
</tr>
<tr>
<td>SQP (50 initial points)</td>
<td>1064</td>
<td>25.8 %</td>
<td>983</td>
</tr>
<tr>
<td>SQP (with MILP solution as initial point)</td>
<td>1064</td>
<td>25.8 %</td>
<td>1.29</td>
</tr>
</tbody>
</table>
MILP approach – Case study – Analysis

- **Uncontrolled** case: only direct/short routes are used. Length of origin queue increases with time
- **Controlled** cases: flows assigned to both short and long routes
Regional controllers

- Control collection of areas
- Determine optimal flows of platoons between areas
- Model: aggregate model – AHS variant of the Macroscopic Fundamental Diagram (MFD)
- Optimization: Nonlinear non-convex programming problem
  Will be approximated using mixed-integer linear programming
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 km²)
- Macroscopic fundamental diagram is independent of the demand
- Outflow of area is proportional to space-mean flow within area
Macroscopic Fundamental Diagram for AHS

- Adopt modified version of MFD for AHS
- Shape of MFD will be sharper and maximal flow will be much higher than in MFD for human drivers
- Represent AHS network by graph
  - links correspond to areas, with inflow $q_{\text{in},a}(k)$, outflow $q_{\text{out},a}(k)$, and density $\rho_a(k)$
  - nodes correspond to connections between areas, external origins (with inflow $q_{\text{orig},o}(k)$), or external exits (with outflow $q_{\text{exit},e}(k)$)
Model for regional controllers

- Network MFD for AHS results in static description of form

\[ q_{out,a}(k) = M_a(\rho_a(k)) \]

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

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

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

- For every node \( \nu \) balance between inflows and outflows:

\[
\sum_{a \in I_{\nu}} q_{out,a}(k) + \sum_{o \in I_{orig,\nu}} q_{orig,o}(k) = \\
\sum_{a \in O_{\nu}} q_{in,a}(k) + \sum_{e \in O_{exit,\nu}} q_{exit,e}(k)
\]
MPC for regional controllers

- Try to keep density in each region below critical density $\rho_{\text{crit},a}$:
  $$J_{\text{pen}}(k) = \sum_{j=1}^{N_p} \sum_{a} \left[ \max(0, \rho_a(k + j) - \rho_{\text{crit},a}) \right]^2$$

- Also minimize total time spent (TTS) by all vehicles in region:
  $$J_{\text{TTS}}(k) = \sum_{j=1}^{N_p} \sum_{a} L_a \rho_a(k + j) T$$

- 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, non-convex optimization problem
Mixed integer linear programming (MILP) – Two properties

- Given function $f$ with lower bound $m$ and upper bound $M$

- **Property 1:**
  \[ f(x) \leq 0 \iff [\delta = 1] \]
  is equivalent to

  \[
  \begin{cases}
  f(x) \leq M(1 - \delta) \\
  f(x) \geq \varepsilon + (m - \varepsilon)\delta
  \end{cases}
  \]

- **Property 2:**
  \[ y = \delta f(x) \text{ with } \delta \in \{0, 1\} \]
  is equivalent to

  \[
  \begin{cases}
  y \leq M\delta \\
  y \geq m\delta \\
  y \leq f(x) - m(1 - \delta) \\
  y \geq f(x) - M(1 - \delta)
  \end{cases}
  \]
Transformation into MILP problem

- Approximate MFD by Piece-Wise Affine (PWA) function

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

- Approximate MFD by Piece-Wise Affine (PWA) function

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

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

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

Can be transformed into MILP equations using Property 1

- Now we have

\[ q_{out,a}(k) = \sum_{i=1}^{N_a} (\alpha_{a,i} \rho_a(k) + \beta_{a,i}) \delta_{a,i}(k) \]

- Introduce real-valued auxiliary variables \( y_{a,i}(k) = \rho_a(k) \delta_{a,i}(k) \)

Can be transformed into MILP equations using Property 2
Transformation into MILP problem

- Results in

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

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

- Recall

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

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

- Removing square in \( J_{\text{pen}}(k) \) results in PWA objective function
  Can be transformed in MILP equations using Properties 1 & 2
- 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, non-convex MPC optimization problem
Related work: Traffic management using MPC

- More viable option on short term: roadside intelligence
  → traffic control center +
  current infrastructure

- Use conventional 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, . . .
Ongoing research

- Address complexity issues for large-scale systems
  - simplified models for urban traffic networks
  - parametrized MPC
- Alternative objective functions + related models
  - emissions: CO, NO_x, CO_2, HC, ...
  - fuel consumption
Cooperative Vehicle Infrastructure Systems

- Intermediate step between current system and AHS
Related work: Traffic management using MPC

Other applications

- Electricity networks
- Water networks
- Railway networks
- Logistic systems
Conclusions

- Hierarchical control framework for automated highway systems (AHS)
- Focus on roadside, area, and regional controllers
- In general: nonlinear, non-convex mixed-integer optimization problems
- Reduce complexity of problem by selecting appropriate models and making approximations
- Results by bi-level, mixed-integer linear programming, or nonlinear, non-convex real-valued optimization problems

Future work

- extensive integrated case study & assessment
- further development of HD-MPC approaches
- further improvements in efficiency and performance
Main issues and topics in HD-MPC for transportation and infrastructure networks

- 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 can existing approaches be extended to hybrid systems?
- How can the computation/iteration time be reduced further? (algorithms, properties, approximations, reductions, . . . )