Preprint of a paper which appeared in the proceedings of
“RoboMat 2007 - Workshop on Robotics and Mathematics”
(Coimbra, Portugal, 17-19 September 2007)

Optimal Control of Cooperative Multi-Robot Systems using Mixed-Integer Linear Programming

Christian Reinl, Oskar von Stryk *

Abstract

A new planning method for optimal control of multi-robot systems is discussed which accounts for the (continuous) physical locomotion dynamics of the robots and its tight coupling to the distribution and allocation of (discrete) subtasks to the robots to fulfill a joint mission. The point of departure is a nonlinear and nonconvex hybrid optimal control problem (HOCP) formulation which incorporates a detailed hybrid automaton model. Because of the many difficulties involved in solving this problem like large computational times and the lack of good or global convergence properties it is transcribed into a mixed-integer linear program (MILP). This can be solved much more efficiently using existing algorithms. The proposed approach is outlined for an example problem of cooperative soccer robots. The MILP solution itself may serve either as a good initial solution estimate for a method addressing the nonlinear HOCP or may later become the kernel of a model predictive control method for cooperative multi-robot systems. Despite the promising results obtained so far a variety of open questions yet remains to be answered including the ”best” way of transcribing HOCP to MILP with respect to both computational efficiency and good HOCP solution approximation.

1 Introduction

In this paper multi-robot systems are considered where the individual nonlinear physical motion dynamics is of fundamental importance for the mission success which depends on optimizing physical values like the robots’ positions or energy consumption.

The problem’s possible combinatorial character complicates an analytical inspection and hardly theoretically proved results are available for the most general problem formulation. The key idea in this paper is to build up a centralized MILP-based (cf. [**?**]) controller for multi-vehicle systems, starting with a systematic HOCP description (Sect. 2) and a consistent transformation (Sect. 3) towards optimization based control (Sect. 4). Tight coupling of discrete states (e.g. actions) an respective continuous state variables (positions, velocities etc.) is a basic feature in there. Especially in an uncertain setting (failures, uncontrollable

*Technische Universität Darmstadt, CS Dept., E-mail: {reinl, stryk}@sim.tu-darmstadt.de.

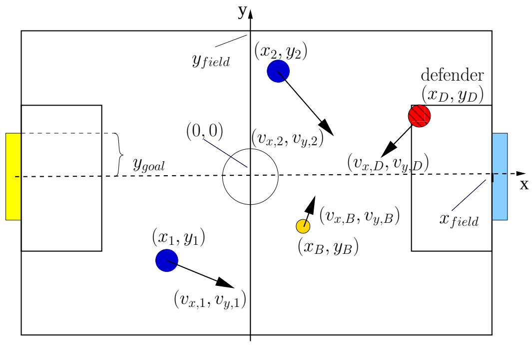

Figure 1: Setting of the soccer benchmark problem

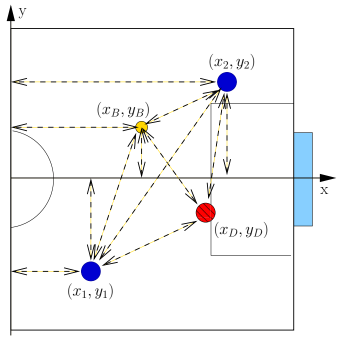

Figure 2: Contributions to objective

objects), the robustness of MILP offers an efficient (cf. [?]) way to be used in receding horizon controllers. To illustrate this approach we refer to a benchmark problem from robot soccer (cf. Fig. 1) with two strikers versus one (passive) defender, all modeled as moving point masses. The intention is to find the control which optimizes the attackers' chances for a considered time horizon $[t_0,\, t_f]$. Results for this representative example will be given in Sect. 4.

2 Modeling the cooperative multi-robot system

We are considering (in-)direct controllable and not controllable moving objects i in our system. Each one is characterized by its continuous dynamic state $\boldsymbol{x}_i$ (e.g. position, velocity,...) and a discrete value q_i that denotes a certain subtask or role. Together with the continuous control variable $\boldsymbol{u}_i$, the continuous state evolves subject to $\dot{\boldsymbol{x}}_i = \boldsymbol{f}_{q_i,i}(\boldsymbol{x}_i, \boldsymbol{u}_i)$. By defining (usually unknown) switching times t_s and corresponding specifications, how to connect $\boldsymbol{x}_i$ when q_i switches at t_s, the individual trajectory for an object is determined.

For the regarded soccer example, $i \in \{1,\, 2,\, B,\, D\}$ denotes two strikers, one ball and a defender. As modes of motion q_i for the strikers we distinguish `free_moving` and `dribbling`.

2.1 Modeling switched dynamics with hybrid automata

We are regarding multi-robot systems consisting of moving objects with specific modes of motion and rules that define feasible sequences for them. Thus we are using hybrid automata to describe the cooperative system. They are well established in the context of robot control.

A hybrid automaton [?] $H = (V, E, \mathbb{X}, \mathbb{U}, ini, \mathbf{f}, \mathbf{j}, \mathbf{i}, \mathbf{e})$ consists of a finite directed multi-graph (V, E) with knots in V (called states) and edges in E (so-called switches), a set of continuous state variables $\mathbb{X}$, a set of continuous control variables $\mathbb{U}$, a map ini which assigns an initial condition to each edge, the invariants provided by the map $\mathbf{i}$ which assigns each knot with a feasible region for the continuous states and controls using equality and inequality constraints, a map $\mathbf{f}$ which assigns a flow equation or state dynamics to each state, a map $\mathbf{j}$ which assigns jump conditions to edges and a map $\mathbf{e}$ which assigns events to edges which occur at switches. For the proposed soccer application (cf. Fig. 3) we added another hierarchy there that contains conditions that are similar in the covered knots. In this model we only

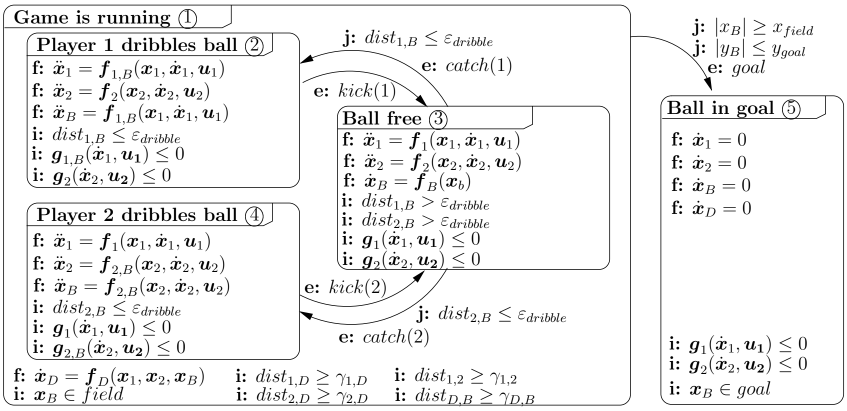

Figure 3: Hierarchical hybrid automaton model of the switched motion dynamics

distinguish whether the ball is dribbled, rolls free or is inside the goal. The respective motion dynamics of a dribbling robot is indexed by "B". The initial conditions ini are defined with the position $\boldsymbol{x}_i(t_0)$ of the objects i. Catching and kicking a ball are modeled by events

$$kick(i): \ \dot{\boldsymbol{x}}_B(t_s+0) = 3 \cdot \dot{\boldsymbol{x}}_i(t_s-0)\,, \ catch(i): \ \boldsymbol{x}_B(t_s+0) = \boldsymbol{x}_i(t_s-0)\ , \tag{1}$$

$t_s \pm 0 := \lim_{\epsilon \to 0, \epsilon > 0} t_s \pm \epsilon$. All other state trajectories are required to be continuous at t_s.

The auxiliary variable $dist_{i_1,i_2}$ represents a distance measure between objects i_1, i_2 and is used to express collision avoidance (with a constant γ_{i_1,i_2}). Further constraints on state and control variables according to the specific motion modes are modelled as invariants **i**.

The dynamic of the defender is not considered to be switched here. We tested our approach with a simple model for the dynamic of robots and ball and set for all states except ⑤

$$\textbf{f: } \dot{\boldsymbol{x}}_B(t) = \boldsymbol{v}_B(t),\ \textbf{f: } \ddot{\boldsymbol{x}}_\Diamond(t) = \begin{pmatrix} \ddot{x}_\Diamond(t) \\ \ddot{y}_\Diamond(t) \end{pmatrix} = \begin{pmatrix} \dot{v}_{x,\Diamond}(t) \\ \dot{v}_{y,\Diamond}(t) \end{pmatrix} = \boldsymbol{u}_\Diamond(t) = \begin{pmatrix} u_{x,\Diamond}(t) \\ u_{y,\Diamond}(t) \end{pmatrix} \ (\Diamond \in \{1,2\}) \tag{2}$$

For a dribbling robot the upper bound on its velocity is reduced by a factor $c^U_{v,dr}$.

The numerical method proposed in [?] for the general HOCP formulation uses piecewise polynomials and binary variables to transfer the hybrid automaton and an objective function into a finite dimensional sparse non-linear mixed-binary optimization problem. It was solved with a combination of sequential quadratic programming and Branch-and-Bound techniques. More details and results for this model are given there. Now we are interested in a MILP-formulation that provides a efficient optimization-based control considering the basic system characteristics.

2.2 The linearized model

We are introducing a fixed (not necessary) equidistant grid of time points with the sampling time $\boldsymbol{T}_s := t_k - t_{k-1}$. For the evolution of the continuous state and control variables we are defining $\boldsymbol{x}(k) := \boldsymbol{x}(t_{k+1})$ and $\boldsymbol{u}(k) := \boldsymbol{u}(t_{k+1})$. The knots in the describing automaton can switch only at these time points and are not free any more. All (in-)equalities that were used to describe the different states and the motion dynamics in the systems will be reformulated or transformed into linear expressions now. Thus the differential flow conditions **f** must be reformulated as difference equations

$$\dot{\boldsymbol{x}} = \boldsymbol{f}_{q,i}(\boldsymbol{x}, \boldsymbol{u}) \quad \rightsquigarrow \quad \boldsymbol{x}(k+1) = A_{q,i} \cdot \boldsymbol{x}(k) + B_{q,i} \cdot \boldsymbol{u}(k) \ . \tag{3}$$

Additional state variables and binary variables may be necessary here for case differentiations and combination of these cases with logical expressions. In the context of hybrid automata, this case differentiations are treated as new subknots. This splitting up strongly depends on the nonlinearity of the expression and the desired accuracy of the transformed model.The regions defined by the invariants **i** are approximated in a polygonal manner. Linear inequalities are combined logically therefore. If nonlinear expressions occur in the jump conditions **j** of events **e**, they have to be treated respectively. Afterwards, the automaton is clocked and covers only linear expressions and logical constraints. Application to the example results in

$$\begin{aligned} x_{\Diamond}(k+1) &= x_{\Diamond}(k) + T_s\, v_{x,\Diamond}(k)\ , \quad & v_{x,\Diamond}(k+1) &= v_{x,\Diamond}(k) + T_s\, u_{x,\Diamond}(k), \\ x_B(k+1) &= x_B(k) + T_s\, v_{x,B}(k)\ , \quad & v_{x,B}(k+1) &= \begin{cases} v_{x,\Diamond}(k) & \text{(dribbling)} \\ c_{trac}\, T_s\, v_{x,B}(k) & \text{(ball free)} \end{cases} \end{aligned}$$

($\Diamond \in \{1, 2\}$, $c_{trac} \cdot T_s < 1$, y_i, $v_{y,i}$, analogously). A simple, reactive defender that is always moving towards the current ball position is modeled by

$$x_D(k+1) \;=\; x_D(k) + Ts\, v_{x,D}(k)\ , \quad v_{x,D} \;=\; \tfrac{v^U_{x,D}}{D_{max}}(x_b(k) - x_D(k)), \tag{4}$$

($D_{max} \geq \max_{x,y,k}\{|x_b(k) - x_D(k)|, |y_b(k) - y_D(k)|\}$, y_D, $v_{y,D}$, analogously). The constant $v^U_{x,D}$ is the upper bound for $|v_D|$. In the investigated example the controls and velocities are constraint by quadratic expressions. Generally expressions of the form $\pm\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \leq \pm r$ can be transformed by using $n_\gamma \geq 4$ linear expressions

$$\pm \sin(\frac{i}{3}\pi)\,(x_1 - x_2) \pm \cos(\frac{i}{3}\pi)\,(y_1 - y_2) \leq \pm r \ \ (i = 1, \ldots, n_\gamma) \ . \tag{5}$$

Thus the invariants **i:** $\boldsymbol{g}_{q_i,i}(\dot{\boldsymbol{x}}_i) = ||(v_{x,i},\, v_{y,i})^T||_2 - v^U_{q_i,i} \leq 0$ were reformulated ($v^U_{q_i,i}$ constant, for $\boldsymbol{u}_{q_i,i}$ respectively). For the distance $dist_{i_1,i_2}$ between two objects as an auxiliary state variable the column-sum norm was used.

3 Transforming the model into a mixed-integer linear program

For each knot and each edge of the (hierarchical) automaton a time-dependent binary variable $b(t)$ is introduced so that $b = 1$ iff the state or edge is active. The structure then is transcribed with simple linear inequalities, e.g. $b_{(2)}(k) + b_{(4)}(k) + b_{(3)}(k) \leq 1$, $b_{(2)}(k) + b_{(4)}(k+1) \leq 1$,etc. Logical relations combined with inequalities are translated using the 'Big-M'-technique (cf. [**?**]). Thus flows and invariants get connected with the respective binary variable, e.g.

$$\texttt{IF } b_{(3)} = 1 \texttt{ THEN } v_{x,B}(k+1) = c_{trac}\, v_{x,B}(k) \ \Leftrightarrow \ \begin{cases} (1 - b_{(3)})m \leq v_{x,B}(k+1) - c_{trac}\, v_{x,B}(k) \\ v_{x,B}(k+1) - c_{trac}\, v_{x,B}(k) \leq (1 - b_{(3)})M \end{cases}$$

$M \geq \max\{v_{x,B}(k+1) - c_{trac} v_{x,B}(k)\},\ m \leq \min\{v_{x,B}(k+1) - c_{trac} v_{x,B}(k)\}$ constant.

To rate the quality of a computed attack, we mainly look at the situation at the final time t_{N+1} and primarily regard the following components (cf. Fig. 2). The positions of the attacking robots and the ball $(x_i(k), y_i(k))^T$, distances between the robots, defender and ball $dist_{i_1,i_2}(k)$ and also the events "*ball in goal*" and "*one robot dribbling*" $b_{(5)}$, $b_{(2)}$, $b_{(4)}$. With carefully determined coefficients then the objective function J is implemented as a weighted sum. Due to remaining degrees of freedom, $\boldsymbol{u}_i(k)$ is further added to it. The intention is to minimize J where the tactical behavior of the team is varied with the coefficients in J.

4 Optimization

Results for the linear implementation of the proposed benchmark problem with the parameters

$$
\begin{array}{llllllllllllllllll}
T_s &=& 0.8, & N &=& 10, & x_{field} &=& 270, & y_{field} &=& 180, & y_{goal} &=& 40, & \varepsilon_{dr} &=& 5, \\
\gamma_{2,D} &=& 30, & \gamma_{B,D} &=& 30, & D_{max} &=& 700, & c_{trac} &=& 0.88, & v^U_{B,x} &=& 135, & c_{v,dr} &=& 60.7, \\
v^U_{\Diamond,y} &=& 45, & u^U_{\Diamond,x} &=& 40, & u^U_{\Diamond,y} &=& 40, & v^U_{\Diamond,y} &=& 90, & v^U_{\Diamond,x} &=& 45,
\end{array}
$$

and the objective function

$$
J = \sum_{k=1}^{N} \left(-0.2\, dist_{B,D} - 320\, b_{(5)} + 0.001 \sum_{i\in\{1,2\}} (\, |u_{x,i}| + |u_{y,i}|\,) \right)\Bigg|_{t=k} + \qquad (6)
$$
$$
\left(-x_B + 0.6\,|y_B| - 0.15\, dist_{B,D} - 0.01\, dist_{1,2} - 160\,(b_{(2)} + b_{(4)}) + 0.02 \sum_{i\in\{1,2\}} (-x_i + |y_i|) \right)\Bigg|_{t=N+1} .
$$

are given. The MILP was solved with CPLEX 10.0 (from ILOG, Inc.) on a PC (Intel(R) Pentium(R) M processor 1.86GHz; 1024 MB RAM) in 30 sec (see Fig. 4 for details).

5 Conclusion and outlook

A MILP formulation has been developed which accounts for the tight coupling of discrete decisions and continuous flow variables in optimal control of cooperative mobile robot systems. A consistent modeling towards a linearized formulation was shown. The numerical approach is applicable to a wide range of scenarios. Ongoing work considers techniques to improve the MILP by decoupling and additional constraints. Also various methods to linearize the given nonlinear description more systematically are investigated. MILP-models can cope well with the non-convexities, the combinatorial character and their efficiency hardly depends on initial guesses. They are therefore well suited for repeated application to account for changes in uncertain environments.

Acknowledgment. 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".

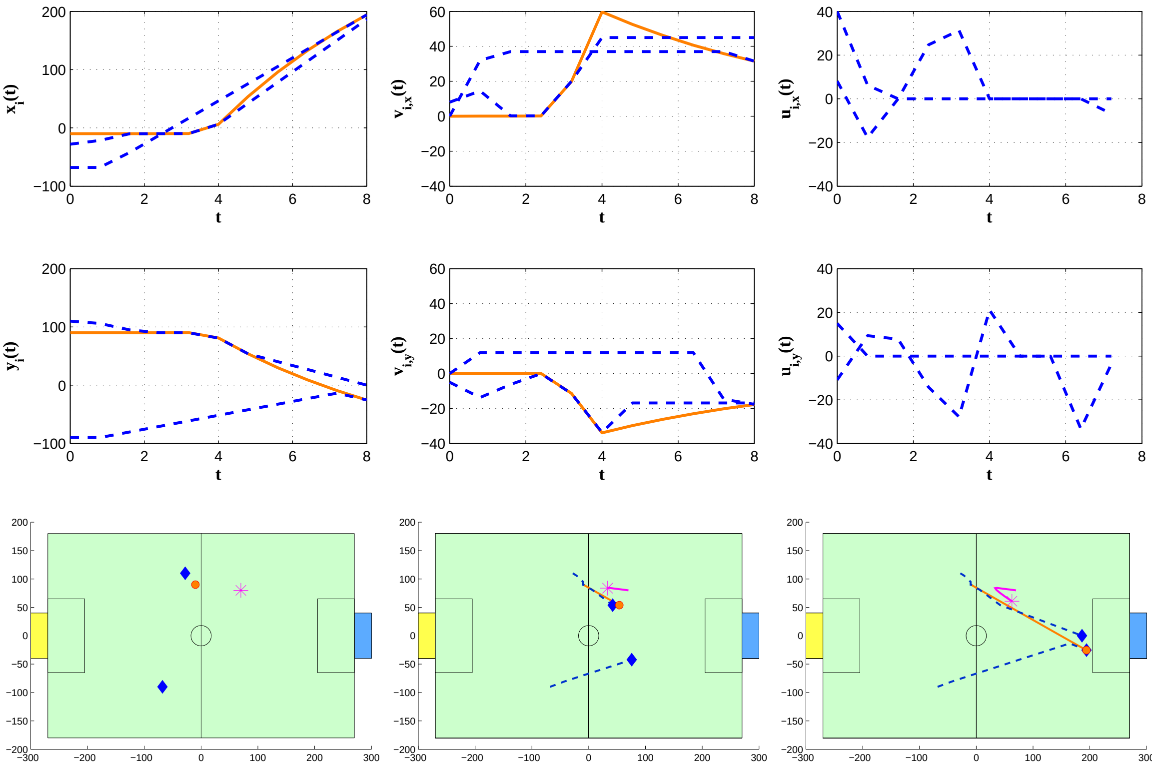

Figure 4: First two rows: Optimal positions, velocities and controls for the attackers (⋯♦) and the ball (—●). Third row: Computed optimal behavior shown at timesteps $k = 1, 7, 11$. Attacker 1 goes to ball, dribbles and kicks it towards the penalty area. Attacker 2 catches it there and dribbles to a promising position. The defender (—✱) follows the ball.

References

[1] A. Bemporad and M. Morari. Control of systems integrating logic, dynamics, and constraints. *Automatica*, 35(3):407–427, 1999.

[2] F. Borrelli, D. Subramanian, A. U. Raghunathan, and L. T. Biegler. MILP and NLP techniques for centralized trajectory planning of multiple unmanned air vehicles. *American Control Conference*, June 2006.

[3] M. G. Earl and R. D'Andrea. Iterative MILP methods for vehicle control problems. In *IEEE Conference on Decision and Control*, volume 4, pages 4369 – 4374, December 2004.

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

[5] T.A. Henzinger. The theory of hybrid automata. In M.K. Inan and R.P. Kurshan, editors, *Verification of Digital and Hybrid Systems*, NATO ASI Series F: Computer and Systems Sciences 170, pages 265–292. Springer, 2000.