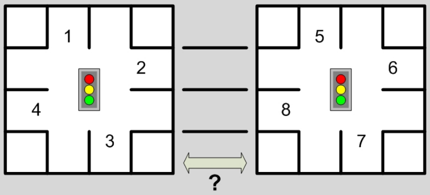
Urban traffic networks: identification, modeling and distributed control

Nicolae Marinica , René Boel
SYSTeMS Research Group, EESA Department
{NicolaeEmanuel.Marinica, Rene.Boel}@ ugent.be

Problem formulation

Urban traffic control poses a challenging problem in terms of coordinating the different traffic lights that can be used to influence the traffic flow.

The goal of this approach is to identify and to develop hybrid system models of controlled and uncontrolled intersections and links in urban traffic networks, based on formation of platoons. The other purpose is to develop a feedback control algorithm that optimizes the synchronization of signal timing plan by timing the traffic lights.



state = queues of waiting vehicles;

agent = local traffic controller from each intersection that has to send, to receive and to take local decisions.

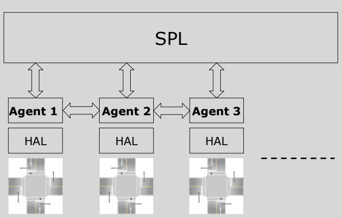
Each agent shall work under the supervision of an upper layer or in a decentralized mode.

Issues to solve:

- **Mathematical model**
  - online state estimation based on noisy and inconsistent information;
  - how to identify model parameters based on very inaccurate measurement data ?
  - what would be a convenient form of aggregation for representing the traffic?
  - how to include the delay introduced by each component?
- **Distributed control**
  - what is the amount and the structure of the information to communicate between agents?

Distributed control

We propose the following architecture :



SPL – Strategic Planning Layer
HAL – Hardware Abstraction Layer
Agent – autonomous agent that interacts with other agents through communication channels

action = the decision of an agent to switch the state of the traffic light or not.

The action of Agent 1 influences the evolution of the state of Agent 2 and vice-versa. This is the reason why the action of both actors has to be coordinated in such a way to minimize the wasted capacity.

Each agent has to know all local (virtual) queue size estimates and merging information from adjacent intersections may allow better estimate of true traffic flowing through boundary. Using communication among agents of neighboring intersection a distributed control strategy and sensor detection faults can be achieved.

What communication is necessary between agent $_i$ ↔ agent $_{i+1}$ and agent ↔ SPL ?

What should be the control law for each agent so that the agents cooperate efficiently?

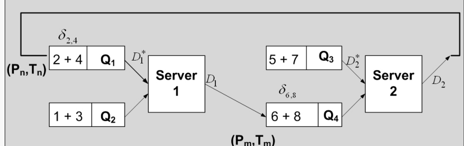
Mathematical model

What is an advantageous form to represent traffic for being used in a distributed control framework ?

It is necessary to find how to represent the delay and the uncertainty between the links of two or more intersections. The representation should yield a way to simulate network behavior at high computational speed, with enough detail.

We propose a **platoon based traffic flow model** introducing **platoons** (one or more vehicles moving together in a closely spaced formation) allows:
- "to aggregate vehicles in platoons" and "try to preserve the platoon formations";
- is a discrete event model that represents the arrivals of platoons of cars instead of arrival of individual vehicles;
- despite the fact that in case of a discrete event micro-simulator where the number of events would grow during congestion, our approach will actually bring a reduction of the number of events during congestion;
- the arrival streams defined at the sensor location boundary i at arrival rate $\lambda_i(t)$ platoons / time unit with a known distribution of platoon size that is dependent on traffic intensity;

Example: model of two intersections and the link in between :



Q1 when P_n entered on the link from 2 to 1 at time T_n:

$$Q_1(T_n+\delta^{+}_{2,4}) = Q_1(T_n+\delta^{-}_{2,4}) + P_n + noise$$

Q4 when P_m entered on the link from 1 to 2 at time T_m:

$$Q_4(T_m+\delta^{+}_{6,8}) = Q_4(T_m+\delta^{-}_{6,8}) + P_m + noise$$

Wasted capacity in serving the queues is:

$$W_1(t) = D_1^{*}(t) - D_1(t)$$

$$W_2(t) = D_2^{*}(t) - D_2(t)$$

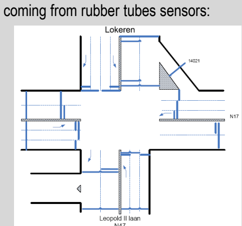
Where D is the number of served cars and D* is the maximum number of cars that can be served.

The necessary time for a platoon to pass the intersection is a stochastic variable with a given average time depending on the size of the platoon;

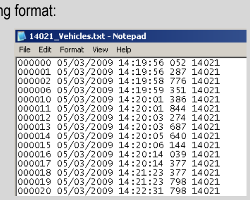
Identification

Unfortunately traffic control requires accurate state estimates especially when traffic load is heavy. It is necessary to use additional points for "resetting the queue length estimator" => extra sensors for long queues are necessary.

The row data is coming from rubber tubes sensors:



in the following format:



14021_Vehicles.txt - Notepad

```
000000 05/03/2009 14:19:56 052 14021
000001 05/03/2009 14:19:56 287 14021
000002 05/03/2009 14:19:58 776 14021
000006 05/03/2009 14:19:59 351 14021
000010 05/03/2009 14:20:01 386 14021
000011 05/03/2009 14:20:01 844 14021
000012 05/03/2009 14:20:03 274 14021
000013 05/03/2009 14:20:03 687 14021
000014 05/03/2009 14:20:05 640 14021
000015 05/03/2009 14:20:06 144 14021
000016 05/03/2009 14:20:14 039 14021
000017 05/03/2009 14:20:14 377 14021
000018 05/03/2009 14:21:23 377 14021
000019 05/03/2009 14:21:23 798 14021
000020 05/03/2009 14:22:31 798 14021
```

Correlation between the data from different sensors used for identification of the mathematical model.

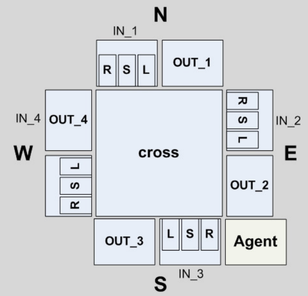
Develop a method:
- to categorize the cars;
- to extract the information from the data provided by sensors;
- anomalies;

What about the convergence of the sensors ?

Simulator

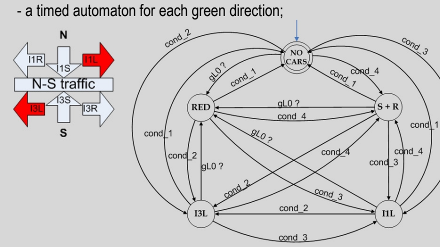
Architecture:
- Intersection = controllable shared resource among the arriving streams;
- Agent = can work in feedback control mode or fixed cycle;
- Link = uncontrollable component; sends messages about the estimation of the state (number of cars and the arrival time of each platoon);
- Events = generated by the transition between states with a random delay;

Intersection + agent :



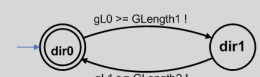
The model is the natural representation of how the cars move through an intersection.

The abstracted model of an intersection is the parallel composition :
- a timed automaton for each green direction;

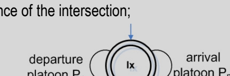


cond_1 : (gL1)! && ((I1S,I1L,I1R,I3S,I3L,I3R) == 0)!
cond_2 : (gL1)! && (not(I1S,I1R))! && (I3L > 0)!
cond_3 : (gL1)! && (not(I3S,I3R))! && (I1L > 0)!
cond_4 : (gL1)! && (I1S || I1R || I3S || I3R)!

- one timed automaton for the green light;



- one timed automaton for each buffer of each possible direction of each entrance of the intersection;



A link is modeled by introducing a stochastic delay for each platoon that entered on it. The delay depends on the queue length of cars that are at the output of that link and the number of cars that are already driving towards the output of the link.

1 hour of traffic of two connected intersection:

```
-- Intersection1        Simulation horizon: 3600 [s]
ans =                   ans =
   0  9  6  1           Time spent : 6.477469e+001 [s]
   0  0  0  0           ans =
   2  6  0  6           Origin C1N - Destination C1S: n=138 avrg_t=5.525903e+001
   3  0  1  0           Origin C1N - Destination C1W: n=250 avrg_t=5.564814e+001
ans =                   Origin C1N - Destination C1S: n=138 avrg_t=5.525903e+001
   0  0  0  0           Origin C1N - Destination C1W: n=250 avrg_t=5.564814e+001
   0  0  0  0           Origin C1N - Destination C2N: n=108 avrg_t=7.627378e+001
   0  0  0  0           Origin C1N - Destination C2E: n=97 avrg_t=7.504096e+001
   0  0  0  0           Origin C1N - Destination C2S: n=102 avrg_t=8.075670e+001
-- Intersection2        Origin C1E - Destination C1N: n=355 avrg_t=2.377350e+001
ans =                   Origin C1E - Destination C1S: n=184 avrg_t=2.580399e+001
   0  0  0  0           Origin C1E - Destination C1W: n=272 avrg_t=2.274210e+001
   0  0  0  0           Origin C1S - Destination C1N: n=273 avrg_t=2.160945e+001
   0  0  0  0           Origin C1S - Destination C1W: n=273 avrg_t=2.231973e+001
   1  0  1  0           Origin C1S - Destination C2N: n=57 avrg_t=3.308450e+001
ans =                   Origin C1S - Destination C2E: n=52 avrg_t=3.372046e+001
   0  0  0  0           Origin C1S - Destination C2S: n=65 avrg_t=3.530616e+001
   0  0  0  0           Origin C1W - Destination C1N: n=298 avrg_t=2.827732e+001
   0  0  0  0           Origin C1W - Destination C1S: n=298 avrg_t=2.674252e+001
   1  0  1  0           Origin C1W - Destination C2N: n=60 avrg_t=4.326126e+001
                        Origin C1W - Destination C2E: n=57 avrg_t=3.653364e+001
                        Origin C1W - Destination C2S: n=54 avrg_t=3.475646e+001
                        Simulation finished!
```