stabilizing and optimizing feedback control of urban traffic

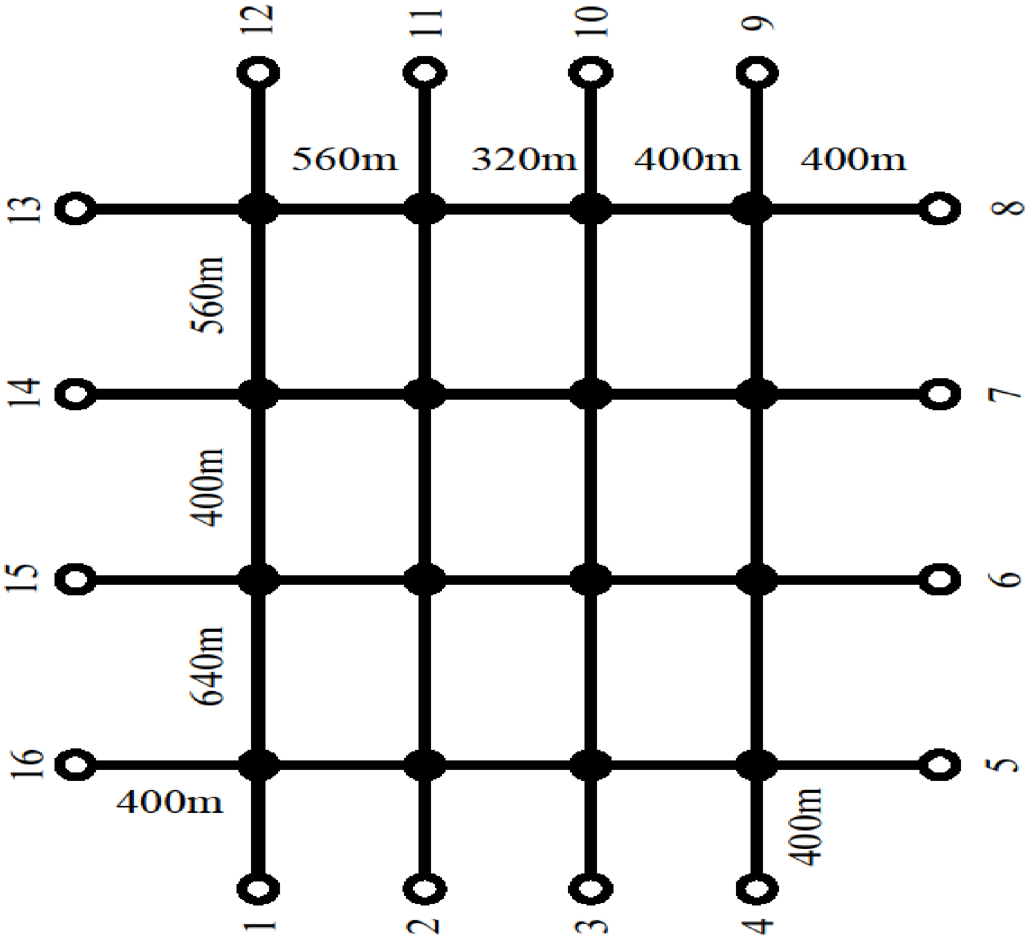
René Boel (Ghent University, Belgium),

Zhenzhen Hao (Xidian Univ., Xi'an, China)

outline

- goals and methodology
- urban CTM (UCTM) model for control design
- reducing delay thanks to local MPC
- stabilizing queues by coordinated MPC
- conclusions and future work

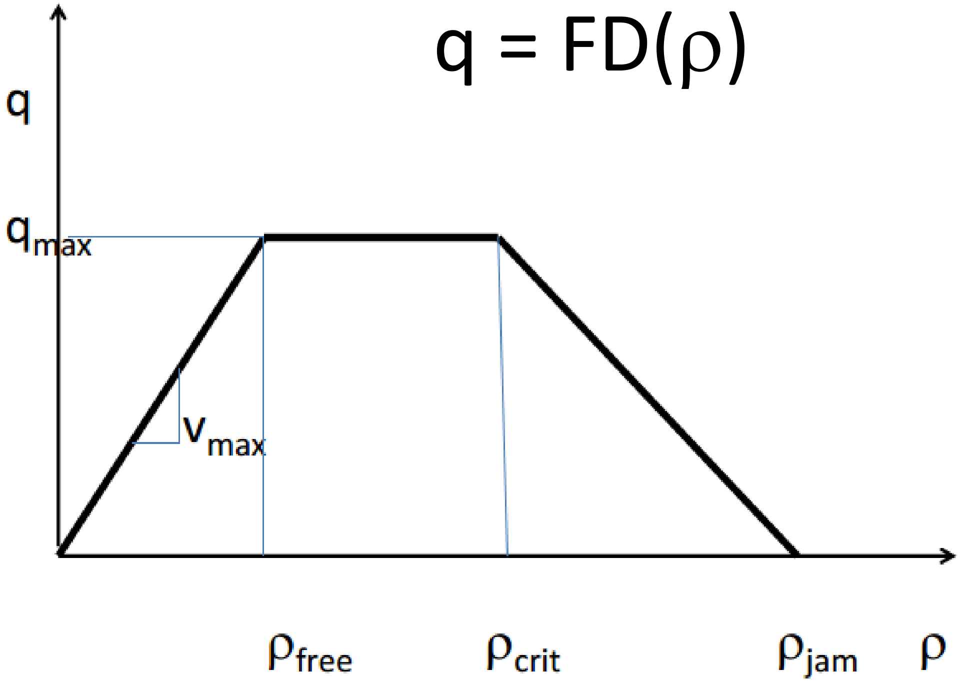
Case study: region to be controlled



macroscopic fundamental diagram (MFD)

long term average flow rate $q_n = FD_n(\rho_n)$ out of region$_n$ depends mainly on average traffic density ρ_n inside region$_n$ (assuming spatial and temporal homogeneity, depending on local traffic control)

$q = FD(\rho)$



ρ veh/m, q veh/sec, v m/sec,

$q = \rho.v$

Goals of regional feedback controller

- Find good compromise between
  - offline green wave timing of traffic lights
  - local feedback adaptation to random arrivals
- Distributed control using local information only
  - robust against communication failures, model uncertainty (e.g. changes in road conditions)
  - easy to modify when layout of network changes

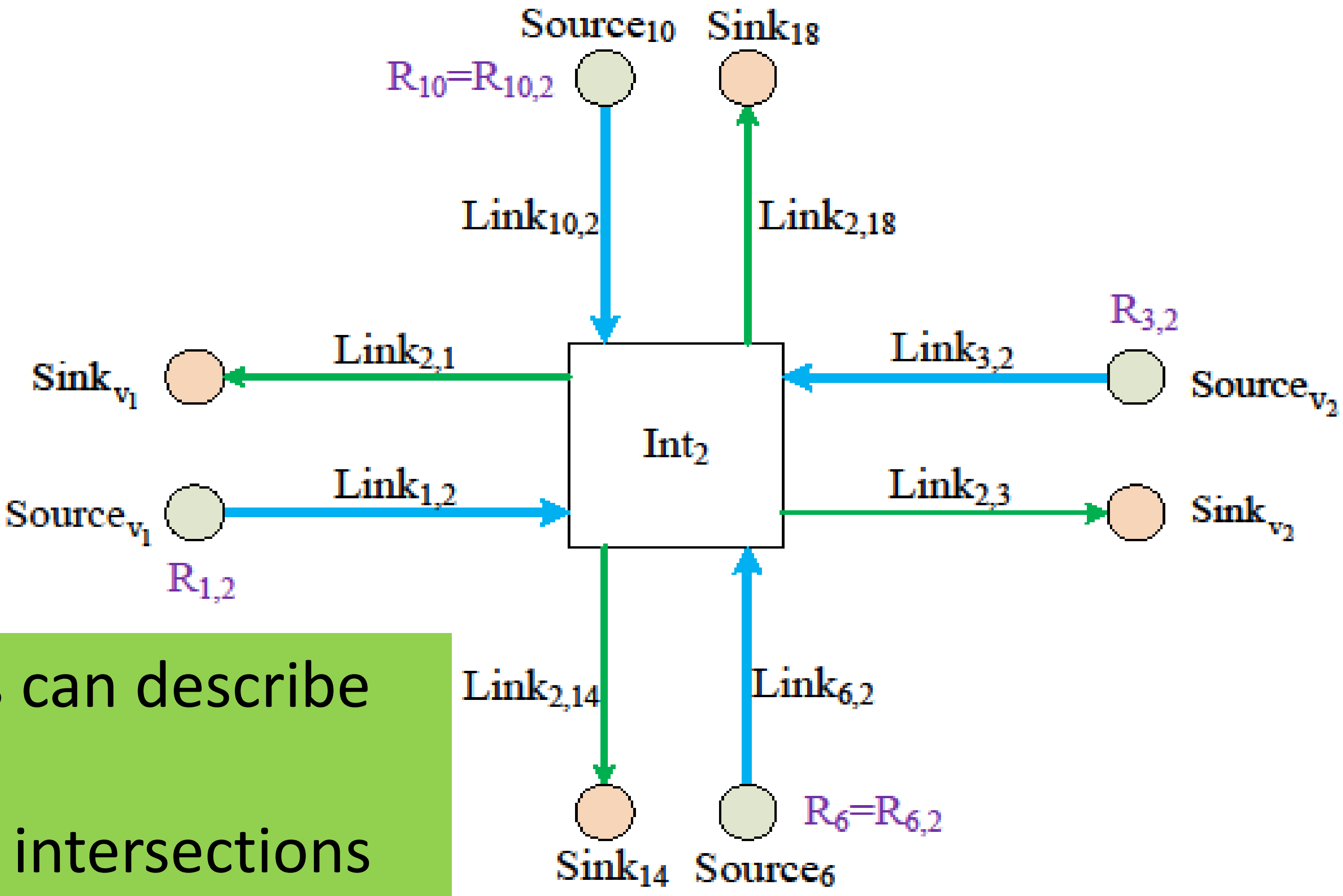
Goals of regional feedback controller

- Design methodology for selecting switching times of traffic lights must be

  easy to tune for very different networks

  - easy to understand
  - depending on few model parameters
  - for each intersection use as much as possible local online and offline information only

Distributed control of a single intersection



Sources/sinks can describe traffic flow at neighbouring intersections or external sources

feedback control of traffic at one intersection

- select whether to extend current phase of traffic light or to switch to next allowable phase
- yellow periods uncontrollable: fixed duration (3sec)
- min and max duration for each green phase (20-60sec)
- each service phase (= mode of timed automaton) enables only nonconflicting flows of vehicles

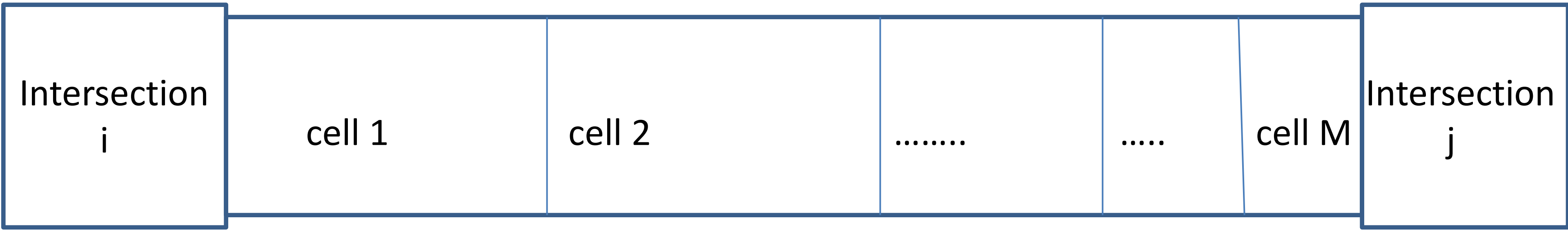
Time scale: adjust every few seconds (5sec)

outline

- goals and methodology of current talk
- urban CTM (UCTM) traffic model used for local control design
- proposed local control MPC
- stabilizing feedback control
- conclusions and future work

Model of link$_{ij}$
(each link representing one way traffic)

- Need model that approximately represents
  state of link = location of all vehicles



Intersection i

cell 1

cell 2

……..

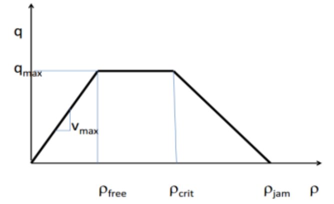
…..

cell M

Intersection j

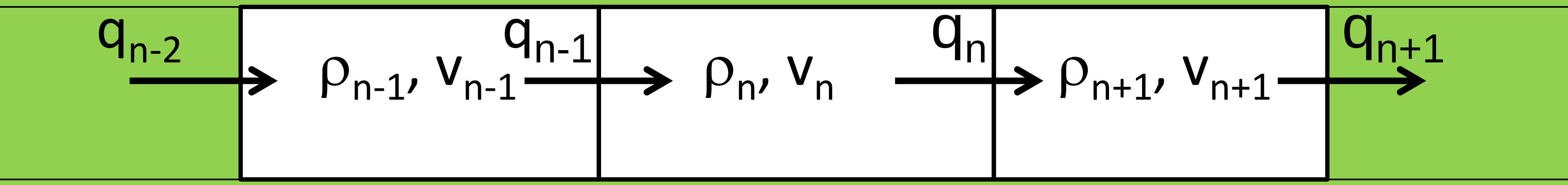
- state variable per cell = traffic density (equivalently: number of vehicles per cell)
- (cells 20m-80m)

Speed calculated according to fundamental diagram



Cell transmission (CTM) model

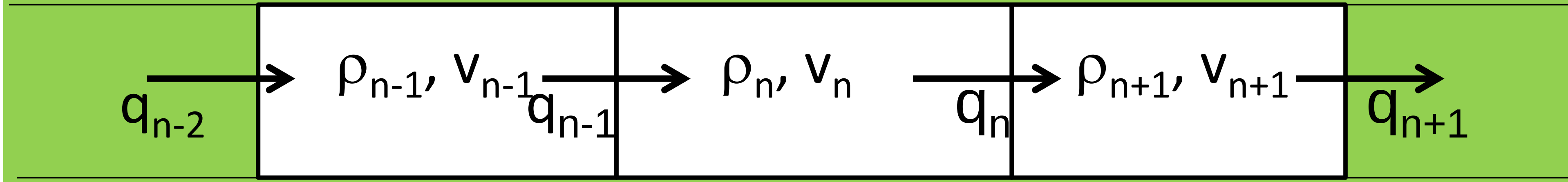
- introduced by Daganzo as efficient first order discretization of LWR pde's for flows q_n of vehicles, guaranteeing physically sensible solutions
- partition one-way road in cells {cell_n, n = 1,...,N} with time updated every Δt sec ($t_k = k.\ \Delta t$; cell length $\geq v_{max}\ .\Delta t$)



- and space homogeneous states in each cell_n
  (vehicle density $\rho_n(t_k)$, speed $v\ _n(t_k)$)

UCTM model for urban traffic

- dynamic evolution completely described by updating $(\rho_{n,k}, q_{n,k}) \rightarrow (\rho_{n,k+1}, q_{n,k+1})$ using relationship conservation of vehicles, $q_{n,k} = \rho_{n,k}.v_{n,k}$ and $q_{n,k} = FD_n(\rho_{n,k})$,



for urban traffic update also must express constraints for:

- merge/diverge, parking garages, unsignalized intersections
- queue discharge delay expressing inertia of queue of stopped vehicles getting green light

State update equations for UCTM

- $q_n(t) = \min\{send(\rho_n(t)), q_{n,max}(t), receive(\rho_{n+1}(\mathrm{t}))\}$

and conservation equations

- $N_n(t+1) = N_n(t) + q_{n-1}(t) - q_n(t)$

with obvious modifications at merge and diverge locations (including blocking due to downstream jams)

outline

- goals and methodology
- UCTM traffic models
- reducing delay using local MPC
- reducing delay and stabilizing queues by coordinated MPC
- conclusions and future work

Local model predictive control

- Each intersection uses local MPC based on
  - online estimate of current state in each upstream and downstream link
  - offline information on average flow rates at upstream cell of links (if necessary via adaptive estimation)
- for each scenario predict state trajectory over horizon and delay,
- using UCTM as a fast simulator, starting in estimated initial state

Local model predictive control

- Use UCTM simulation to calculate predicted average delay for all traffic in upstream links, over horizon of at least 1 cycle (currently: 160sec ≈ 2 average cycles),

  for different scenarios of switching times,

  starting in current state

Average delay = $\Sigma_{k \in [t,t+H)} \Sigma_{\text{all cells } n} (q_{\text{free flow}} - q_n(t_k))/\text{outflow}$

Cost to be optimized

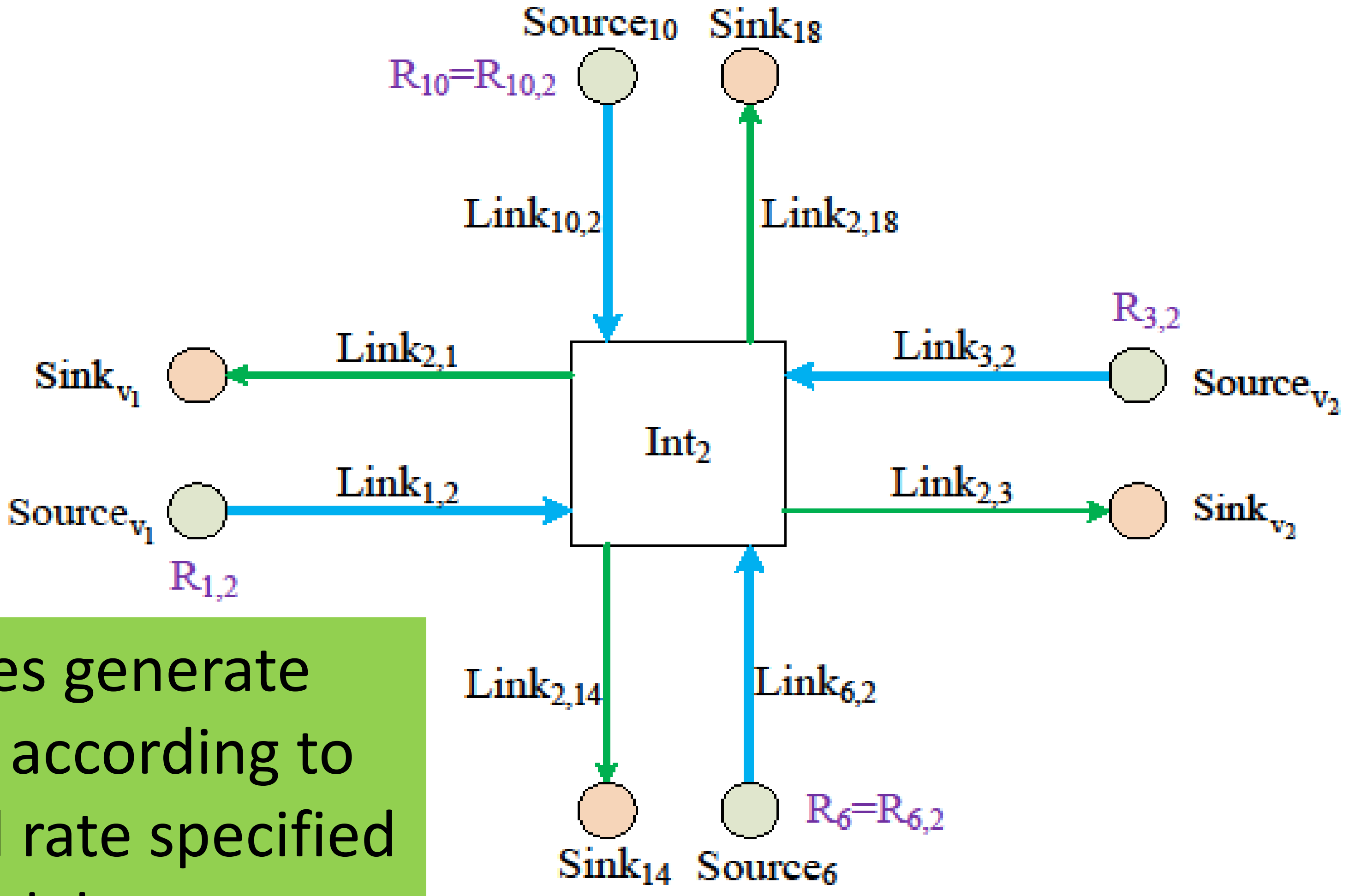
- Average delay + final cost

$$= \Sigma_{k \in \text{horizon}} \Sigma_{\text{all cells } n} (q_{\text{free flow}} – q_n(t_k))/\text{outflow}$$

$$+ \chi.\Sigma_{\text{all links}} \text{ queues at final time } t_k + H$$

where all cells in all incoming (upstream) links are considered

Intersection model used for local MPC



Sources generate traffic according to arrival rate specified by model

Current implementation: Local MPC

- scenario based optimization every 20sec
- Switch phase of traffic light if lowest cost scenario requires switch after 5, 10, 15, 20sec;

  otherwise recalculate after 20sec
- Reduce number of scenarios using
  - min and max duration of phases
  - 2nd cycle in horizon is identical repetition of 1st cycle
  - duration of 2nd phase in 1st cycle determined as explicit function of past values of $Q_{upstream} – Q_{downstream}$
  - simple state feedback law for left turning phases

local MPC

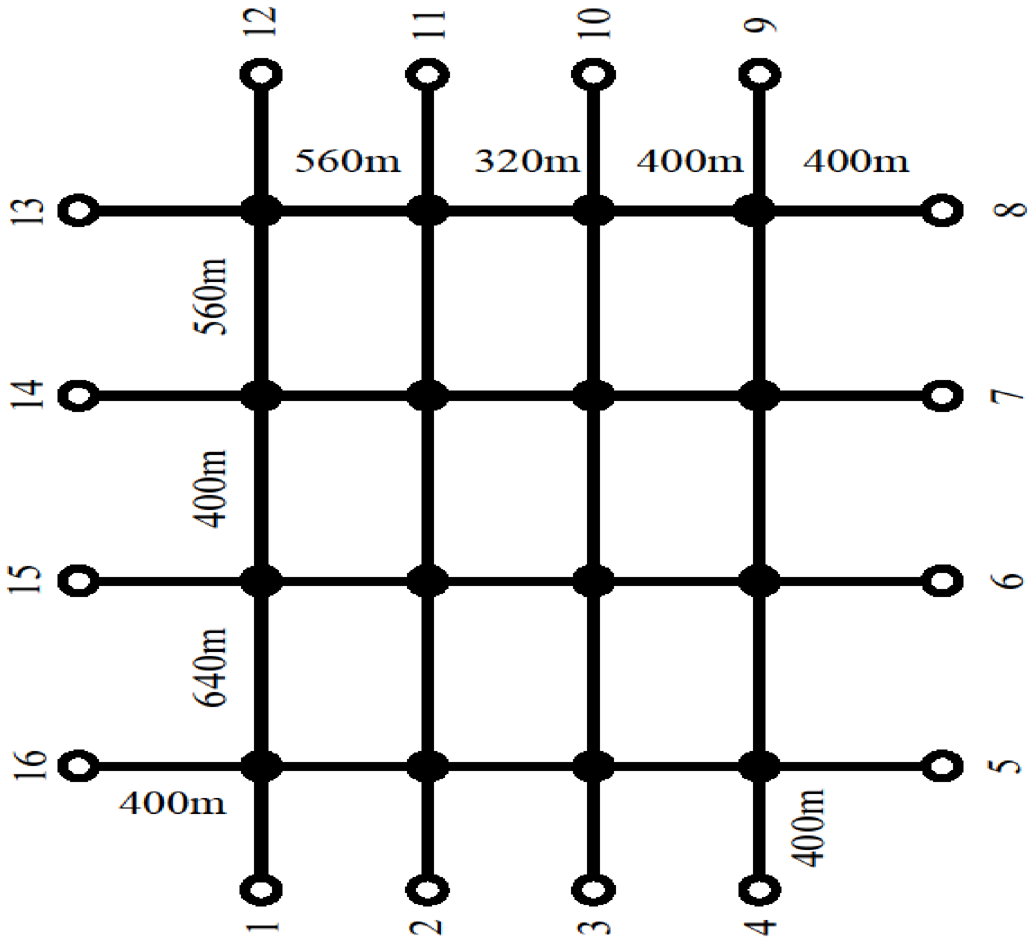
- Local MPC control strategy is **scalable**
  - "local": optimization only for local delays, selecting local switching times, at intersection
  - "distributed": uses only local online information on number of vehicles in all cells in links connected to intersection under control
  - but needs information on average arrival rate of vehicles at upstream links

Performance evaluation

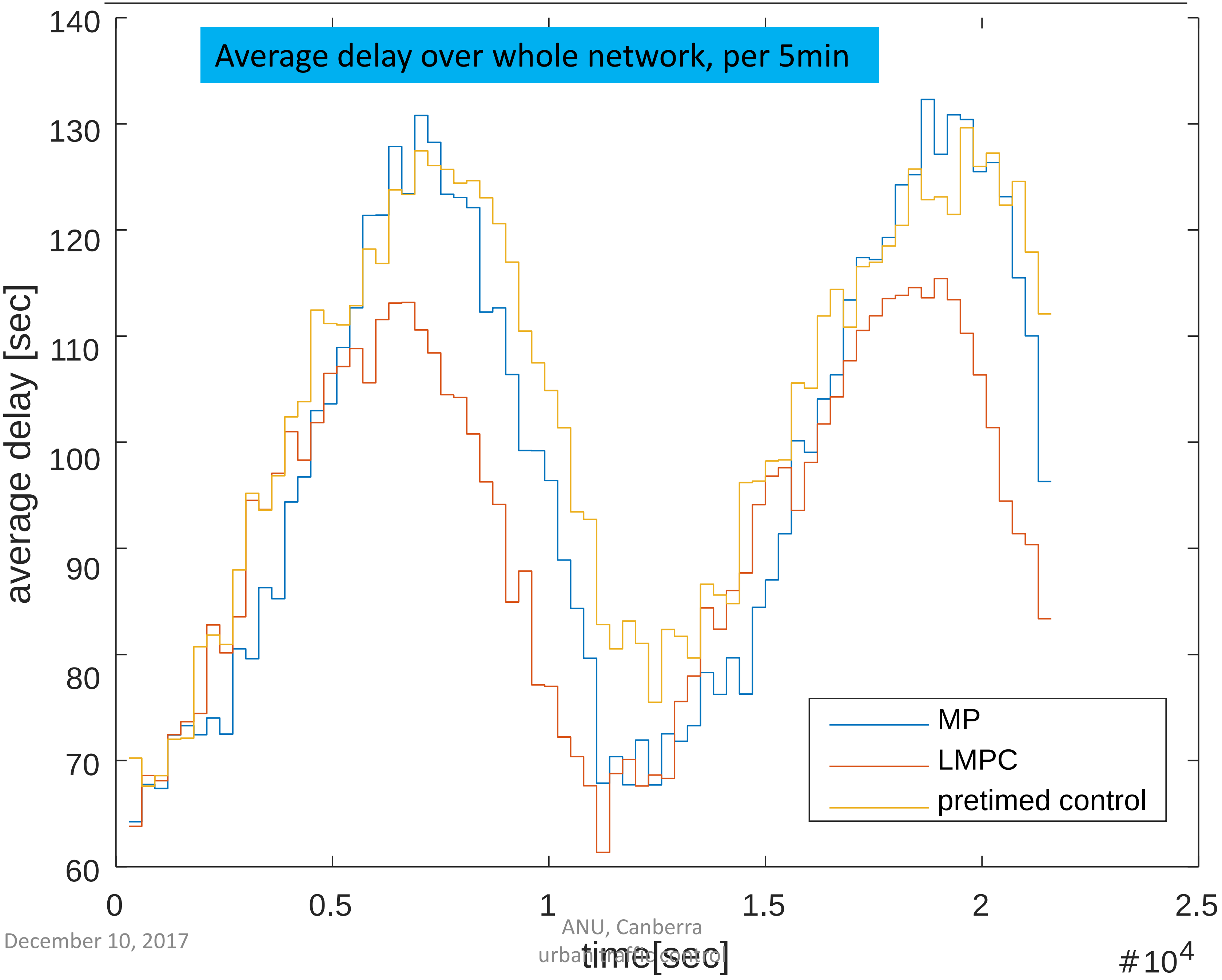
- Comparison via simulation of performance of
  - Pretimed control Webster's rule (adjusted every 15min to adapt to changed load - very optimistic, not practically feasible!)
  - Max pressure control (requires no model information at all!)
  - Local MPC (of course in our simulated case study the measurement noise is missing)

For 4 by 4 Manhattan type urban traffic network

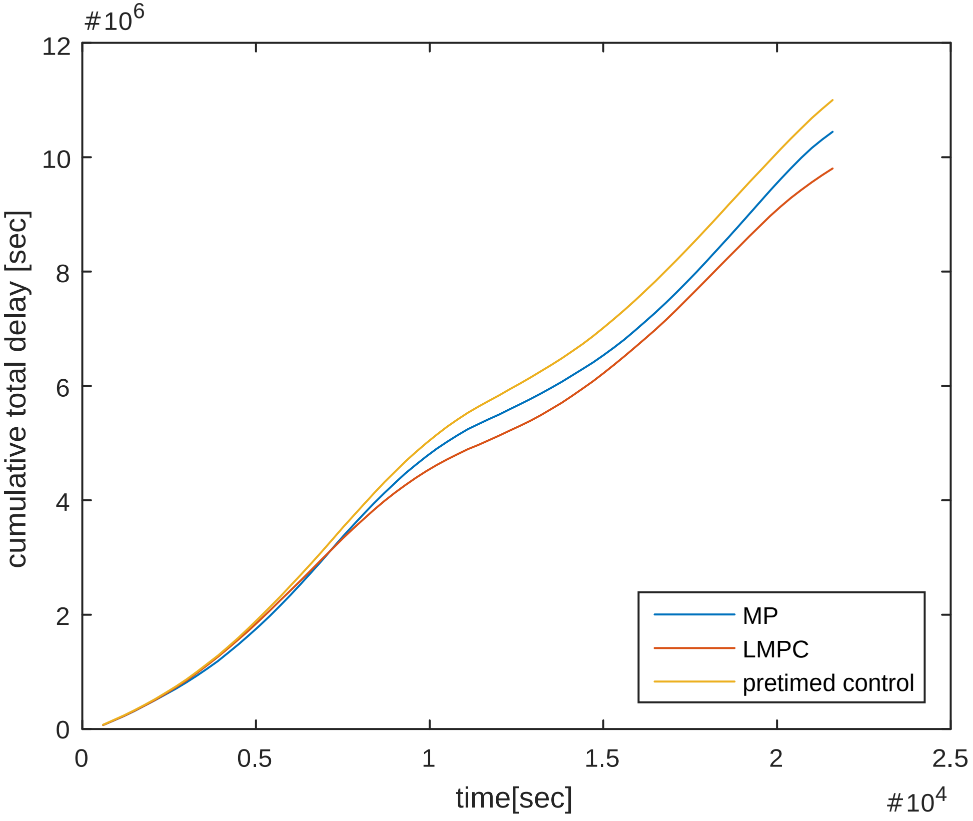
Case study for performance analysis



Time varying load (2 periods of sinusoidal load variations, with different phase shifts at different sources, successively making different intersections heavily loaded)


Average delay over whole network, per 5min
140
130
120
110
100
90
80
70
60
average delay [sec]
0
0.5
1
1.5
2
2.5
time[sec]
# 10^4
MP
LMPC
pretimed control

cumulative delay for local MPC



includes delay of vehicles stored in sources at edge of network when links at edge are too full to receive more vehicles

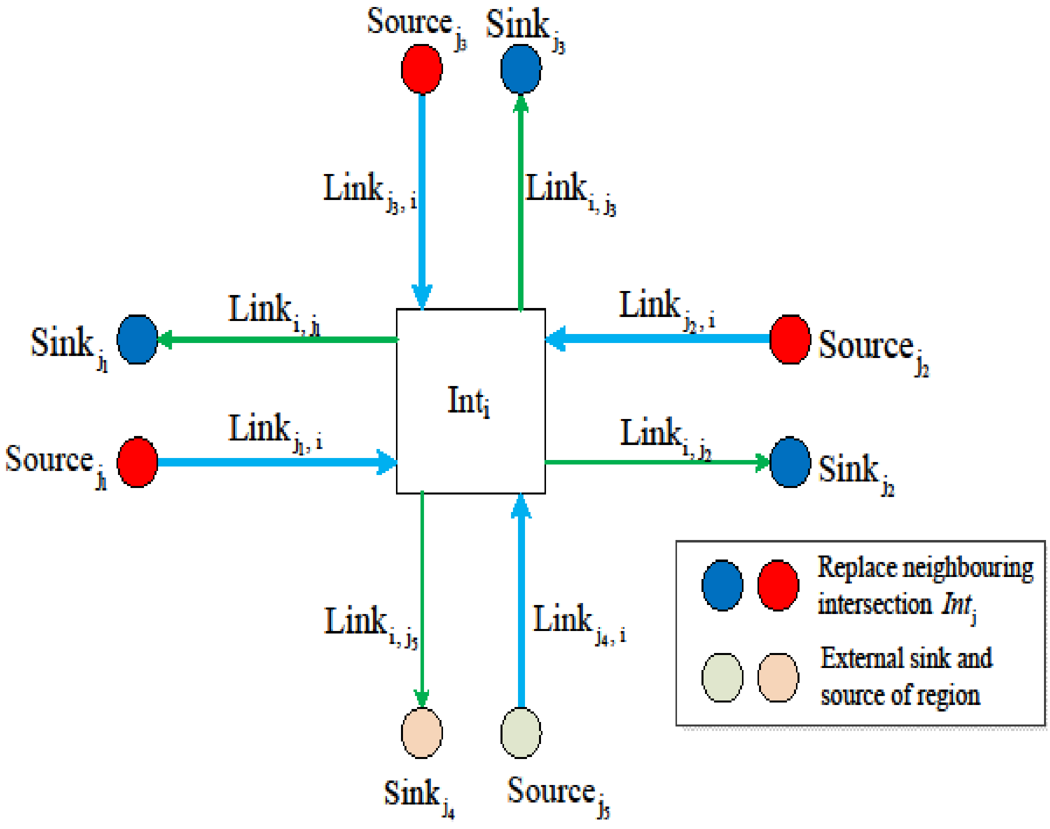
- goals and methodology
- traffic models used for control design:
  - extended CTM for local control
  - macroscopic fundamental diagram for higher level controller
- reducing delay thanks to local MPC
- Reducing delay and stabilizing queues by coordinated MPC
- conclusions and future work

Limitations of local MPC

- Traffic flow at neighbouring intersections are replaced by sources generating average flow, and uncontrolled sinks
- Performance can be improved by using information on planned switching times (optimal MPC scenario) of neighbouring intersections

Coordinated control of intersections

Delay predictions
for different scenarios
replace source$_i$
i = 1, 2, 6, 10
by planned output
at intersections Int$_i$
and keep track of planned
green times at Int$_j$,
j = 1,14, 2, 18



Coordinated MPC

- Needs extra communication capability but can generate “closed loop synchronisation” of traffic lights
- Coordination requires extra cost for changes to planned scenario
- Cost = average delay + final cost + change cost

$$= \Sigma_{k \in \text{horizon}} \Sigma_{\text{all cells n}} (q_{\text{free flow}} - q_n(t_k))/\text{outflow}$$

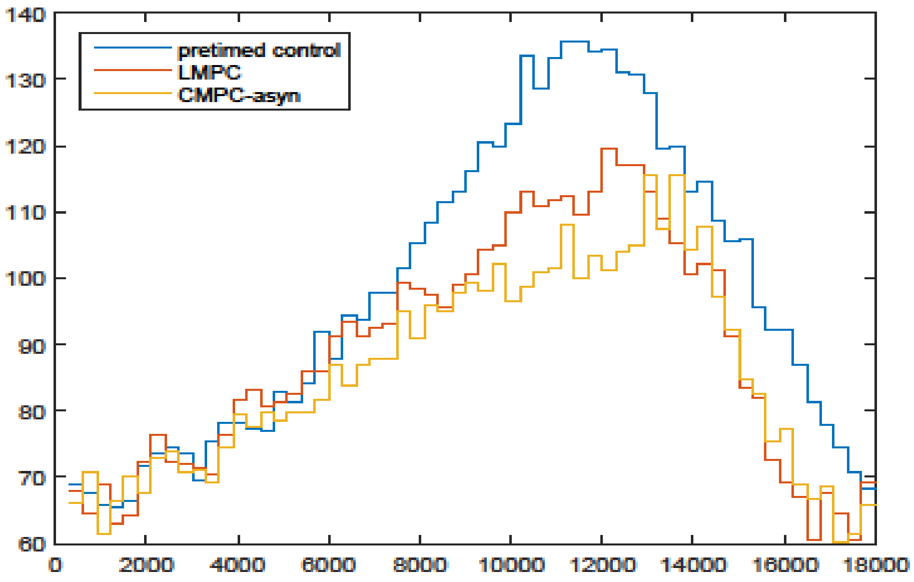
$$+ \chi.\Sigma_{\text{all links}} \text{ queues at final time } t_k + H$$

$$+ \kappa.k(Sce^*(t) - Sce^*(t\text{-}1))$$

Coordinated MPC

- Same as local MPC but replace sources by predictions based on current optimal scenario for traffic light switching at each neighbouring intersection
- More accurate prediction of arrival times of platoons from upstream intersection
- Better prediction of evolution of queues in downstream link

Average delay for coordinated MPC

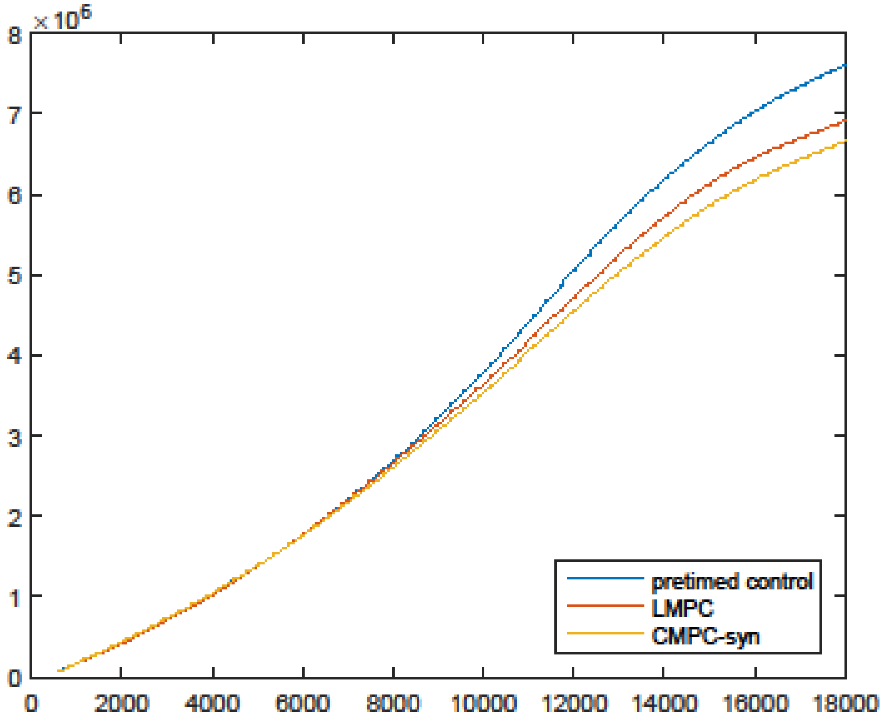


Average delay comparison for different load

Asynch refers to updating optimal decisions at
at different times
at different intersections,
improving performance compared to
updating all at same time

Neglect communication delays

Cumulative delay comparison



Coordination reduces delay by a few percent only in this experiment, but only 4 out of 16 intersections coordinate with all 4 neighbours, other intersections connected to 1 or 2 external sources anyway

Tentative control Lyapunov function for stability constraint

- $V(X_t) = \Sigma Q_n^2$ = sum of squares of queue lengths (according to current state)
- Σ queue lengths $\approx$ constant #vehs inside region
  $\Rightarrow$ minimum of V achieved
  when all queues equal
- desirable property for stability is to keep congestion uniformly distributed over region

Improve stability of coordinated MPC

- Replace queue by total number N_ℓ of vehicles in link$_\ell$ when using control Lyapunov function for stabilizing CMPC:

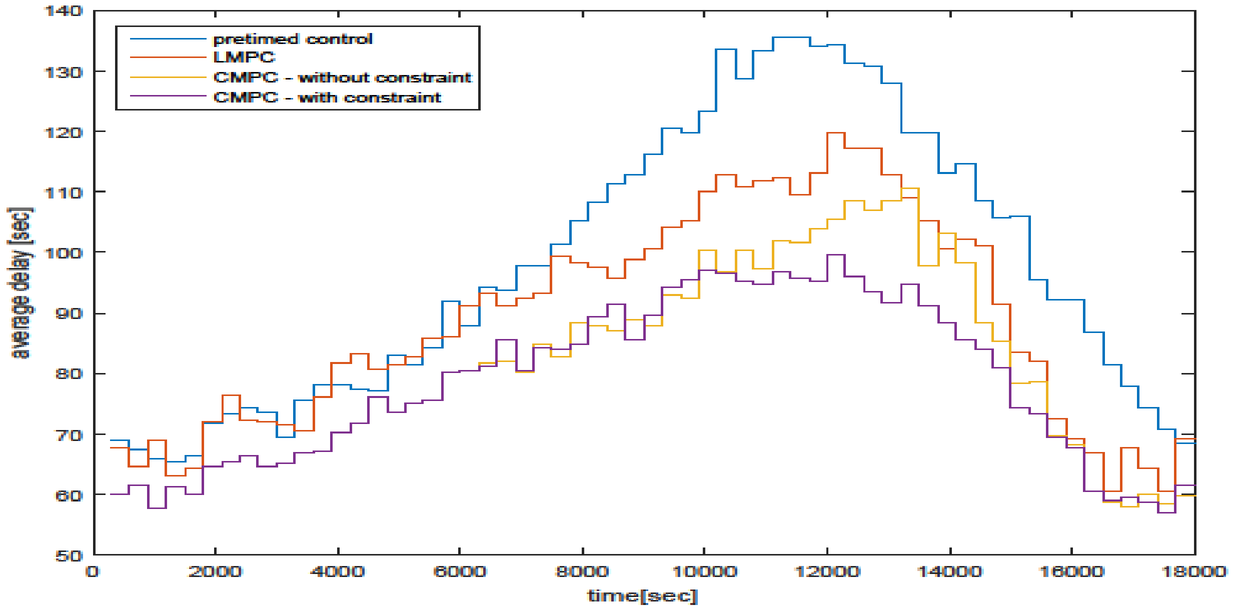
$$V(N) = \sum(N_\ell{}^2 / N_{\ell,max}{}^2)$$

- Allow for coordinated MPC only those scenarios that ensure V(N) decreases whenever $N_\ell / N_{\ell,max}$> a (threshold to be selected),
  by ensuring each local component decreases local Lyapunov function component

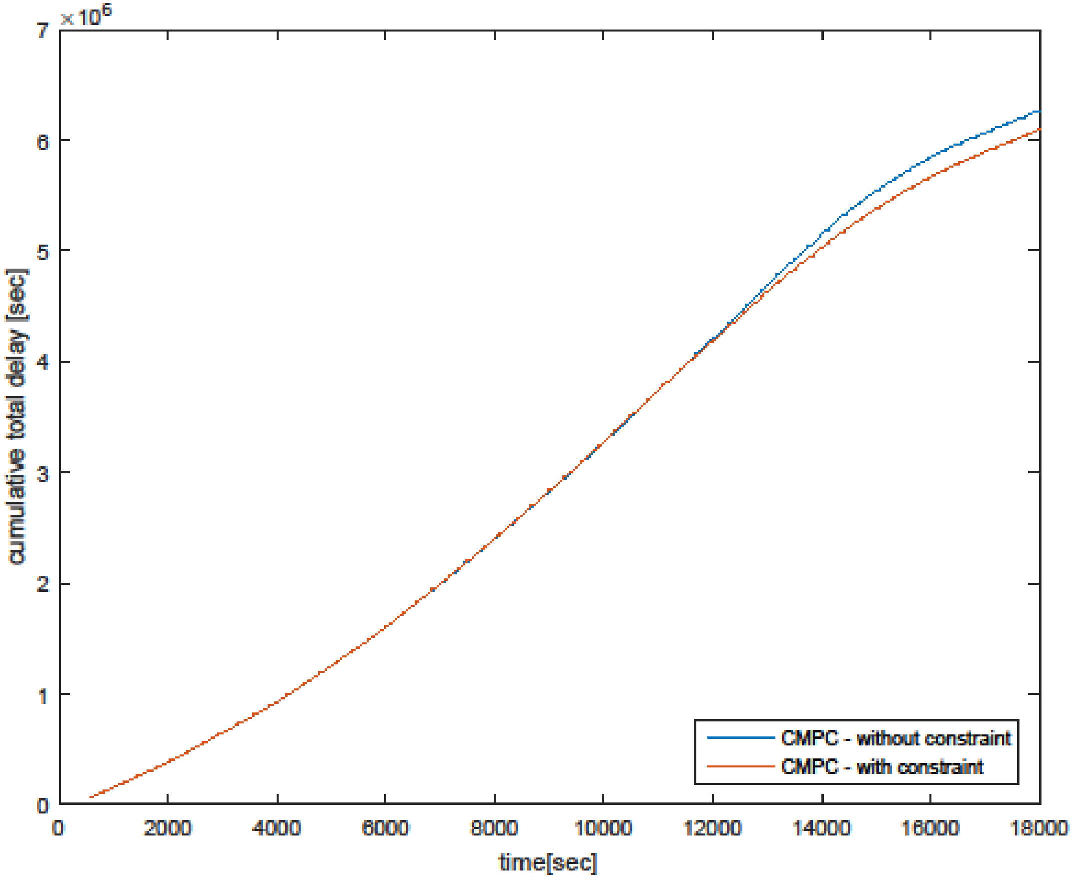
Stabilization constraint

- Adding this stabilization constraint
  “only allow scenarios that reduce
  local sum of squares of Q over one cycle,
  whenever large queues are present”
- is expected to improve performance and
  to guarantee stability (not proven yet)
- constraint overly restrictive for stability,
  but at least it is a distributed constraint

Performance comparison LMPC, CMPC, stabilized CMPC



Additional improvement thanks to stabilization constraint



Stabilized coordinated MPC

- Intuitive idea behind stabilization: uniformizing queues over all links can increase delay slightly, but avoiding to send traffic to already long queues avoids congestion that may spread

Future technological developments

- Added flexibility of autonomous/connected vehicles sending information on their current location/speed, receiving speed control signals from roadside controller (at each intersection) could increase maximal load where stable behaviour is possible by
  - avoiding waste of capacity due to yellow period
  - improving synchronization by platooning vehicles
  - further improve homogeneisation of traffic density
- Similar feedback control approach feasible?
- Can one calculate maximal performance improvement thanks to autonomous/connected vehicles?

Conclusions

- Proposed controller for switching times of traffic lights combines stabilization of backpressure control, with performance oriented distributed MPC
- Proposed controllers easy to adjust when structure of network changes
- Easy to extend methodology to
  - cases with priority vehicles,
  - modified cost function reflecting environmental impact
  - multimodal traffic

Questions?

for further info contact

rene.boel@ugent.be

haozhenzhenxd@gmail.com