Proceedings of the 18th World Congress
The International Federation of Automatic Control
Milano (Italy) August 28 - September 2, 2011

Predictive Control for Multi-Robot Observation of Multiple Moving Targets Based on Discrete-Continuous Linear Models

Juliane Kuhn *, **Christian Reinl** *, **Oskar von Stryk** *

* *Technische Universität Darmstadt, Germany*
(e-mail: {kuhn,reinl,stryk}@sim.tu-darmstadt.de)

Abstract: The observation of multiple moving targets by cooperating mobile robots is a key problem in many security, surveillance and service applications. In essence, this problem is characterized by a tight coupling of target allocation and continuous trajectory planning. Optimal control of the multi-robot system generally neither permits to neglect physical motion dynamics nor to decouple or successively process target assignment and trajectory planning.
In this paper, a numerically robust and stable model-predictive control strategy for solving the problem in the case of discrete-time double-integrator dynamics is presented. Optimization based on linear mixed logical dynamical system models allows for a flexible weighting of different aspects and optimal control inputs for settings of moderate size can be computed in real-time.
By simulating sets of randomly generated situations, one can determine a maximum problem size solvable in real-time in terms of the number of considered robots, targets, and length of the prediction horizon. Based on this information, a decentralized control approach is proposed.

Keywords: Decentralized control; Control design for hybrid systems; Real-time control

1. INTRODUCTION

The cooperative multi-robot observation of multiple moving targets (CMOMMT, Parker and Emmons (1997)) is a key problem in many security, surveillance, and service applications. As it is inherently cooperative and scalable, it is a well-accepted, NP-hard benchmark problem for investigating situation-based allocation of roles and subtasks as well as the determination of vehicle-specific trajectories (Parker (2002); Luke et al. (2005); Ding et al. (2006); Markov and Carpin (2007)). Most applications require decentralized robust online control strategies which allow real-time adaptation to a dynamic environment.

Although the CMOMMT problem is characterized by a tight coupling of discrete decisions (target assignment) and continuous trajectory planning, these aspects are usually considered subsequently. This raises, in general, a significant loss of optimality. Most existing approaches for task-allocation in cooperative, autonomous multi-robot systems are based on heuristics like market-based methods or behavior-based decision rules (e.g. Parker (1998); Risler (2009)). Therefore these approaches do not guarantee any optimality.

In order to apply optimization-based control methods, a model of robots and targets within the considered environment is needed. Modeling a cooperative control problem that allows for continuous as well as discrete variables and logical rules results in a possibly nonlinear *Hybrid Optimal Control Problem* (Glocker et al. (2006)). In general, solving this problem is computationally very expensive. A discrete-time linear model approximation simplifies the problem. Therefore, many optimization-based control strategies use highly efficient *Mixed Integer Linear Programs* (MILP) to approximate the real system. This strategy has already been applied to some specific examples of cooperative multi-robot games (cf. e.g. Earl and D'Andrea (2007); Reinl and von Stryk (2007)) and multi-vehicle coordination problems (e.g. Richards and How (2004)).

Modeling frameworks for linear hybrid dynamical systems – particularly *mixed logical dynamical* (MLD) and *piecewise affine* (PWA) systems – were studied extensively in control theory within the last decade (cf. Bemporad and Morari (1999); Morari and Baric (2006)). Especially when combined with a model-predictive control (MPC) strategy, the MLD framework provides a powerful tool for a wide range of modeling and control tasks. MPC, in general, combines optimality with robust, stabilizing control and various software tools were developed for controller synthesis (e.g. Kvasnica (2008)).

As the applicability of MPC based on MLD systems is hardly investigated for simultaneously planning discrete decisions and continuous trajectories in cooperative multi-robot systems (as in Fierro and Wesselowski (2004)), this paper presents a new MLD-based decentralized, numerically robust, and stable MPC strategy for the problem of CMOMMT.

A MLD formulation is used to model the multi-vehicle systemand an objective function, which permits flexible weighting of different aspects, is set up (Section 3). At each time step, the resulting optimal control problem is transformed into a MILP before being solved. When considering the entire system, global optimality of the resulting control inputs is guaranteed. By investigating randomly generated situations, a maximum real-time computable problem size is determined in terms of the number of robots, targets, and length of the prediction horizon. Based on this information, the decentralized control approach

⋆ Parts of this research have been supported by the German Research Foundation (DFG) within the Research Training Group 1362 "Cooperative, adaptive, and responsive monitoring in mixed mode environments".

is derived. In comparison to results from Parker (2002) (Section 4) it shows its potential to outperform existing heuristic approaches to the CMOMMT problem.

2. CMOMMT PROBLEM FORMULATION

2.1 Global View

A fixed number of n_R robots and n_T targets is considered as vehicles in a bounded work area $\mathscr{S} \subset \mathbb{R}^2$ with no obstacles. Each robot has a 360° *sensing range* of radius R_2, in which it is able to detect targets. It is said to *observe* a target if the target is located within the robot's *observation range* with radius R_1, where $R_1 \leq R_2$ (cf. Fig. 1). The overall region covered by the robots' observation sensors is significantly smaller than the considered work area, forcing them to dynamically adjust their movement to nearby targets. It is assumed that the robots' maximum velocity is greater than those of the targets.

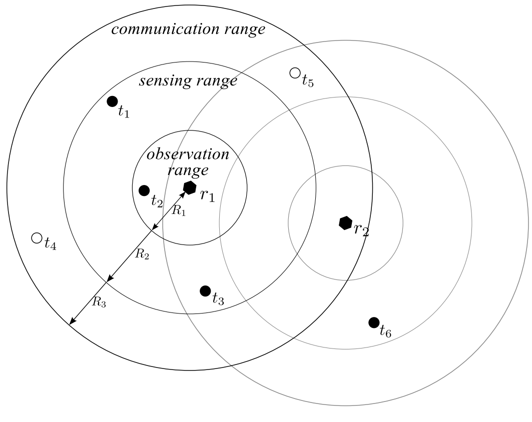

Fig. 1. Robot r_1 with its observation, sensing, and communication range. r_1 observes target t_2 and is able to detect targets t_1 and t_3. The location of t_6 is passed to r_1 by r_2. The targets t_4 and t_5 are invisible for r_1.

The common objective is to minimize the total time in which targets escape observation. In order to formalize this goal, the *A-metric* (Parker (2002)) is introduced:

$$A = \frac{1}{n_T} \sum_{i=1}^{T} \sum_{j=1}^{n_T} \frac{a_j^{(t_i)}}{T} \ , \qquad \text{(A-metric)}$$

where $a_j^{(t_i)}$ are binary variables that equal 1 if target j is observed at time step t_i and 0 otherwise. The *A-metric* represents the average percentage of targets being observed by at least one robot at some instant in time throughout a period of T time steps. Thus, the shared goal can be restated as the maximization of A, where $A = 1$ represents the absolute maximum.

The robots are able to communicate information on targets within their sensing range as well as their own position to other team mates. The *range of communication* R_3 is assumed to be significantly larger than the observation range, but too small to cover the whole work area. Positions are communicated based on a shared global coordinate system.

2.2 Local View

From a single robot's point of view, only an excerpt of the overall CMOMMT problem is considered. This subproblem involves the robot itself and all team mates within its communication range. Positions of all targets within the robot's sensing range plus those targets sensed by the robots it is able to communicate with (cf. Fig. 1) are available for locally planning a cooperative strategy.

The basic idea of the decentralized approach presented in this paper is to use a model-predictive controller to provide the robot's optimal next move based on the locally available information on other vehicles. For this purpose, a model of the local subsystem which includes the currently involved vehicles is set up and serves as a basis for predicting the evolution of the system state. Determination of a single robot's control will be based on the assumption that the movement of all other robots is optimal in the regarded subsystem. This means, optimal behavior for all involved robots is computed, from which only the control input for the currently considered robot is actually applied. Disturbances resulting from the fact that the actual system state differs from its locally predicted evolution are compensated by the model-predictive control strategy.

3. PREDICTIVE CONTROL BASED ON A MIXED LOGICAL DYNAMICAL MODEL

3.1 The MLD Framework

The *Mixed Logical Dynamical (MLD) Framework* was proposed for modeling and controlling constrained linear systems containing interacting physical laws and logical rules (Bemporad and Morari (1999)). This comprises not only hybrid systems, but also finite state machines, (constrained) linear systems or nonlinear dynamic systems where nonlinearity can be expressed through combinatorial logic. Hence, the MLD framework is a powerful tool for a wide range of modeling and control tasks, especially when combined with a predictive feedback control strategy.

In order to develop a MLD representation of the considered system, logical statements have to be transformed into linear inequalities (e.g. by using the Big-M method from Williams and Brailsford (1996)). The following system of linear dynamic equations subject to mixed-integer linear inequalities is obtained:

$$x^{(k+1)} = Ax^{(k)} + B_1 u^{(k)} + B_2 \delta^{(k)} + B_3 z^{(k)} \qquad (1a)$$

$$y^{(k)} = Cx^{(k)} + D_1 u^{(k)} + D_2 \delta^{(k)} + D_3 z^{(k)} \qquad (1b)$$

$$E_2 \delta^{(k)} + E_3 z^{(k)} \leq E_1 u^{(k)} + E_4 x^{(k)} + E_5 \ , \qquad (1c)$$

where $k \in \mathbb{Z}$ represents the current time step τ_k and
$x = [x_c \ x_b]^T, x_c \in \mathbb{R}^{n_c}, x_b \in \{0,1\}^{n_b}$ describes the system state,
$y = [y_c \ y_b]^T, y_c \in \mathbb{R}^{p_c}, y_b \in \{0,1\}^{p_b}$ is the output vector,
$u = [u_c \ u_b]^T, u_c \in \mathbb{R}^{m_c}, u_b \in \{0,1\}^{m_b}$ is the control input,
and $\delta \in \{0,1\}^{r_b}$ and $z \in \mathbb{R}^{r_c}$ represent auxiliary binary and continuous vectors, respectively.

For the proposed model-predictive control of CMOMMT subsystems it suffices that eq. (1) involves discrete-time dynamics and time-invariant matrices. Note, that the MLD framework also covers the time-variant case and can be restated with continuous time dynamics. Since the framework directly allows for the interaction of continuous physical characteristics and discrete decision logic, the MLD framework is an appropriate choice for modeling cooperative mobility problems like the CMOMMT.

3.2 MLD Model of the CMOMMT Subsystem

This section presents an approach to modeling the CMOMMT scenario using mixed-integer linear (in-)equalities and formulating a specific linear cost function. The model includes $\widetilde{n_R}$ robots and $\widetilde{n_T}$ targets.

Motion dynamics The main focus of investigation is control of cooperative behaviour. Therefore, simplifying double-integrator vehicle dynamics are used and robots as well as targets are considered point masses moving in the plane. However, the modular structure of the overall system model permits to substitute a simple dynamic model with a more complex model, if necessary.

Let x_r and y_r denote a robot's position coordinates, $v_{r,x}$ and $v_{r,y}$ the corresponding velocities, $u_{r,x}$ and $u_{r,y}$ the accelerations. The robot's locomotion at time τ then is described by $\ddot{x}_r(\tau) = u_{r,x}(\tau)$ and $\ddot{y}_r(\tau) = u_{r,y}(\tau)$. In case of piecewise constant inputs on a fixed time grid the formulation is equal to

$$\mathbf{x}_r^{(k+1)} = \begin{pmatrix} 1 & 0 & \Delta_\tau & 0 \\ 0 & 1 & 0 & \Delta_\tau \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \mathbf{x}_r^{(k)} + \begin{pmatrix} \frac{1}{2}\Delta_\tau^2 & 0 \\ 0 & \frac{1}{2}\Delta_\tau^2 \\ \Delta_\tau & 0 \\ 0 & \Delta_\tau \end{pmatrix} \mathbf{u}_r^{(k)} , \tag{2}$$

where $\mathbf{x}_r^{(k)} = (x_r^{(k)}, y_r^{(k)}, v_{r,x}^{(k)}, v_{r,y}^{(k)})$, $\mathbf{u}_r^{(k)} = (u_{r,x}^{(k)}, u_{r,y}^{(k)})$, $r = 1, \ldots, \widetilde{n_R}$, $k = 1, \ldots, N$, $\Delta_\tau = \tau_{k+1} - \tau_k$, $x^{(k)} := x(\tau_k)$, and $y^{(k)}$, $v_{x,y}^{(k)}$, $u_{x,y}^{(k)}$ defined analogously.

Simple target dynamics $\dot{x}_t(\tau) = v_{t,x}(\tau)$, $\dot{y}_t(\tau) = v_{t,y}(\tau)$ are included in the model in order to predict target positions and adjust the robot's control accordingly. The *target movement prediction* is based on an estimation supposing each target to continue linearly in its current direction. In fact, it might change its orientation during the N predicted time steps. In order to cope with this uncertainty, the predicted target velocity is reduced by a constant factor $\rho \in (0,1)$ in every time step:

$$\mathbf{x}_t^{(k+1)} = \begin{pmatrix} 1 & 0 & \Delta_\tau & 0 \\ 0 & 1 & 0 & \Delta_\tau \\ 0 & 0 & \rho & 0 \\ 0 & 0 & 0 & \rho \end{pmatrix} \mathbf{x}_t^{(k)} , \tag{3}$$

where $\mathbf{x}_t^{(k)} = (x_t^{(k)}, y_t^{(k)}, v_{t,x}^{(k)}, v_{t,y}^{(k)})$, $t = 1, \ldots, \widetilde{n_T}$. As the target positions and velocities are updated at every call of the controller the deviation from the real target behaviour is kept at a minimum.

Constraints on vehicle position, velocity, and acceleration For the example scenarios investigated in this paper, the work area $\mathscr{S}$ is considered to be a circle with radius R_{work}. Hence, the model contains the following constraints on the vehicles' positions, which linearly approximate the circular work area by a polygon with n_s edges:

$$x_r^{(k)} \sin \frac{2\pi j}{n_s} + y_r^{(k)} \cos \frac{2\pi j}{n_s} \leq R_{work} , \tag{4}$$

$$x_t^{(k)} \sin \frac{2\pi j}{n_s} + y_t^{(k)} \cos \frac{2\pi j}{n_s} \leq R_{work} , \tag{5}$$

where $j = 1, \ldots, n_s$. In addition, position, velocity, and acceleration variables are restricted by lower and upper bounds:

$$\mathbf{x}_{min} \leq \mathbf{x}_r^{(k)}, \mathbf{x}_t^{(k)} \leq \mathbf{x}_{max} \quad \text{and} \tag{6}$$

$$\mathbf{u}_{min} \leq \mathbf{u}_r^{(k)} \leq \mathbf{u}_{min} . \tag{7}$$

Distances The exact Euclidean distance between a robot r and a target t is $\hat{d}_{rt} = \sqrt{(x_r - x_t)^2 + (y_r - y_t)^2}$. A linear approximation is obtained by introducing a set of inequalities

$$(x_r^{(k)} - x_t^{(k)}) \sin \frac{2\pi j}{n_d} + (y_r^{(k)} - y_t^{(k)}) \cos \frac{2\pi j}{n_d} \ \leq \ d_{rt}^{(k)} \tag{8}$$

for $j = 1, \ldots, n_d$. If $d_{rt}^{(k)}$ is set to a minimum value, such that all of the inequalities (8) hold, then $d_{rt}^{(k)} \approx \hat{d}_{rt}^{(k)}$, whereby the accuracy of the approximation can be scaled by the constant parameter $n_d \in \mathbb{N}$. The overall optimization will ensure that $d_{rt}^{(k)}$ is driven to its smallest possible value.

Observation constraints For each robot target pair (r,t), a binary variable $b_{rt}^{(k)} \in \{0,1\}$ indicates whether or not robot r observes target t at time τ_k:

$$b_{rt}^{(k)} = 1 \quad \Rightarrow \quad d_{rt}^{(k)} \leq R_1 . \tag{9}$$

All robots are assumed to have observation ranges of equal size. Applying the Big-M method to (9) results in

$$d_{rt}^{(k)} - R_1 \quad \leq \quad M(1 - b_{rt}^{(k)}) , \tag{10}$$

where $M \geq \max_{x_r, y_r, x_t, y_t} \{d_{rt} - R_1\}$.

Since it is not of interest *which* robot observes target t, but *whether* it is observed by *any* of them, another binary variable $s_t^{(k)} \in \{0,1\}$ is introduced and represents the general observation status of target t:

$$s_t^{(k)} = 0 \quad \Leftrightarrow \quad \sum_{r=1}^{\widetilde{n_R}} b_{rt}^{(k)} \geq 1 . \tag{11}$$

A linear formulation of eq. (11) is given by the inequalities

$$1 - \sum_{r=1}^{\widetilde{n_R}} b_{rt}^{(k)} \leq M \cdot s_t^{(k)} \quad \text{and} \tag{12}$$

$$1 - \sum_{r=1}^{\widetilde{n_R}} b_{rt}^{(k)} \geq \varepsilon + (m - \varepsilon)(1 - s_t^{(k)}) ,$$

where $M \geq \max\{1 - \sum b_{rt}\} = 1$, $m \leq \min\{1 - \sum b_{rt}\} = 1 - \widetilde{n_R}$ and ε is a small tolerance close to machine precision.

The aspect of cooperation is realized by minimizing the number of unobserved targets, i.e. $\sum_{t=1}^{\widetilde{n_T}} s_t^{(k)}$, and by minimizing each robot's distance to those targets not yet observed by any other robot. The latter decision is expressed using the binary variables s_t and an additional set of auxiliary variables $h_{rt} \in \mathbb{R}$, which equal the distances d_{rt} in the case of unobserved targets and equal zero in the case of already observed targets:

$$h_{rt}^{(k)} = s_t^{(k)} \cdot d_{rt}^{(k)} . \tag{13}$$

The linear representation comprises the inequalities

$$h_{rt}^{(k)} \leq M \cdot s_t^{(k)} , \tag{14}$$

$$h_{rt}^{(k)} \leq d_{rt}^{(k)} , \quad \text{and}$$

$$-h_{rt}^{(k)} \leq -d_{rt}^{(k)} + M(1 - s_t^{(k)}) ,$$

where $M = \max_{x_r, y_r, x_t, y_t} \{d_{rt}\}$.

Objectives Clearly, the main objective in an observation scenario is to observe as many targets as possible for as many time steps as possible. In order to achieve this goal, the robots are to move towards strategically good positions. This basically means that a robot is supposed to approach unobserved targets without losing sight of the target(s) it might already be observing. In addition, the robots are to move at a minimum control effort, which in reality could correspond to energy consumption or other limiting factors. In summary, the cost function contains the following elements:

-The number of unobserved targets: $\min \sum_{k=0}^{N}\sum_{t=1}^{\widetilde{n_T}} s_t^{(k)}$ (15)

-The distances to unobserved targets: $\min \sum_{k=0}^{N}\sum_{r=1}^{\widetilde{n_R}}\sum_{t=1}^{\widetilde{n_T}} h_{rt}^{(k)}$ (16)

-The required control effort: $\min \sum_{k=0}^{N-1}\sum_{r=1}^{\widetilde{n_R}} |u_{r,x}^{(k)}|+|u_{r,y}^{(k)}|$ (17)

The elements (15) - (17) have to be weighted according to the different objective priorities and the best expected task performance. For this purpose, the weights q_δ, q_z, $q_u \in \mathbb{R}$ are introduced and the assembled cost function is of the form

$$\min_{\mathbf{u_r}} \sum_{k=0}^{N}\Big(q_\delta \sum_{t=1}^{\widetilde{n_T}} s_t^{(k)} + q_z \sum_{t=1}^{\widetilde{n_T}}\sum_{r=1}^{\widetilde{n_R}} h_{rt}^{(k)}\Big) + q_u \sum_{k=0}^{N-1}\sum_{r=1}^{\widetilde{n_R}} |u_{r,x}^{(k)}|+|u_{r,y}^{(k)}|. \tag{18}$$

3.3 MLD-Based Model-Predictive Control

The basic idea of *model-predictive control* is to use sequences of optimal control inputs $u^{(0)},\ldots,u^{(N-1)}$ computed by model-based prediction over a finite time horizon N according to some optimality criterion. The first element of the control sequence is applied to the system, then its new state is measured for computing an updated control input sequence. In this manner, the prediction horizon N is shifted over time. The main advantage of this strategy is its ability to compensate modeling inaccuracies and disturbances.

In order to apply the concept of MPC, problem (2)-(18) is reformulated as *Constrained Finite Time Optimal Control (CFTOC)* problem employing the MLD framework:

$$\min_{U_N} |Px^{(N)}| + \sum_{k=0}^{N-1} |Q_1 u^{(k)}| + |Q_2 \delta^{(k)}| + |Q_3 z^{(k)}| + |Q_4 x^{(k)}| \tag{19a}$$

$$\text{s.t.}\quad x^{(k+1)} = Ax^{(k)} + B_1 u^{(k)} + B_2 \delta^{(k)} + B_3 z^{(k)} \tag{19b}$$

$$y^{(k)} = Cx^{(k)} + D_1 u^{(k)} + D_2 \delta^{(k)} + D_3 z^{(k)} \tag{19c}$$

$$E_2 \delta^{(k)} + E_3 z^{(k)} \leq E_1 u^{(k)} + E_4 x^{(k)} + E_5\ , \tag{19d}$$

where $U_N := \{u^{(k)}\}_{k=0}^{N-1}$ is the optimization variable.

In this representation, the vector $x^{(k)} \in \mathbb{R}^{4\widetilde{n_R}+4\widetilde{n_T}}$ contains positions and velocities of all involved vehicles. All binary variables are contained in $\delta^{(k)} \in \{0,1\}^{\widetilde{n_R}\cdot\widetilde{n_T}+\widetilde{n_T}}$. $z^{(k)} \in \mathbb{R}^{2\cdot\widetilde{n_R}\cdot\widetilde{n_T}}$ comprises all other (auxiliary) continuous variables like the distances $d_{rt}^{(k)}$ and $h_{rt}^{(k)}$, respectively. The vector $u^{(k)} \in \mathbb{R}^{2\widetilde{n_R}}$ represents the robot control inputs. Eq. (19b) - (19d) comprise $4\widetilde{n_R}+4\widetilde{n_T}$ equalities and $4\widetilde{n_R}\widetilde{n_T}+2\widetilde{n_T}+\widetilde{n_R}\widetilde{n_T}n_d+(\widetilde{n_R}+\widetilde{n_T})n_s$ inequalities.

The MLD framework was exploited as favorable as possible in terms of problem complexity also taking into account how minimizing the cost function affects the variables. As an example, consider the first observation constraint in section 3.2. A correct representation would require an "*if-and-only-if*"-relation instead of the single implication (9). However, the optimization will always drive the variables b_{rt} to value 1 instead of 0, which is why eq. (9) suffices to completely describe the relation between b_{rt} and d_{rt} for problem (19).

Problem (19) is a mixed-integer *linear* formulation. Therefore a numerically robust, efficient computation can be performed, which guarantees global optimality without strongly depending on initial guesses or bounds, as it would be the case in mixed-integer non-linear programming.

4. DECENTRALIZED CONTROL

In order to use the MLD-based MPC as described in section 3 for the decentralized CMOMMT control approach, a priori the application, several MLD models are set up offline, one for each possible combination of the number of robots and the number of targets. During the online application, an appropriate model and the corresponding controller is selected by the robot based on the number of teammates and targets within its sensing / communication range as described in section 2.2. That way, each robot is controlled individually based on the current constitution of its local environment. Targets within robot r's observation range that are closer to other robot team members, will be ignored by r. This reduces the risk of multiple robots observing the same target.

In order to obtain an efficient online control strategy, a maximum number of robots $\widetilde{n}_{Rmax}$ and targets $\widetilde{n}_{Tmax}$ in a model may not be exceeded. During application, if there is information on more targets or robots available than the defined maximum, only those vehicles closest to the currently considered robot are included in the subsystem model. As complexity and calculation time grows exponentially with the model size, it is suggested to sound out reasonable calculation times by solving MILPs for different numbers of robots $\widetilde{n_R}$ and targets $\widetilde{n_T}$ as well as different lengths of prediction horizons N. As an example, Fig. 2 shows the computing times needed to solve a single MILP describing a system of $\widetilde{n_R} = 1,\ldots,5$ robots and $2\cdot\widetilde{n_R}$ targets over a prediction horizon $N=5$ and $N=3$, respectively. The boxplots were obtained from 200 solver calls for randomly generated system states for each instantiation of $\widetilde{n_R}$, $\widetilde{n_T}$, and N (performed on a Dual Core CPU, 2.53 GHz, 4GB RAM using the Multi-Parametric Toolbox (Kvasnica (2008)) for Matlab® and the solver CPLEX (ILOG (2007))).

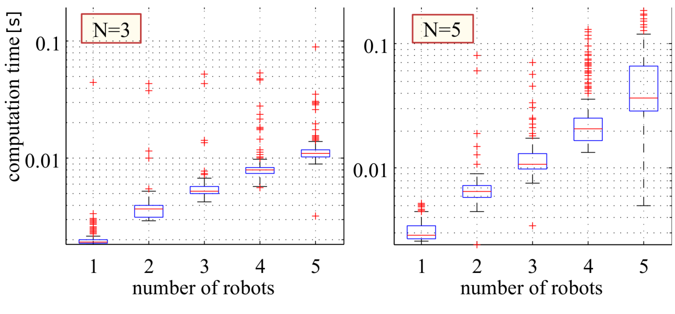

Fig. 2. Boxplots of the computing times needed to solve a single mixed-integer linear program over prediction horizon $N=3$ and $N=5$, respectively, for systems of $\widetilde{n_R}$ robots and $2\widetilde{n_R}$ targets.

Comparing the computing times permits to define an upper bound $(\widetilde{n}_{Rmax},\widetilde{n}_{Tmax},N)$ for the model size in order to maintain online efficiency of the control computation performed at each time step. However, it has to be taken into account that appropriate combinations of $\widetilde{n}_{Rmax}$, $\widetilde{n}_{Tmax}$, and N also depend on the quality of the obtained solutions.

For reasons of stability, there is no controller for subsystems without targets. If no target information is available, the robot is supposed to perform some kind of target search instead of just staying idle at a certain location. The strategy implemented in this paper causes the robot to follow the teammate currently observing the most targets (at least 2) at the same time. If one of the targets escapes the teammate's observation, the following robot can take over and switches back to the regular controlled movement as soon as it again senses a target. This strategy requires a teammate observing two or more targets within the current robot's communication range. If that is not the case, the robot switches to random movement until it either finds a target to observe or a robot to follow. Alternative approaches include, among others, systematic search algorithms based on Voronoi decomposition of the work area or probability-based patrol behavior, but will not be further explored in this paper.

Since the decentralized control approach, in general, does not reach the global optimality achieved by the centralized one, the approach is now enhanced by inter-robot communication and several rules regarding robot/target selection as was mentioned above. This is in order to make the most of every controller call and come as close as possible to the globally optimal solution.
One of the main advantages of the decentralized strategy is that the size of the subproblem can be adapted to the given system characteristics, e.g. the overall number of involved vehicles, and available computing capacities. Moreover, the decentralized approach is flexible in terms of the number of robots and targets involved in the entire system. The control strategy does not depend on a central component and a permanent, stable communication with it.

4.1 Stability

For each robot, which is controlled by the proposed model-based predictive strategy, there exists a set of equilibrium points. Namely, these are all robot positions (with velocity 0 respectively) that cannot be improved within the considered time horizon N.
In case of only one target in the robot's environment, that would correspond to positions in observation distance to the target. Therefore, for any pertubation that effects the robot's state to be outside the set of stationary points, sufficiently high weights q_δ and q_z in eq. (18) lead to a minimization of the distance between the current robot state vector and the set of equlibrium points. This distance is expressed by s_t and h_{rt}.
In case of multiple considered robots and targets within the sensing and communication range, the set of equilibrium points is not connected any more and a pertubation may direct the robot to be controlled towards a stationary point that lies in a different subset.

4.2 Comparison to a heuristic solution (Parker (2002))

In this section, solutions of the CMOMMT problem obtained with the proposed decentralized approach are presented and by simulations compared to the heuristic approach in Parker (2002). For this purpose, some parameters regarding the robots' sensing capabilities are adapted to satisfy equal conditions.

At the beginning of each simulation run, robots and targets are randomly positioned within a 1000×1000 square in the work area center. Since the robots' observation range is set to $R_1 = 2600$ and the sensing range $R_2 = 3000$, all targets are observed when the simulation starts. A robot is assumed to know about two types of targets, those it is currently observing itself and those which are located within its sensing range *and* are observed by some other robot. This further restricts the problem definition in section 2, where each robot itself was able to detect targets within its sensing range. Communication with teammates is possible within a communication range of radius $R_3 = 5000$. At the beginning, the targets are assigned a random orientation and a random velocity up to 150 units per second, which they keep constant during the run. At a 5% chance, they randomly change their orientation (max. $\pm 90°$) at each time step. If a target gets close to the work area boundary, it is repelled and moves on along the reflected direction. Robots can move with a velocity up to 200 units per second.

Settings with a fixed robot target ratio of $\frac{n_R}{n_T} = \frac{1}{4}$ and $n_R = 1,\ldots,5$ are considered. They represent the class of "harder" problems since there are more targets than robots. Hence, the robots need to observe more than one target at the same time in order to maximize the A-metric. The work area radius R_{work} varies between 1000 and 50000 units. For each instance (n_R, n_T, R_{work}), 250 simulation runs were performed and evaluated based on the average value of the A-metric. Fig. 3 shows the results obtained with the proposed decentralized MPC approach ($\widetilde{n_{R}}_{max} = \widetilde{n_{T}}_{max} = 3, N = 5$) in comparison to the results obtained with Parker's A-CMOMMT approach.

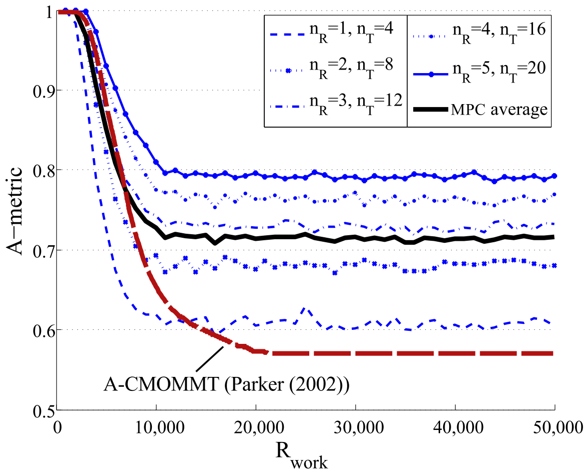

Fig. 3. Comparison of the simulation results obtained with the decentralized approach and with Parker's A-CMOMMT approach (Parker (2002)) based on the average A-metric for settings with $n_R = 1,\ldots,5$ robots and $n_T = 4 \cdot n_R$ targets in a work area with radius $R_{work} = 1000,\ldots,50000$.

Due to the small work area and frequently reflecting targets, an average value of $A = 1$ is obtained for very small values of R_{work}. Along with a rising work area size comes the increasing risk of targets escaping observation. A more or less constant average value of A is reached as soon as the work area radius does not influence the success of the observation task anymore, which is the case for $R_{work} \approx 11000$ for the MPC approach and $R_{work} \approx 22000$ for the A-CMOMMT approach. Hence, the success rate of Parker's approach keeps shrinking while that of the MPC approach has already settled at a level around $A = 0.715$. For $R_{work} > 22000$ the MPC method outperforms the A-CMOMMT method by approx. 25%.

Fig. 4 (a) shows an example of a simulation run with 4 robots and 16 targets. It can be seen that the robots cover 2 or more targets with their observation ranges as long as the targets are close enough together.

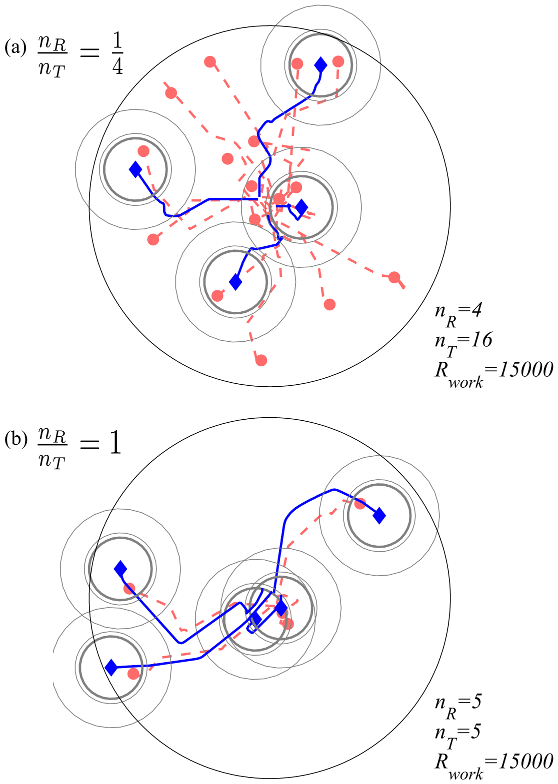

Fig. 4. Examples of simulation runs. Robots are represented by diamond shaped markers surrounded by circles indicating their observation, sensing, and communication range, respectively. The solid lines represent the robots' trajectories, the dashed lines and circular markers the targets.

It turned out, that the superiority of the MPC approach to the A-CMOMMT approach becomes even more obvious for a robot target ratio of $\frac{n_R}{n_T} = 1$. In a simulation setup similar to the $1/4$-case, average values for A around 0.98 were obtained for $R_{work} > 10000$. An improvement of approx. 38% compared to Parker's method was achieved for $R_{work} > 22000$. Fig. 4 (b) depicts a simulation run for 5 robots and 5 targets, which shows how the robots are controlled to either follow one target each or a robot which observes more than 2 targets (work area center).

5. CONCLUSION

A decentralized, model-predictive control strategy that considers a tight coupling of discrete decisions and continuous vehicle dynamics was proposed and applied to a benchmark problem of cooperative multi-robot observation of multiple moving targets. It simultaneously provides stable situation-based allocation of targets and vehicle-specific path-planning while meeting real-time requirements.

In contrast to existing heuristic approaches, the proposed method guarantees a certain degree of optimality, that is scalable by the number of considered robots, targets and the length of the MPC prediction horizon. In comparison to an approach presented by Parker (2002), the model-predictive strategy results in significantly better cooperative strategies.

Thus, for a key problem in many security, surveillance and service applications an online control strategy based on mixed-integer linear programming is presented, which is generalizable to various cooperative multi-vehicle scenarios that require the consideration of a tight coupling of discrete decisions and continuous vehicle dynamics.

REFERENCES

Bemporad, A. and Morari, M. (1999). Control of systems integrating logic, dynamics, and constraints. *Automatica*, 35, 407–427.

Ding, Y., Zhu, M., He, Y., and Jiang:, J. (2006). P-CMOMMT algorithm for the cooperative multi-robot observation of multiple moving targets. In *The Sixth World Congress on Intelligent Control and Automation (WCICA), 2006*, volume 2.

Earl, M.G. and D'Andrea, R. (2007). Multi-vehicle cooperative control using mixed integer linear programming. In J.S. Shamma (ed.), *Cooperative Control of Distributed Multi-Agent Systems*, 231–259. John Wiley & Sons, Ltd.

Fierro, R. and Wesselowski, K. (2004). *Cooperative Control*, chapter Optimization-Based Control of Multi-Vehicle Systems, 461–463. Springer Berlin / Heidelberg.

Glocker, M., Reinl, C., and von Stryk, O. (2006). Optimal task allocation and dynamic trajectory planning for multi-vehicle systems using nonlinear hybrid optimal control. In *Proc. 1st IFAC-Symposium on Multivehicle Systems*, 38–43. Salvador, Brazil.

ILOG (2007). *CPLEX 11.0 User's Manual*.

Kvasnica, M. (2008). *Efficient Software Tools for Control and Analysis of Hybrid Systems*. Ph.D. thesis, Swiss Federal Institue of Technology (ETH) Zurich.

Luke, S., Sullivan, K., Panait, L., and Balan, G. (2005). Tunably decentralized algorithms for cooperative target observation. In *AAMAS '05: Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, 911–917. ACM, New York, NY, USA.

Markov, S. and Carpin, S. (2007). A cooperative distributed approach to target motion control in multirobot observation of multiple targets. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2007*.

Morari, M. and Baric, M. (2006). Recent developments in the control of constrained hybrid systems. *Computers & Chemical Engineering*, 30(10–12), 1619–1631.

Parker, L.E. (1998). Alliance: An architecture for fault tolerant multi-robot cooperation. *IEEE Transactions on Robotics and Automation*, 14 (2).

Parker, L.E. (2002). Distributed algorithms for multi-robot observation of multiple moving targets. *Autonomous Robots*, 12, 231–255.

Parker, L.E. and Emmons, B.A. (1997). Cooperative multi-robot observation of multiple moving targets. In *In Proceedings of the 1997 IEEE International Conference on Robotics and Automation*, 2082–2089.

Reinl, C. and von Stryk, O. (2007). Optimal control of cooperative multi-robot systems using mixed-integer linear programming. In *Proc. RoboMat 2007 - Workshop on Robotics and Mathematics*, 145–151. Coimbra, Portugal.

Richards, A. and How, J. (2004). Decentralized model predictive control of cooperating UAVs. volume 4, 4286 – 4291.

Risler, M. (2009). *Behavior Control for Single and Multiple Autonomous Agents Based on Hierarchical Finite State Machines*. Fortschritt-Berichte VDI, Technische Universität Darmstadt.

Williams, H.P. and Brailsford, S.C. (1996). Advances in linear and integer programming. 249–281. Oxford University Press, Inc., New York, NY, USA.