

Technical report 05-021

Multi-agent model predictive control of transportation networks*

R.R. Negenborn, B. De Schutter, and H. Hellendoorn

If you want to cite this report, please use the following reference instead:

R.R. Negenborn, B. De Schutter, and H. Hellendoorn, "Multi-agent model predictive control of transportation networks," *Proceedings of the 2006 IEEE International Conference on Networking, Sensing and Control (ICNSC 2006)*, Ft. Lauderdale, Florida, pp. 296–301, Apr. 2006.

Delft Center for Systems and Control
Delft University of Technology
Mekelweg 2, 2628 CD Delft
The Netherlands
phone: +31-15-278.51.19 (secretary)
fax: +31-15-278.66.79
URL: <http://www.dcsc.tudelft.nl>

*This report can also be downloaded via http://pub.deschutter.info/abs/05_021.html

Multi-Agent Model Predictive Control of Transportation Networks

Rudy R. Negenborn, Bart De Schutter, and Hans Hellendoorn

Abstract—We consider multi-agent, or distributed, control of transportation networks, like traffic, water, and power networks. These networks typically have a large geographical span, modular structure, and a large number of components that require control. We discuss the necessity of a multi-agent control setting in which multiple agents control parts of the network. As potential control methodology we consider Model Predictive Control (MPC) in a multi-agent setting. We first outline a framework for modeling transportation networks into subsystems using external variables and then discuss issues that arise when controlling these networks with multi-agent MPC. Several approaches to these issues are structured and discussed in terms of the outlined framework.

I. INTRODUCTION

Transportation networks, like power networks, traffic networks, and water networks, are usually large in size, consist of multiple subsystems, have many actuators and sensors, and therefore show complex dynamics. These transportation networks can be considered at a generic level, at which commodity is brought into the network at sources, flows over links to sinks, and is influenced in its way of flowing by elements inside the network. Control goals for transportation networks involve avoiding congestion of links, maximizing throughput, minimizing costs of control actions, etc.

In a single-agent control approach, a single-agent is used to control a system. The agent collects information from measurements of the system to determine which actions to take. However, the transportation networks we consider are hard to control by a single agent, since this agent would have to gather information from all sensors, and process this directly to provide inputs to all actuators. This is not only hard due to technical issues like communication delays and computational requirements, but also due to practical issues like unavailability of information from one subsystem to another and restricted control access. For transportation networks a multi-agent (or distributed) control approach [1] has to be employed in which several control agents, each with only limited information gathering and processing skills and moreover limited action capabilities, control the subsystems (e.g., subnetworks) of which the overall network is composed. Since the subsystems they control are part of an overall network, actions taken by one agent influence actions taken by other agents. The challenge for multi-agent control design is therefore to make the control agents work together such that the overall network performance is as desired.

Multi-agent control in general is not something new; it is in everyday use for the control of transportation networks. However, increasing quantitative and qualitative demands make ongoing research in multi-agent control approaches a necessity.

The authors are with the Delft Center for Systems and Control, Delft University of Technology, Delft, The Netherlands (e-mail: {r.negenborn,b.deschutter,j.hellendoorn}@dscs.tudelft.nl).

Already in 1978, Sandell *et al.* [2] surveyed a wide range of alternative methods for distributed control. They found that a good combination of engineering judgment and analysis can be used to define in a reasonable way an ad-hoc control structure for a dynamic system. However, instead of ad-hoc control structures we are interested in structured approaches applicable to multiple domains of transportation networks. Important aspects that play a role in this are the way in which a network is decomposed into subsystems, the way in which control problems are formulated on these subsystems, and the way in which agents communicate with one another to come to their decisions. Those issues are the topics of this paper.

We consider Model Predictive Control (MPC) [3]–[5] for controlling the transportation networks. MPC in a single-agent setting has shown successful application in the process industry over the last decades, and is now gaining increasing attention in fields like power networks [6], [7], and traffic networks [8]. MPC is based on solving at each decision step an optimization problem subject to system dynamics, an objective function, and constraints on states, actions, and outputs. At each decision step the optimization yields a sequence of actions optimizing expected system behavior over a certain horizon. Actions are implemented until the next decision step, after which the procedure is repeated. MPC has its success mainly due to its explicit way of handling constraints, its possibility to operate without intervention for long periods, and its built in robustness properties. Basic issues, e.g., stability and robustness, have extensively been studied for the single-agent case [5].

For controlling transportation networks in a multi-agent setting, in this paper we address some issues arising in MPC for multi-agent settings. These issues have started to be investigated over the last decades, e.g., by [9]–[21]. This paper is organized as follows. In Section II we introduce a framework for modeling the transportation networks and control objectives of our interest. In Section III we then discuss two important multi-agent control issues: how to predict subsystem evolution under limited information, and how to communicate and make decisions. We present several approaches to these issues in a structured way and relate current state-of-the-art methods for multi-agent MPC to these approaches in terms of the given framework. In Section IV we consider in more detail a particular method to obtain agreement between control agents. This exposition of topics gives insight into how multi-agent MPC can be applied to transportation networks.

II. NETWORK MODELING AND CONTROL FRAMEWORK

A. Subsystem-Based Network Model

To describe the transportation networks of our interest we will employ a subsystem-based network model, which

follows from a bottom-up approach in which interconnections between smaller subsystems are defined leading to the overall network. This way of modeling networks allows us to abstract from subsystem specific dynamics, while emphasizing the modularity of the overall network.

Consider n subsystems, e.g., subnetworks. Each of the subsystems $i \in \{1, \dots, n\}$ has its own dynamics, modeled by discrete-time dynamic and algebraic relations, i.e.,

$$x_{k+1}^i = f^i(x_k^i, y_k^i, u_k^i) \quad (1)$$

$$0 = g^i(x_k^i, y_k^i, u_k^i) \quad (2)$$

$$0 \leq h^i(x_k^i, y_k^i, u_k^i), \quad (3)$$

where at time step k , for subsystem i , x_k^i are the dynamic states, y_k^i are the outputs and algebraic states, and u_k^i are the inputs. These variables are referred to as local variables. Function f^i represents the dynamic state transitions, while functions g^i and h^i form algebraic and inequality constraints.

Interconnections between subsystems, e.g., due to transportation lines, give rise to external variables. A local variable of subsystem i 's model is an external variable if it corresponds to a variable in the model of another subsystem j . Using only the external variables all constraints between subsystems are expressed, see Fig. 1. We denote for each subsystem i the external variables by w^i , and add them to the subsystem i 's model by:

$$w_k^i = A^i [(x_k^i)^T (y_k^i)^T (u_k^i)^T]^T, \quad (4)$$

where T is the transpose, and A^i is of appropriate dimensions and contains on each entry of a row 0, except for a single 1 at the entry corresponding to a variable that is an external variable of subsystem i . The interconnecting constraints between the subsystems are then:

$$0 = A^{\text{inter}} [(w_k^1)^T \dots (w_k^n)^T]^T, \quad (5)$$

where A^{inter} is of appropriate dimensions and contains a single 1 and a single -1 on each row to indicate which external variables represent the same quantity; all other entries are 0.

Equations (1)–(4) for all subsystems together with (5) describe the dynamics of the overall network. When two subsystems have interconnecting constraints they are referred to as neighboring subsystems.

B. Multi-Agent Control Problem Formulation

Given the description of the subsystems and the interconnections between them, we assume that for each subsystem there is a control agent that selects which inputs to send to

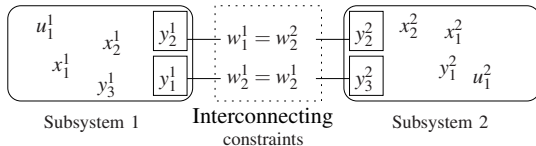


Fig. 1. Illustration of external variables and interconnecting constraints. Each subsystem has a set of variables. Some of these have corresponding variables in other subsystem models. External variables introduce these relations.

the actuators of that subsystem. Each control agent has direct access only to the sensors and actuators within its subsystem, and indirect access to external variables of neighbors through communication.

The control agents have to choose their actions such that desired behavior is obtained. This behavior is specified through objective functions and constraints. Behavior specified through an objective function implies soft desires, i.e., desires that should be achieved as well as possible, but need not necessarily be achieved. This type of behavior is based on costs of trajectories of local variables and external variables involved in the evolution of the subsystem over a horizon N . Desired behavior is formed by trajectories that have relatively low costs. Typically the objective function is taken as the summation of costs per step over the horizon:

$$J^i(\tilde{x}_{k+1}^i, \tilde{y}_k^i, \tilde{u}_k^i, \tilde{w}_k^i, \tilde{W}_k^i) = \sum_{j=0}^{N-1} [c^{\text{local},i}(x_{k+1+j}^i, y_{k+j}^i, u_{k+j}^i) + c^{\text{inter},i}(w_{k+j}^i, W_{k+j}^i)], \quad (6)$$

where $\tilde{\cdot}$ over variables indicates the respective variables over a horizon of N steps, e.g., $\tilde{u}_k^i = [(u_k^i)^T \dots (u_{k+N-1}^i)^T]^T$, \tilde{W}_k^i indicates the external variables of neighboring subsystems of subsystem i over the horizon, and $c^{\text{local},i}$ is a cost function for computing local costs, while $c^{\text{inter},i}$ computes costs due to interconnections. Behavior specified through constraints implies hard desires, i.e., desires that have to be met. These desires can thus be formulated using a function h_{des} as:

$$0 \leq h_{\text{des}}(x_k^i, y_k^i, u_k^i). \quad (7)$$

Since the variables that each control agent i can set consist of the inputs \tilde{u}_k^i to its subsystem, each agent effectively tries to solve the following optimization problem:

$$\min_{\tilde{u}_k^i} J^i(\tilde{x}_{k+1}^i, \tilde{y}_k^i, \tilde{u}_k^i, \tilde{w}_k^i, \tilde{W}_k^i), \quad (8)$$

subject to the desired constraints (7), the subsystem's dynamics and constraints (1)–(4), and the interconnecting constraints (5) involving subsystem i 's external variables over the horizon.

In a centralized setting, there would be only one system description, one objective function, one set of desired constraints, and no interconnecting constraints. Employing an MPC strategy comes down to solving the optimization problem, implementing the inputs for time step k , and moving on to the next decision step. An agent can make relatively adequate predictions about the evolution of the system over the horizon with all information it has access to, and therefore it can adequately determine the optimal inputs. Contrarily, in a multi-agent setting there are multiple subsystems, objective functions, and constraints, the evolution of each depending on the interconnecting constraints, and thus on neighboring external variables. In the next section we discuss how multi-agent MPC approaches deal with this.

III. MULTI-AGENT MPC

Extending single-agent MPC to multi-agent MPC involves knowing in advance that the subsystem model a control agent

has is inherently limited to a certain part of the overall network, which directly follows from the assumption that agents in a multi-agent control scheme have no global overview and can only access a relatively small number of sensors and actuators. Therefore predictions about the future evolution of external variables of neighbors may be inaccurate and feasibility may even be lost with the use of longer horizons. Communication is a means to improve this in multi-agent MPC, and is part of practically each approach, see, e.g., [12], [21], [22]. A general multi-agent MPC scenario for an agent i at time step k is outlined as follows, see also Fig. 2:

- 1) Make a measurement of the current state of the local subsystem x_k^i , and obtain information from neighboring agents through communication.
- 2) Solve optimization problem (8) to find the sequence of inputs \tilde{u}_k^i over the horizon that minimizes the local objective function subject to desired constraints (7), the local dynamics (1)–(4), interconnecting constraints (5), and predictions of the external variables of neighbors \hat{W}_k^i over the horizon based on the communicated information. During this optimization there may also be communication with other agents.
- 3) Implement the first input u_k^i of this sequence.
- 4) Perhaps communicate with other agents.
- 5) Move on to the next time step $k+1$.

The way in which external variables of neighbors are predicted over the horizon and the way in which communication and decision making is scheduled between agents define important characteristics of distributed MPC approaches and are considered in the following subsections.

A. External Variable Predictions

In order to determine the evolution of the local variables over a certain time horizon, a control agent i needs to have the current state of the system x_k^i , a candidate sequence of inputs \tilde{u}_k^i , and predictions of the evolution of the external variables \hat{W}_k^i of neighbors, see Fig. 3. The predictions of the evolution of the

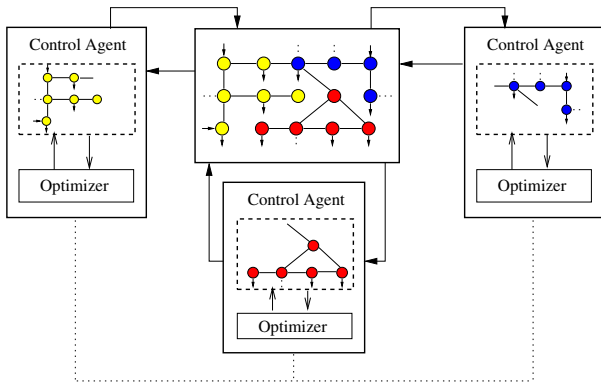


Fig. 2. Schematic example illustration of multi-agent MPC. Control agents control parts of the overall system. Each of the agents has a model of the subsystem it controls. Agents communicate with neighboring agents about external variables. Through an optimization procedure they decide which actions to implement on their subsystems.

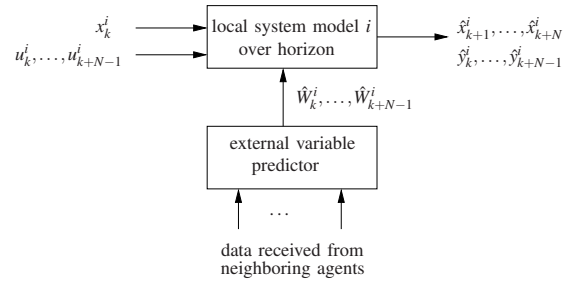


Fig. 3. Elements used by agent i to predict its subsystem's evolution.

external variables is based on the information communicated with the neighboring agents. An external variable predictor processes this information to come up with the predictions over the horizon. There are several options in which the external variable predictor can determine the predictions, briefly summarized as:

- 1) Ignore the influence of an external variable.
- 2) Assume a constant value over the whole horizon based on a local measurement made or obtained from a neighboring agent.
- 3) Assume predictions over the whole horizon as obtained from a neighboring agent [11], [12], [19].
- 4) Assume upper and lower boundaries on the external variables, as obtained from a neighboring agent.
- 5) Assume a model that predicts the values of the external variables based on dynamics of neighboring subsystems [21].
- 6) Learn a model about the evolution of the external variables given available information from neighboring agents.
- 7) Use knowledge about the objective function of neighboring agents together with models of the dynamics of the neighboring system [22].

Each of the options yields different information that has to be communicated, and requires different types of built-in skills.

Option 1 is used in a completely decentralized setting. A control agent ignores the presence of other subsystems completely. Basically, it employs its own conventional MPC approach. This type of control scheme can be used when constraints between external variables are weak or non-existing. The advantage is the lack of communication overhead, although if the interconnecting constraints turn out not to be negligible, control performance can degenerate.

Option 2 is used when a control agent can measure the values of external variables of interest or obtain their values through communication, and it assumes the values of these external variables to be valid over the whole horizon. This type of control may be useful when the external variables change slowly. This approach may also be used to on-line monitor the interconnecting constraints and switch to a different option when the constraints start changing significantly. E.g., when the interconnecting constraints do not change, control could be completely decentralized. When they do change, commu-

nication has to be introduced.

Option 3 is employed by an agent that obtains predictions of the external variables over the whole horizon. This is an ideal situation for the local agent, as long as the neighboring agent providing the predictions will make sure that they are correct. In practice, if the subsystem of the neighboring agent relies on other neighboring subsystems this will be difficult to ensure. Iterations as discussed in the next section are necessary.

Option 4 assumes that neighboring agents do not communicate exact trajectories, but instead bounds on the values of the external variables. By enforcing these bounds, the local agent can compute worst-case optimal inputs. The agent providing the bounds also has to make sure that its actual trajectory stays within the bounds it communicated. So-called compatibility constraints can do this for certain linear-time invariant systems [21].

When option 5 is used the local control agent knows the dynamics or part of the dynamics that generate the external variables [21]. This is, e.g., the case when the local agent has a copy of the subsystem models used by its neighbors. These models will depend on variables from the neighboring subsystems, like inputs, and perhaps external variables of neighbors of neighbors. The local agent can deal with these variables as with any external variables. The advantage of this option is that more about the structure of the external variables is known. A disadvantage can be increased computational time to determine the predictions.

Option 6 can be employed if the agent does not have a model of the subsystem that generates the external variables. Instead it may employ learning techniques and build up experience to learn a model.

Finally, option 7 can be used when the agent has information about the objectives and subsystem models of its neighboring agents. The agent can then use this to compute which actions the neighbors will take [22]. It can determine the actions that will be applied to that subsystem and consequently determine the evolution of the external variables. Knowledge about the objectives of neighboring systems can be used to make local decisions that are not counteracting the objectives of other subsystems. A disadvantage of this approach is that one controller effectively is solving the control problems of multiple subsystems. In an approach that somehow communicates the computed actions to the neighboring systems this could become an advantage however.

B. Communication and Decision-Making Schemes

The goal of communication is to inform neighbors and get information from neighbors that is necessary to choose the right actions. An agent can employ different schemes for communication and decision making, see Fig. 4:

- 1) It can receive information while deciding and only send directly after having decided (Fig. 4a).
- 2) It can receive and send after all agents have decided [19], [21], [22] (Fig. 4b).
- 3) It can receive after a neighboring agent has decided and send after deciding on its own actions (Fig. 4c).

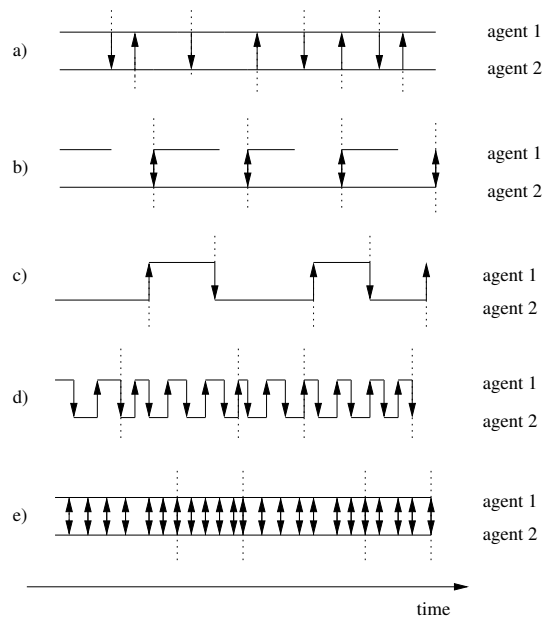


Fig. 4. Different communication schemes between two agents. Arrows are information exchange. Dotted lines are actions being implemented. Horizontal lines are optimization problems being solved.

- 4) It can receive and send a number of times before deciding when all agents are ready to do so [11], [12], [20] (Fig. 4d,e).

Scheme 1 forms the ideal case, since agents do not have to wait for other agents to finish their optimization problems. Also, the amount of communication between agents is minimal, since it only takes place after an optimization. However, agents will have to include newly received information from neighboring agents at any time while solving their own optimization problems. No multi-agent MPC methods can do this at present.

In Scheme 2 the agents solve their problems at the same time as well. The amount of communication is low, since agents wait with exchanging information and implementing their action until all agents have finished their optimization problems. This however is a disadvantage for fast agents, since they will have to wait for the slow agents. Moreover, since communication is done afterward, it is delayed, which may prevent convergence to a feasible overall solution. For certain linear time-invariant systems stability can be proven when a contracting so-called stability constraint is placed on the first state of each subsystem [19]. Stability results for settings where the evolution of external variables does not depend on neighbors of neighbors are given in [21], [22].

In Scheme 3 only one of the agents solves its optimization problem at a time. At the end of its optimization, it implements its decision, and sends information to a next agent. Clearly, also in this scheme a disadvantage is that agents have to wait until it is their turn to solve their optimization problem.

In Scheme 4 decisions are made after a sequence of optimization-communication steps. The iterations are a way for the agents to obtain agreement about the values of the

external variables over the horizon. The iterations continue until convergence is obtained or until a maximum number of iterations is reached. This scheme requires more communication and computation resources than the previous schemes, however, it may result in globally feasible solutions. A distinction can again be made between serially working agents [12], [20] and parallel working agents [11]. Conditions for convergence of iterations to local solutions and global solutions are given in [12] for the case of serially working agents. A Lagrangian-based scheme for the parallel case is employed in [11]. In the next section we consider in more detail this type of scheme.

IV. LAGRANGIAN-BASED MULTI-AGENT MPC

In this section we discuss a specific communication and decision scheme that falls in the Scheme 4 class as discussed in Section III-B: it is a scheme in which a number of optimization-communication iterations is performed before agents decide on their actions. In particular, the scheme is based on parallel working agents. Moreover, each agent uses Option 3 of Section III-A to determine the values of the neighboring external variables: each agent receives predictions of these external variables of its neighboring agents through communication.

A. Conceptual Description

For feasible overall solutions, the interconnecting constraints as defined in (5) need to be satisfied at all times. As discussed above, when one agent solves its optimization problem it needs to assume a trajectory for the external variables of its neighboring subsystems over the horizon. As discussed above as well, it is likely that an assumed trajectory will not appear in the true system evolution if not all agents having external variables relating to the assumed trajectory keep their respective assumed trajectories. They would only have an incentive to stick to their communicated trajectories if these trajectories yield optimal inputs for their own subsystems.

Even if the agents make an agreement in advance to stick to the trajectories communicated, in practice they may not be able to stick to this agreement, since at the time of trajectory generation the agents did not know what the values of the external variables of the other agents will be and may therefore require infeasible inputs to local subsystems to stick to the communicated trajectories. Instead of holding on to the first trajectories communicated agents perform a number of iterations to come to an agreement on external variable trajectories that are optimal to all agents.

In each iteration each agent optimizes both over its local inputs and over the predictions of neighboring external trajectories. In this way, each agent is sure that the predicted trajectories it assumes are optimal for its own subsystem. After each of the agents has in this way determined its own optimal inputs and predicted external variables trajectory, it communicates the predicted external variable trajectories to the neighboring agents. This basically means that each agent tells its neighboring agents how it would like to see the external variables of those agents to evolve over the horizon.

Ideally, the external variable trajectories that those neighboring agents receive will exactly correspond to their predictions of their external variable trajectories if they would implement their optimal input sequences. However, it is more likely that the received trajectories will not correspond to their predicted trajectories, as discussed before. To encourage the agents to come to an agreement on the predicted external variable trajectories a penalty term is added to the objective function of each agent. By adjusting the penalty terms over a series of iterations, convergence can be obtained.

B. Algorithmic Description

The conceptual ideas described above are implemented by the following algorithm which is the result of a Lagrangian relaxation [23]. We consider two agents, i and j , that have to agree on their external variable trajectories.

- 1) Both agents initialize penalty terms λ_0^i, λ_0^j for each of their external variables to arbitrary but similar values, i.e., $\lambda_0^i = \lambda_0^j$, and set iteration counter s to 0.
- 2) The agents solve their own optimization problems in parallel. I.e., Agent i solves at step k , iteration s :

$$(\tilde{u}_{k|s+1}^i, \tilde{w}_{k|s+1}^i) = \arg \min_{\tilde{u}_{k|s+1}^i, \tilde{w}_{k|s+1}^i} J^i(\tilde{x}_{k+1}^i, \tilde{y}_k^i, \tilde{u}_k^i, \tilde{w}_k^i) + (\lambda_s^i)^T \tilde{w}_k^i,$$

subject to (1)–(4), while Agent j solves:

$$(\tilde{u}_{k|s+1}^j, \tilde{w}_{k|s+1}^j) = \arg \min_{\tilde{u}_{k|s+1}^j, \tilde{w}_{k|s+1}^j} J^j(\tilde{x}_{k+1}^j, \tilde{y}_k^j, \tilde{u}_k^j, \tilde{w}_k^j) - (\lambda_s^j)^T \tilde{w}_k^j,$$

subject to (1)–(4).

- 3) After the agents have solved their problem and found inputs and external variable trajectories, the agents exchange external variable trajectories. I.e., Agent i sets $\hat{W}_{k|s+1}^i$ to $\tilde{w}_{k|s+1}^j$ as received from Agent j , while Agent j sets $\hat{W}_{k|s+1}^j$ to $\tilde{w}_{k|s+1}^i$ as received from Agent i .
- 4) After this, Agent i updates its λ^i terms as:

$$\lambda_{s+1}^i = \lambda_s^i + \varepsilon(\tilde{w}_{k|s+1}^i - \hat{W}_{k|s+1}^i),$$

where ε is a small positive real number, while Agent j performs the update:

$$\lambda_{s+1}^j = \lambda_s^j + \varepsilon(\hat{W}_{k|s+1}^j - \tilde{w}_{k|s+1}^j).$$

- 5) The agents increase iteration counter s and continue with step 2, until convergence of the external variables is obtained or a maximum number of iterations has been reached.

It is easy to verify that the λ updates encourage agents to converge to agreement. Intuitively, the term penalizes the value of the external variables that an agent broadcasts to its neighbors. The term is adjusted in such a way that when the value of the external variables of the agent itself are higher than a neighbor asks them to be, the term is increased, such that the agent will lower its values for the external variables. On the contrary, when the values of the agent itself are lower than requested by its neighbors, the term is decreased, such that it is less costly to increase the value of its external variables. For strongly convex problems the agents are guaranteed to converge to the overall optimal solution [23].

V. CONCLUSIONS AND FUTURE RESEARCH

Control of transportation networks that can be found in traffic, power, and water management, requires multi-agent control approaches. Model Predictive Control (MPC) in a multi-agent setting is proposed for this task. In such a multi-agent setting each control agent employs MPC to control its subsystem by making predictions about the evolution of its subsystem over a horizon. Due to the interconnecting constraints between subsystems making predictions about the evolution of an agent's subsystem by an individual agent is hard. Communication between agents has to be employed to improve the predictions and subsequently improve the decision making. What information to communicate, when to do this, and when to decide on which actions to take are main issues in multi-agent MPC.

In this paper we have introduced a framework for modeling networks by considering them as consisting of subsystems with external variables that define interconnecting constraints. We have discussed several ways in which agents can make predictions about the evolution of their subsystems. These ways differ in how information from neighboring agents is used to make predictions about the evolution of the external variables. Moreover, we have considered several communication and decision-making schemes that structure in which order agents communicate and at what time they decide to take actions. Finally, we have elaborated on a particular multi-agent MPC scheme that is based on Lagrangian multipliers, and we have given both a conceptual and algorithmic description of this scheme.

Some of the options for taking into account information from neighbors and for communicating and decision making have been published before, mostly in the context of networks consisting of linear time-invariant subsystems. Future research consists of extending these results to networks with nonlinear dynamics, both involving continuous and discrete elements. Moreover, comparisons will be made between the different ways of making external variable predictions and the different communication and decision-making schemes. These comparisons are necessary to further clarify how multi-agent MPC can optimally be used for control of transportation networks.

ACKNOWLEDGMENTS

This research was supported by project "Multi-agent control of large-scale hybrid systems" (DWV.6188) of the Dutch Technology Foundation STW, Applied Science division of NWO, and the European 6th Framework Network of Excellence "HYbrid CONtrol: Taming Heterogeneity and Complexity of Networked Embedded Systems (HYCON)", contract number FP6-IST-511368.

REFERENCES

- [1] D. D. Siljak, *Decentralized Control of Complex Systems*. Boston: Academic Press, 1991.
- [2] N. R. Sandell, jr., P. Varaiya, M. Athans, and M. G. Safonov, "Survey of decentralized control methods for large scale systems," *IEEE Transactions on Automatic Control*, vol. 23, no. 2, pp. 108–128, Apr. 1978.
- [3] E. Camacho and C. Bordons, *Model Predictive Control in the Process Industry*. Berlin, Germany: Springer-Verlag, 1995.
- [4] J. Maciejowski, *Predictive Control with Constraints*. Harlow, England: Prentice Hall, 2002.
- [5] M. Morari and J. H. Lee, "Model predictive control: past, present and future," *Computers and Chemical Engineering*, vol. 23, pp. 667–682, 1999.
- [6] T. Geyer, M. Larsson, and M. Morari, "Hybrid emergency voltage control in power systems," in *Proceedings of the European Control Conference*, Cambridge, England, Sept. 2003.
- [7] P. Hines, L. Huaiwei, D. Jia, and S. Talukdar, "Autonomous agents and cooperation for the control of cascading failures in electric grids," in *Proceedings of the 2005 IEEE International Conference on Networking, Sensing and Control*, Tucson, Arizona, Mar. 2005, pp. 273–278.
- [8] A. Hegyi, B. De Schutter, and J. Hellendoorn, "Optimal coordination of variable speed limits to suppress shock waves," *IEEE Transactions on Intelligent Transportation Systems*, vol. 6, no. 1, pp. 102–112, Mar. 2005.
- [9] M. W. Braun, D. E. Rivera, M. E. Flores, W. M. Carlyle, and K. G. Kempf, "A model predictive control framework for robust management of multi-product, multi-echelon demand networks," *Annual Reviews in Control*, vol. 27, pp. 229–245, 2003.
- [10] M. R. Katebi and M. A. Johnson, "Predictive control design for large-scale systems," *Automatica*, vol. 33, no. 3, pp. 421–425, 1997.
- [11] G. Georges, "Decentralized adaptive control for a water distribution system," in *Proceedings of the 3rd IEEE Conference on Control Applications*, Glasgow, UK, 1999, pp. 1411–1416.
- [12] E. Camponogara, D. Jia, B. H. Krogh, and S. Talukdar, "Distributed model predictive control," *IEEE Control Systems Magazine*, vol. 1, pp. 44–52, Feb. 2002.
- [13] M. Aicardi, G. Casalino, R. Minciardi, and R. Zoppoli, "On the existence of stationary optimal receding-horizon strategies for dynamic teams with common past information structures," *IEEE Transactions on Automatic Control*, vol. 37, pp. 1767–1771, Nov. 1992.
- [14] L. Acar, "Some examples for the decentralized receding horizon control," in *Proceedings of the 31st IEEE Conference on Decision and Control*, Tucson, Arizona, 1992, pp. 1356–1359.
- [15] S. Sawadogo, R. M. Faye, P. O. Malaterre, and F. Mora-Camino, "Decentralized predictive controller for delivery canals," in *Proceedings of the 1998 IEEE International on Systems, Man, and Cybernetics*, San Diego, California, 1998, pp. 3380–3384.
- [16] H. El Fawal, D. Georges, and G. Bornard, "Optimal control of complex irrigation systems via decomposition-coordination and the use of augmented lagrangian," in *Proceedings of the 1998 IEEE International on Systems, Man, and Cybernetics*, San Diego, California, 1998, pp. 3874–3879.
- [17] M. Gomez, J. Rodellar, F. Veja, J. Mantecon, and J. Cardona, "Decentralized predictive control of multireach canals," in *Proceedings of the 1998 IEEE International on Systems, Man, and Cybernetics*, San Diego, California, 1998, pp. 3885–3890.
- [18] M. Baglietto, T. Parisini, and R. Zoppoli, "Neural approximators and team theory for dynamic routing: A receding-horizon approach," in *Proceedings of the 38th IEEE Conference on Decision and Control*, Phoenix, Arizona, 1999, pp. 3283–3288.
- [19] D. Jia and B. H. Krogh, "Distributed model predictive control," in *Proceedings of the 2001 American Control Conference*, Arlington, Virginia, June 2001, pp. 2767–2772.
- [20] D. Jia and B. Krogh, "Min-max feedback model predictive control for distributed control with communication," in *Proceedings of the 2002 American Control Conference*, Anchorage, Alaska, May 2002, pp. 4507–4512.
- [21] W. B. Dunbar and R. M. Murray, "Model predictive control of coordinated multi-vehicle formations," in *Proceedings of the 41st IEEE Conference on Decision and Control*, Las Vegas, Nevada, Dec. 2002, pp. 4631–4636.
- [22] T. Keviczky, F. Borrelli, and G. Balas, "A study on decentralized receding horizon control for decoupled systems," in *Proceedings of the 2004 American Control Conference*, Boston, Massachusetts, 2004.
- [23] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Nashua, New Hampshire: Athena Scientific, 1997.