SEVENTH FRAMEWORK PROGRAMME
THEME – ICT
[Information and Communication Technologies]

SEVENTH FRAMEWORK PROGRAMME

Contract Number:	223854
Project Title:	Hierarchical and Distributed Model Predictive Control of Large-Scale Systems
Project Acronym:	HD-MPC

HD−MPC

Deliverable Number:	D3.1.4 - Software
Deliverable Type:	Demonstrator
Contractual Date of Delivery:	June 1, 2011
Actual Date of Delivery:	**June 1, 2011**
Title of Deliverable:	**Software tool with different methods and variants for different problem classes**
Dissemination level:	Restricted
Workpackage contributing to the Deliverable:	WP3
WP Leader:	Wolfgang Marquardt
Partners:	RWTH, POLIMI, TUD, USE, UNC, SUPELEC
Author(s):	H. Scheu, M.D. Doan, A. Núñez, F. Valencia

Table of contents

Project co-ordinator

Name: Bart De Schutter
Address: Delft Center for Systems and Control
Delft University of Technology
Mekelweg 2, 2628 Delft, The Netherlands
Phone Number: +31-15-2785113
Fax Number: +31-15-2786679
E-mail: b.deschutter@tudelft.nl

Project web site: `http://www.ict-hd-mpc.eu`

Executive Summary

We provide a software package including distributed model predictive control methods, developed in the project. This documentation contains only a brief description of the files attached. For a detailed description of the methods, references to detailed research articles are given. The descriptions can also be found in recent deliverables of the HD-MPC project related to WP3. See the official webpage `http://www.ict-hd-mpc.eu/` for the corresponding downloads of the reports.

In Chapter 1 the software for the “Distributed model predictive control using Fenchel’s duality” is described. The underlying decomposition technique relies on Fenchel’s duality and allows subproblems to be solved using local communications only. We investigate two techniques aimed at improving the convergence rate of the iterative approach. The distributed algorithms are illustrated with a numerical example, the Matlab code is provided so that users can reuse the code for different settings.

Chapter 2 contains descriptions for the example files of “Feasible-cooperation distributed model predictive control scheme based on game theory”. Distributed Model Predictive Control (DMPC) can be interpreted as a game in which the players are the subsystems, the actions are the control inputs, and the payoff of each subsystem is given by the value of its cost function. The approach is implemented for the four-tanks benchmark as well as a Hydro Power Valley problem.

In Chapter 3 the software for a dynamic pickup and delivery problem based on hierarchical multi-objective model predictive control approach is explained. In the approach, the hierarchical multilayer structure of the system is used to decompose the optimization problem, which is big and NP-hard, into smaller but more tractable subproblems. Each proposed layer represents the viewpoint of different decision-makers. In one of those layers, the dispatcher routes the vehicles when a new request appears, and minimizes user and operator costs. As those two components are usually aimed at opposite goals, the problem in this layer is formulated and solved through multi-objective model predictive control. The dispatcher participates in the dynamic routing decisions by expressing his/her preferences in a progressively interactive way, seeking the best trade-off solution at each instant among the Pareto optimal set.

Chapter 1

Distributed model predictive control using Fenchel's duality

The Matlab code in the folder:
`/Distributed_MPC_with_Han_method`
is an implementation of the Distributed Model Predictive Control (DMPC) algorithms that were presented by Doan et al. in the paper [1].

Here are some short notes about how to use the files in this folder:

- init_model_canal.m: initialize the model and the optimization problem for the MPC problem
- test_DMPC_Han_method.m: run the simulation of the DMPC algorithms
- @intermodel: this subfolder describes an instrumental object for describing the subsystem model
- lookupcell.m: another instrumental function

Parameters for the problem are given at the beginning of the two files: init_model_canal.m and test_DMPC_Han_method.m.

Run the script test_DMPC_Han_method.m for the simulation.

Chapter 2

Feasible-cooperation distributed model predictive control scheme based on game theory

The methods can be found in the three folders:

1. `/DMPC_4T_BENCH_09062010`
2. `/DMPC_4T_SYM_28022011`
3. `/DMPC_HPV_23052011`

Each folder contains the same method, however, for different applications. While the first two folders contain the benchmark on the quadruple tank system, the last one deals with the Hydro Power Valley (HPV) Benchmark. A description of the quadruple tank system and theHPV benchmark can be found on the official HD-MPC webpage `http://www.ict-hd-mpc.eu/`.

Each folder has a file `inicio*` with the parameters and global variables for the simulation, a file `fcmpc*` which is the proposed DMPC algorithm, a file `const*` with the constraints, and a Simulink model with the simulation of the plant.

The file `fcmpc*` has three parts:

1. The initialization: in this stage the counters used along the algorithm are set at their initial values.
2. Reference and cost function matrices: in this stage the matrices H and f coming from the quadratic cost of the MPC problem are computed for each subsystem.
3. Control input and disagreement point: in this stage the negotiation model is implemented and the disagreement point is updated.

In the approach presented, it is assumed that all subsystems exchange the information of the control inputs and the current values of the states, then these are the inputs of the function `fcmpc*`. A more detailed description of the algorithms and the proposed method can be found in the papers [2, 3].

Chapter 3

HMO-MPC for Dial-a-ride systems

3.1 Process description

Dial-a-ride systems (DARS) are transit services which provide a shared-ride door-to-door service with flexible routes and schedules. The quality of service of a DARS is supposed to be in between of public transit buses and taxis. The typical specifications are the users pickup and delivery destinations and desired pickup or delivery times. We will assume that all the requests are known only after the dispatcher receives the associated call and that all the users want to be served as soon as possible. Thus, even we will not include explicitly hard windows, to provide a good service we propose a user-oriented objective function that deals with the problem of undesired assignments to clients, and keeps the service provided as regular (stable) as possible.

The service demand η_k comprises the information of the request and is characterized by two positions, pickup p_k and delivery d_k, the instant of the call t_k, a label r_k that identifies the passenger who is calling and the number of passengers waiting there Ω_k. Also we consider the expected minimum arrival time tr_k which is the best possible time to serve the passenger, considering a straight journey from origin to destination (like a taxi service) and a waiting time obtained with the closest available vehicle (in terms of capacity) to pick up that passenger.

We assume a fixed and known fleet size F over an urban area A. The dispatcher receives calls asking for service every instant k. Once a new request enters the system, the assignment of the vehicle and the insertion position of the new request into the previous sequence of that vehicle, are control actions decided by the dispatcher (controller), based on a dynamic objective function. Then, at any instant k, each vehicle j is assigned to complete a sequence of tasks which includes several points of pickup and delivery. Only one of those vehicles will serve the last new request. The set of sequences $u(k) = S(k) = \{S_1(k), \ldots, S_F(k)\}$ correspond to the control variable. The sequence of stops assigned to vehicle j at instant k is given by $S_j(k) = \left[S_j^0(k), S_j^1(k), \ldots, S_j^{w_j(k)}(k)\right]$, where $S_j^i(k)$ is the information about the i-th stop and $w_j(k)$ is the number of planned stops of vehicle j at instant k. The i-th stop information comprises the label of the user $r_j^i(k)$, the spatial coordinate $P_j^i(k)$, whether the stop is a pickup or delivery $z_j^i(k)$ and the number of users waiting at the i-th stop $\Omega_j^i(k)$.

Two sources of stochasticity are considered: the first regarding the unknown future demand entering the system in real-time, and the second coming from the network traffic conditions, in its spatial and temporal dimension represented by a speed distribution $v(t,p)$ at instant t in a position p. We will assume in this work a conceptual network, where the trajectories are defined as the straight line that joins two consecutive stops. Besides, a speed distribution for the urban zone during a typical period represented by a speed model $\hat{v}(t,p)$ is supposed to be known, which could be obtained from historical data.

The details of the proposed HMO-MPC can be found in [4] and [5]. In summary the scheme considers three layers. In the first one, variables with a long-term effect in the system are determined. In the second layer, the vehicles are characterized by fuzzy membership functions, which are used in the next layer to optimize in a better way the fleet. The last layer consists of a MO-MPC problem. Under the implemented on-line system it is easier and transparent for the operator to follow service policies as weighting parameters are not tuned. Next the codes of the implementation using Matlab are explained.

3.2 Codes description

The corresponding code referring to this work can be found in the folder:
`/Routing MO3`

The main code is the file $MAIN.m$. In the first part the code $generator\_calls.m$ generate the sequence of calls for the simulation, following trip patterns and probabilities provided in the program.

$calls = generator\_calls(tautasa, ncall, ploplo)$ is a matrix of random requests, where the rate is $tautasa$, $ncall$ the number of calls, and the variable $ploplo$ equals 1 if we don't want to see figures and 0 otherwise.

With "calls" data we extract the trip patterns and their probabilities. The function $trip\_patterns.m$ provide the patterns and their probabilities using a classical hard partition, and also the parameters of a fuzzy partition. $[probb\_fuzzy, probb\_clas] = trip\_patterns(calls, nclus, ploplo, threshold, parx, pary)$, where $nclus$ is the number of patterns we want from the data, $ploplo$ equals 1 if we want to see the membership function of the data, the calls with a membership degree lower than the $threshold$ are discarded, $parx$ and $pary$ define the hard partition axis x and y.

To obtain some statistics, we save the information of each passenger in the global variable $users$. The k row has the information of the passenger number k, where $r(k)$ is the label of user k, $v_j(k)$ the vehicle assigned to user k, t_k the instant when user k calls, $\hat{(T)}_{pick}$ the expected pickup time when user is assigned, $\hat{T}_{deli}$ the expected delivery time when user is assigned, T_{pick} is the effective pickup time, T_{deli} is the effective delivery time, $tr(k)$ is the best delivery time possible for user k (when the call occurs). $users(k) = [r(k), v_j(k), t_k, \hat{T}_{pick}, \hat{T}_{deli}, T_{pick}, T_{deli}, tr(k)]$

In the variable global $fleet$ the row i is the information of vehicle i, comprising the total traveled distance, dead times (vehicle are stopped) and time traveling. The variables $theta1$, $theta2$, $theta3$ and $theta4$ are the cost of travel time, cost of waiting time, cost of time for vehicle, and the cost of distance for a vehicle respectively. The variable tau is the predicted time minutes. The variable F is the fleet size, we assume this fleet size is fix for the whole simulation time. For each vehicle i in the variable $vei, 1$ we have the information of sequences. Consider a warm up and cold down period to not consider in the statistic $a = F + 5$, $b = ncall - 10$.

The variables $modepareto$ and $lalamba$ includes some ways we model the dispatcher. If $Modepareto$ equals 1 then a maximum cost for users is the criterion to use, if it equals 2 a maximum cost for operator, with 3 we use $lalamba * Jusario + (1 - lalamba) * Joperador$, if $Modepareto$ equals 4 a maximum effective user cost $theta1 * t\_effective\_waiting + theta2 * t\_effective\_travel$ is considered, and if it is equal to 5 then the selection of the solution is done manually, step by step.

The global variable ob is 1 if we use the user oriented objective function, and equals 2 if we use the total time oriented objective function. The variable N is the prediction horizon. Even for short prediction horizon, the time traveled are predicted usually in a 30 minutes horizon.

The parameters of the Layer 2, the parameters of the membership functions are defined (mean and standard deviation of the Gaussian membership function).

Finally, the simulation is run with the program $[X, costos] = routing\_hmo\_mpc(calls, prob, ve)$.

Bibliography

[1] M. D. Doan, T. Keviczky, B. De Schutter, An iterative scheme for distributed model predictive control using Fenchel's duality, Journal of Process Control 21 (5) (2011) 746–755. doi:10.101/j.jprocont.2010.12.009.

[2] I. Alvarado, D. Limon, D. Muñoz de la Peña, J. Maestre, M. Ridao, H. Scheu, W. Marquardt, R. Negenborn, B. De Schutter, F. Valencia, J. Espinosa, A comparative analysis of distributed mpc techniques applied to the hd-mpc four-tank benchmark, Journal of Process Control 21 (5) (2011) 800 – 815, special Issue on Hierarchical and Distributed Model Predictive Control. doi:10.1016/j.jprocont.2011.03.003.

[3] F. Valencia, J. Espinosa, B. De Schutter, K. Staňková, Feasible-cooperation distributed model predictive control scheme based on game theory, in: 18th IFAC world congress, 2011.

[4] A. Núñez, C. Cortés, D. Sáez, M. Gendreau, B. De Schutter, Multiobjective model predictive control applied to a dial-a-ride system, in: Proceedings of the 90th Annual Meeting of the Transportation Research Board, Washington, DC, 2011, paper 11-1942.

[5] A. Núñez, B. De Schutter, D. Sáez, C. Cortés, Hierarchical multiobjective model predictive control applied to a dynamic pickup and delivery problem, in: Proceedings of the 13th International IEEE Conference on Intelligent Transportation Systems (ITSC 2010), Madeira Island, Portugal, 2010, pp. 1553–1558. doi:10.1109/ITSC.2010.5625193.