

# Solution of a Multi-Agent Transport Problem by Hybrid Optimization

Herbert Mangesius\* Marion Sobotka\* Olaf Stursberg\*\*

\* *Institute of Automatic Control Engineering, Technische Universität München, Germany*

\*\* *Control and System Theory, Universität Kassel, Germany*

---

**Abstract:** As reported in earlier work, methods for hybrid optimal control can be used for optimal path planning of multi-agent systems. This paper considers a multi-vehicle transport scenario in which a hybrid model is used to represent the continuous dynamics of the vehicle motion and the dynamics arising from the docking-events between vehicles and a transported object. The decisions on the vehicles' heading and speed as well as on the docking sequence are strongly coupled and require a tailored numeric solution. The paper proposes a hierarchical solution strategy with three layers: on the upper layer a discrete event sequencing problem is solved, the middle layer determines docking positions and times, and the lower layer provides optimal vehicle paths. By a combination of graph search and embedded continuous optimal control, the efficiency of obtaining (sub-)optimal solutions is improved. The algorithm is illustrated by a scenario in which three vehicles transport one object.

*Keywords:* Multi-agent systems, discrete event systems, optimal control, hybrid dynamics.

---

## 1. INTRODUCTION

This contribution deals with the task how a group of agents moving in a plane can optimally transport a passive object to a desired location. The task involves discrete decisions with respect to where and when the vehicles dock to the transport object and continuous decisions for the vehicle motion, thus leading to a hybrid optimal control problem. Hybrid systems, as e.g. introduced in Alur et al. (1995) or Lynch et al. (2003) are widely used for modeling multi-agent systems (see e.g. Jadbabaie et al. (2003) or Fierro et al. (2001)) as they allow for continuous control and logic-based decisions in one model.

There is a number of intensively studied tasks for multi-vehicle systems. In formation control, stability of synchronization of the vehicles is discussed. Dunbar and Murray (2006) present an optimization-based receding-horizon approach. Waypoint problems are examples for cooperative tasking, see Reinl et al. (2009) for a mixed-integer optimization approach. Further, consensus problems, coverage problems, or rendezvous problems are discussed, see the overview on cooperative control for multi-vehicle systems by Murray (2007). Our application scenario is a novel cooperative tasking scenario and a solution is provided by a decoupling approach of hybrid optimal control. A large object is transported from a start to a goal position by several small agents that sequentially dock to the object in order to share workload. On the discrete-event level, it has to be decided in what order agents will take part in transportation. The continuous problem delivers state trajectories as well as the places and times of docking. It is evident that optimal agent paths can only be found when the docking order is known, but also a decision on the optimal docking order requires knowledge on the cost of the individual paths.

The presented approach starts from a known three-layer approach to hybrid optimal control, where the middle layer provides switching states and times for a lower layer which computes optimized continuous trajectories, see Shaikh and Caines (2007); Xu and Antsaklis (2004); Gokbayrak and Cassandras (2000). On the upper layer, the switching sequence is determined by enumeration of all possible sequences. The best switching sequence is then chosen by comparison of costs for the embedded continuous optimization problems. The possible number of discrete sequences in general increases exponentially with the number of discrete states or decisions to be made. Therefore, decoupling is used here, by replacing the optimal switching states and times on the middle layer by approximations. This allows to use efficient solution methods which are available for discrete systems to find the appropriate switching sequences without the necessity of enumeration. The continuous optimal control problems of the lower layer are considered as embedded problems of the graph search. In most applications decoupling of the optimal control problem delivers only sub-optimal solutions (Stursberg (2004)). For the given scenario, three different techniques for decoupling the discrete and the continuous dynamics are proposed. The first approach approximates docking sites by geometric considerations. The latter two establish two variants of optimization based approximation of the docking sites, compare to Li and Cassandras (2006).

Mixed integer approaches also showed promising performance for some multi-agent cooperative tasks, see Reinl et al. (2009); Earl and D'Andrea (2002). However, the computational efforts are still high. The hierarchical three-layer approach presented here is intended to lead to a good compromise between optimality and numerical efficiency.

Sec. 2 introduces the multi-agent setting, its modeling, and the optimal control problem. Sec. 3 provides the solution approach based on hierarchical optimization, Sec. 4 presents numerical results, and Sec. 5 concludes the paper.

## 2. TRANSPORTATION PROBLEM

In the multi-agent system (MAS) transportation problem illustrated in Fig. 1, the vehicles  $V_1, \dots, V_N$  have to cooperate in order to transport the object  $TO$  from its initial to a given target position. The object  $TO$  is passive, i.e. it can only be moved and steered by a certain number of driving vehicles being attached to it. Realizing the cooperative task with maximum performance requires to determine the best suited sequence for the vehicles to dock to  $TO$  as well as the most suitable docking sites and docking times. The initial positions of the vehicles are denoted by  $\mathbf{p}_{V_1}^0, \dots, \mathbf{p}_{V_N}^0$ . Note that all vehicles, which have not yet docked to  $TO$ , may decide to move towards a future docking position, what calls for a centralized planning method. Let  $\mathcal{D}_V(t)$  denote the index set of docked (or active) vehicle(s) moving  $TO$  at time  $t$ , and conversely, let  $\mathcal{M}_V(t)$  denote the set of indices of the vehicles not yet docked to  $TO$ .

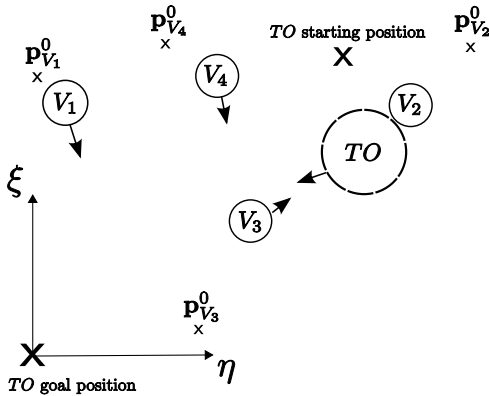


Fig. 1. Scenario: vehicles  $V_1, \dots, V_4$  cooperate to transport object  $TO$  from a starting to a goal position. The arrows show the heading of the vehicles, and  $\mathbf{p}_{V_1}^0, \dots, \mathbf{p}_{V_4}^0$  denote the vehicles' initial positions in a plane spanned by axes  $\xi$  and  $\eta$ .

### 2.1 Hybrid System Model

In a hybrid model, the transportation planning scenario with multiple vehicles is mathematically described by collections of dynamical systems that evolve on continuous-variable state spaces and are subject to continuous control and discrete transitions.

A continuously-controlled autonomously-switching hybrid dynamical system formally is given by the tuple

$$\mathbb{H} = [Q, \Sigma, \mathcal{A}, \mathcal{G}] \quad (1)$$

(see Branicky et al. (1998)) and can be used to model the MAS transportation scenario as follows:

$Q = \{q_1, \dots, q_{N_q}\}$  is the finite set of discrete states with  $q(t) \in Q$ . Within a discrete state, the MAS evolves continuously on an ordered set of time intervals

$$\Lambda = \{[t_0, t_1[, [t_1, t_2[, \dots, [t_{L-1}, t_e]\} \quad (2)$$

where  $L$  is the number of discrete states that appear in a solution. A discrete state  $q$  is assigned to every possible set of indices of docked vehicles in  $\mathcal{D}_V(t)$ , compare Fig. 2. Switching between two discrete states occurs at switching times  $t_i$  in  $\Lambda$ , at which a vehicle docks to  $TO$ . A set of logic rules is introduced to impose bounds on the number of discrete states and transitions between them: A structure of the MAS is considered, in which docking occurs sequentially and the following assumptions holds:

- undocking is not allowed once a vehicle has docked to the object  $TO$ ,
- at each docking event only one vehicle in addition to the object  $TO$  is involved,
- all vehicles have to dock once to the object  $TO$ .

Therefore, the number of discrete states in a feasible run of  $\mathbb{H}$  is  $L = N_V + 1$ . With these rules, the discrete structure of the MAS is an acyclic directed graph as depicted in Fig. 2. A possible ratio for the last assumption is that a preceding analysis has determined already the set of agents that should contribute to the transportation task.

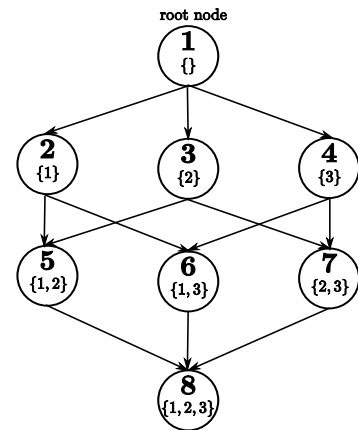


Fig. 2. Graph structure for three vehicles and sequential docking: bold numbers stand for the discrete state index  $q$ , the indices contained in  $\mathcal{D}_V$  are given in braces.

$\Sigma = \{\Sigma_q\}_{q \in Q}$  is the collection of controlled dynamical systems, where each  $\Sigma_q = [X_q, \mathbf{F}_q, U_q]$  is a continuously-controlled dynamical system. The continuous state vectors  $\mathbf{x}_q$  of dimension  $n_{x_q} \in \mathbb{N}^+$  are defined on the continuous state spaces  $X_q \subseteq \mathbb{R}^{n_{x_q}}$ . Accordingly, the continuous control inputs  $\mathbf{u}_q(t)$  of dimension  $n_{u_q} \in \mathbb{N}^+$  are defined on the continuous input spaces  $U_q \subseteq \mathbb{R}^{n_{u_q}}$ . The time-invariant vector fields  $\mathbf{F}_q : \mathbb{R}^{n_{x_q}} \times \mathbb{R}^{n_{u_q}} \rightarrow \mathbb{R}^{n_{x_q}}$  provide the continuous dynamics of  $\mathbb{H}$ .

The state spaces  $X_q := X$  of the MAS are identical for all  $q$  and are defined as the product of the configuration spaces of the vehicles and the  $TO$ :

$$X = C^{V_1} \times C^{V_2} \times \dots \times C^{V_N} \times C^{TO}. \quad (3)$$

Within this paper, a unicycle dynamics is chosen as a nonlinear model for the motion of each agent:

$$\begin{pmatrix} \dot{\eta} \\ \dot{\xi} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} u_s \cos(\theta) \\ u_s \sin(\theta) \\ u_\omega \end{pmatrix}, \quad C^V = \mathbb{R}^2 \times [-\pi; \pi] \quad (4)$$

The control input for every vehicle is  $(u_\omega, u_s)^T$ , where  $u_\omega$  directly sets the angular velocity of the orientation and  $u_s$  sets the speed.

The object  $TO$  moves passively, and hence the dynamics for  $TO$  is:

$$\begin{pmatrix} \dot{\eta} \\ \dot{\xi} \\ \dot{\theta} \end{pmatrix} = wf(b(t)) \begin{pmatrix} u_s \cos(\theta) \\ u_s \sin(\theta) \\ u_\omega \end{pmatrix}, \quad C^{TO} = \mathbb{R}^2 \times [-\pi; \pi] \quad (5)$$

The constant  $w$  is an arbitrary gain. The function  $f(b) = \tanh(\frac{2}{N_V}b)$  with  $b(t) = |\mathcal{D}_V(t)|$  is an activation gain: the more vehicles dock to the object the faster can it be moved; however, ultimately the transported object has a maximum moving speed.

The state  $\mathbf{x}_q = \mathbf{x} \in X$  specifies all agent configurations and is

$$\mathbf{x} = ((\mathbf{x}^{V_1})^T, \dots, (\mathbf{x}^{V_N})^T, (\mathbf{x}^{V_{TO}})^T)^T, \quad (6)$$

where the dimension of  $\mathbf{x}$  is  $n_x = N \dim(C^V) + \dim(C^{TO})$ . In summary, the dynamics of the MAS is:

$$\dot{\mathbf{x}} = \mathbf{F}_q(\mathbf{x}, \mathbf{u}_q), \quad (7)$$

with  $\mathbf{F}_q$  and  $\mathbf{u}_q$  such that  $\dot{\mathbf{x}}$  is given by (4) for free vehicles, and by (5) for docked vehicles and the transported object  $TO$  in accordance with the state vector in (6).

$\mathcal{A} = \{\mathcal{A}_q\}_{q \in Q}$  is the *collection of autonomous jump sets*. An autonomous jump set  $\mathcal{A}_q$  is a specified region in the state space  $X_q$  where autonomous switching occurs in case the state vector  $\mathbf{x}_q$  is hitting its boundary. In our application, it is used to describe docking that occurs when a vehicle meets the transportation object:

$$\mathcal{A}_q = \{\mathbf{x} \mid \mathbf{x}^{V_i} = \mathbf{x}^{V_{TO}}, i \in \mathcal{M}_V\}. \quad (8)$$

$\mathcal{G} = \{G_q\}_{q \in Q}$  is the *autonomous jump transition map* with hybrid transition functions  $G_q : \mathcal{A}_q \times Q \rightarrow X_q \times Q$ . Hence,  $G_q$  contains all possible discrete transitions and the update of the continuous states, when the state vector  $\mathbf{x}_q$  hits  $\mathcal{A}_q$  according to a transition  $q \rightarrow q^+$ .

The MAS evolves in the hybrid state space  $S = X \times Q$  of  $\mathbb{H}$  and  $\boldsymbol{\sigma}(t) = (\mathbf{x}(t), q(t))$  is the hybrid state vector with  $\boldsymbol{\sigma} \in S$ . Let  $\Phi_q = (q(t_0), q(t_1), q(t_2), \dots, q(t_{L-1}))$  be the discrete state sequence,  $\Phi_{\mathcal{A}} = (\mathbf{x}(t_0), \mathbf{x}(t_1), \dots, \mathbf{x}(t_{L-1}))$  the sequence of switching points,  $\Phi_{\mathbf{x}}$  the continuous state trajectory, and  $\Phi_{\mathbf{u}}$  the trajectory of the control inputs.

## 2.2 Optimal Control Problem

The objective is to move the object  $TO$  from its initial position to a final configuration in an energy and time optimal manner. The following hybrid optimal control problem is considered: Find a discrete state sequence  $\Phi_q$ , a sequence of switching points  $\Phi_{\mathcal{A}}$ , and continuous control inputs  $\Phi_{\mathbf{u}}$  such that the state trajectories  $\Phi_{\mathbf{x}}$  and  $\Phi_q$  of the autonomously-switching, continuously-controlled hybrid dynamical system  $\mathbb{H}$  satisfy the following hybrid boundary value problem:

$$\boldsymbol{\sigma}(t_0) = (\mathbf{x}_q^0, q(t_0)), \quad \boldsymbol{\sigma}(t_e) = (\mathbf{x}_q^e, q(t_e)), \quad t_e \text{ free} \quad (9)$$

such that the following real-valued cost function is minimized:

$$J = \sum_{k=0}^{L-1} \left\{ \int_{t_k}^{t_{k+1}} \mathbf{u}_{q(t_k)}^T \mathbf{u}_{q(t_k)} + \mu_k d\tau \right\}. \quad (10)$$

The first part of  $J$  formulates the running costs due to the continuous dynamics and the second term encodes time optimality weighted by  $\mu_k$ .

Finally, let the solution of the optimization problem be denoted by  $\Phi_q^{\text{opt}}, \Phi_{\mathcal{A}}^{\text{opt}}, \Phi_{\mathbf{u}}^{\text{opt}}$ . Note that a globally best solution involves the coupling of locally best solutions ( $\Phi_{\mathbf{u}}^{\text{opt}}$ ) and globally best plans ( $\Phi_q^{\text{opt}}$ ) through  $\Phi_{\mathcal{A}}$  and  $t_s$ .

## 3. SOLUTION BY A HIERARCHICAL ALGORITHM

The hybrid optimal control problem (HOCP) is solved by separating the optimization of the continuous and the discrete dynamics. A hierarchical structure is used where the middle layer accounts for the coupling of optimal discrete and optimal continuous dynamics via optimal docking sites and times.

The three-layer approach is illustrated in Fig. 3: On the upper layer, the optimal discrete state sequence is determined, on the middle layer the optimal docking positions are computed, and on the lower layer, the continuous optimal control problems are solved as two-point boundary value Problems (TPBVP). Each of the loops uses the optimal cost value of the embedded problem on the next lower layer as performance index for variations of the respective argument on its own layer. In order to adequately measure changes in the performance index due to variations on a higher layer, all optimal solutions of the embedded sub-problems on lower layers need to be recalculated. By that, an embedded sub-problem can be considered as a 'black-box' where the input (varied argument) and the output (corresponding performance index) are the only measurable quantities from the perspective of the upper layer.

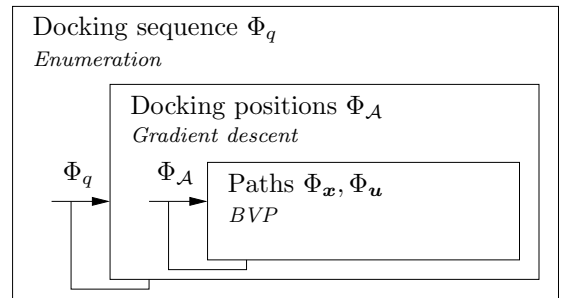


Fig. 3. Structure of hierarchical optimal control algorithm

First, the algorithm generates all possible docking sequences  $\Phi_q$  that are in accordance with the transition maps in  $\mathcal{G}$ . Furthermore, the algorithm has to be initialized with one feasible location sequence  $\Phi_q$  and  $L - 1$  docking sites  $\Phi_{\mathcal{A}}$ . As a consequence,  $L$  TPBVPs are considered on the lower layer and are solved for the initialized docking positions. By gradient-descent applied to the vector of docking sites  $\Phi_{\mathcal{A}}$ , the docking configuration is varied until the optimal sequence of docking sites  $\Phi_{\mathcal{A}}^{\text{opt}}$  is found. The overall cost as the sum of costs of the  $L$  TPBVPs has to be computed for each variation of the docking positions. The information stored with each optimized feasible run  $\Phi_q$  is a structure  $HR = (\Phi_q, \Phi_{\mathbf{u}}, \Phi_{\mathbf{x}}, \Phi_{\mathcal{A}}, t_e, J)$  and is passed to the upper layer. The algorithm continues with a re-initialization using the next location schedule  $\Phi_q$  and an initial sequence  $\Phi_{\mathcal{A}}$ , and it calculates the optimal values  $HR$ . In this enumerative approach, the algorithm finishes when all possible docking sequences have been examined. The best plan for the vehicles' transportation scheme is the one with lowest cost index  $J$ .

### 3.1 Nonlinear Programming for embedded Continuous Subproblems

The embedded continuous subproblem is to find a discretized trajectory of the control inputs  $\Phi_u$  leading to a discretized state trajectory  $\Phi_x$  via (4), such that the transportation is successful within each discrete state of the sequence fixed by the higher layer. Here, a discrete state sequence  $\Phi_q$ , a sequence of docking sites  $\Phi_A$ , and the initial state vector  $\mathbf{x}^0$  of the system are assumed to be provided. The TPBVPs are solved in a two-step procedure: First, a shooting method is used to provide a sufficiently good initial solution. This solution is then further improved using a collocation technique, see W.H.Press et al. (1992). In both steps, one boundary is given by the initial system configuration. The second boundary corresponds to either a docking event or the *TO* target configuration.

In the shooting approach, the second boundary of a single TPBVP is considered as a soft constraint by implementing a penalty term  $\vartheta_{\text{pen}}(\mathbf{x}(t_{k+1}))$  into the local cost function  $J_{\text{loc}}$ .

$$J_{\text{loc}} = \vartheta_{\text{pen}}(\mathbf{x}(t_{k+1})) + \int_{t_k}^{t_{k+1}} \mathbf{u}_q^T \mathbf{u}_q + \mu_k d\tau. \quad (11)$$

The penalty term penalizes deviations of the vehicles boundary state  $\mathbf{x}(t_{k+1})$  from the desired docking position. The system dynamics (7) is satisfied by numerical integration with fixed step-width. The free variables (discretized control input trajectory  $\mathbf{u}_q$  and end time  $t_{k+1}$ ) are adjusted by gradient search until the local cost function  $J_q$  is minimized. In general, the desired docking position is not met exactly and the optimal trajectory is refined by the collocation algorithm.

For the collocation method, the system ODEs (7) are replaced by finite difference equations on a mesh of collocation points that cover the range of integration. The initial solutions for the control inputs and the state variables are obtained from the preceding shooting and are assigned to the appropriate mesh points. A polynomial is sought which satisfies the ODEs at the collocation points and the boundary conditions at the initial and the final time. The shooting and the collocated problem can be solved using the MATLAB routine 'fmincon', which provides an implementation of Sequential Quadratic Programming.

### 3.2 Combinatorics versus Suboptimality

In particular the optimization of the docking configurations is a computational costly task, as the component-wise gradient calculation requires to run the embedded continuous control problems of the lower layer for every variation of docking sites on the middle layer. Therefore, testing all feasible location schedules of a graph structure is very time consuming. It is proposed here to substitute the optimization of docking sites on the middle layer by an approximation of optimized docking sites. This decouples the discrete and continuous dynamics, such that methods of dynamic programming can be used to determine the optimal discrete sequence.

An overview on the decoupling algorithm is given in Fig. 4. For every sequence of locations  $\Phi_q$  on the upper layer, an approximation for the sequence of docking sites  $\Phi_A$

is calculated on the middle layer. Thus, the problem of finding the appropriate docking sequence with its docking sites is decoupled from finding the continuous controls on the lower layer. Therefore, branches in the tree which are not likely to lead to optimal solution candidates can be pruned. This approach is an instance of dynamic programming and implemented using a depth-first label correcting technique, see e.g. Bertsekas (2006). The final candidates are further evaluated using the enumerative optimization approach. Fig. 5 depicts the weighted tree structure for three vehicles with pruned branches.

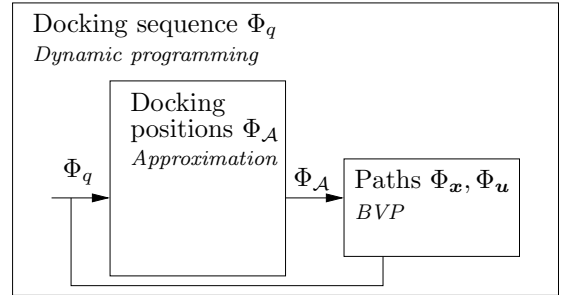


Fig. 4. Structure of the hierarchically decoupled optimal control algorithm.

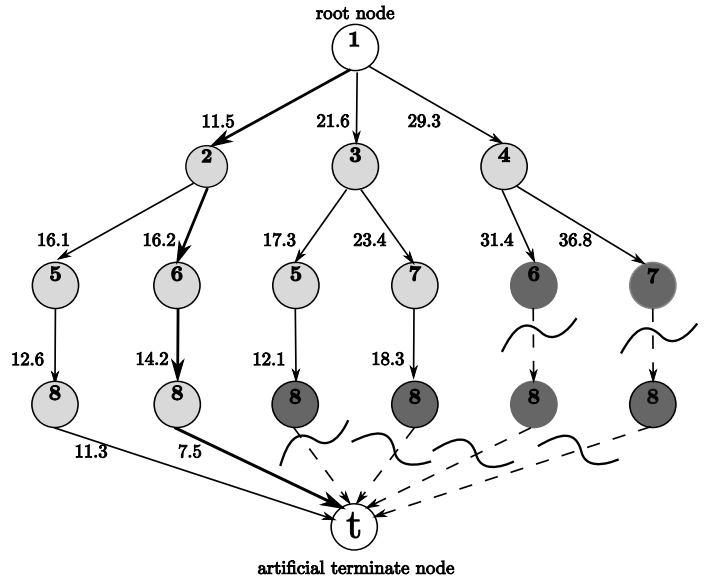


Fig. 5. Label correcting technique: transitions where accumulated costs exceed an upper limit are disabled. The given numbers provide transition costs for one transition.

Different approaches are proposed to approximate sub-optimal docking configurations - a partitioning of the state space into polygonal regions with focal points and two optimization based consensus techniques.

For the first approach, named *centroid approach*, the docking configurations are approximated as centroids of a triangular region. For a fixed docking sequence  $V_1, \dots, V_N$  (received from the upper layer), the corners of the centroid (that defines the docking position of the vehicle  $V_i$  are the current position of the transported object *TO* (with vehicles  $V_1, \dots, V_{i-1}$ ), the initial position of vehicle  $V_i$ , and the initial position of vehicle  $V_{i+1}$  that will dock

afterwards. Thus, the sub-optimal docking configuration is assumed to be the average of the positions of the vehicles that are considered within an action horizon of three vehicles.

For the second approach, the docking position of vehicles and transported object is the solution of an optimization problem. There, a consensus configuration between one agent and the object is sought with respect to energy and time optimality and used as docking position. Thus, for the definition of the docking positions, small subproblems of the total optimal control problem from Sec. 3.1 are solved.

The objective of the related TPBVP on  $[t_k, t_{k+1}[$  is to remove the docking error for  $t_{k+1}$ :

$$\vartheta_{\text{err}}(\mathbf{x}(t_{k+1})) = \|\mathbf{x}_{V_i}(t_{k+1}) - \mathbf{x}_{TO}(t_{k+1})\|, \quad (12)$$

while initial values are given for  $t_k$  and only the dynamics of one vehicle  $V_i$  and the transported object  $TO$  are considered. Energy and time optimality of the docking configuration is assured by additionally considering the original objective function:

$$\tilde{J} = \vartheta_{\text{err}}(\mathbf{x}(t_{k+1})) + \int_{t_k}^{t_{k+1}} \mathbf{u}^T \mathbf{u} + \mu_k \, d\tau. \quad (13)$$

The approximated docking point is chosen to be the final value of the obtained transport object trajectory  $\mathbf{x}_{TO}(t_{k+1})$ . This approach is termed *direct consensus technique*.

As an extension, the *finite horizon consensus* approach seeks to involve future dockings into the calculation of optimized docking sites. An optimization based consensus method for a horizon of two transitions minimizes the error:

$$\vartheta_{\text{err}}(\mathbf{x}_q(t_{k+1})) = \|\mathbf{x}_{V_i}(t_{k+1}) - \mathbf{x}_{TO}(t_{k+1})\| + \|\mathbf{x}_{V_{i+1}}(t_{k+1}) - \mathbf{x}_{TO}(t_{k+1})\|, \quad (14)$$

instead of (12).

In all of these alternatives, the configuration of the free vehicles still has to be defined after finding the docking sites. These configurations are approximated to be on the path towards their future docking position.

#### 4. NUMERICAL RESULTS OF OPTIMIZATION

In the following, the solutions of the decoupling approaches are discussed in comparison to the solution of the enumerative optimization. In Fig. 6, the solution of the enumerative approach is displayed as reference solution for an example. For a setup with three vehicles, the computation time is about 14 hours, while the optimal solution of a single TPBVP on the lower layer is computed in average within approx. 12 seconds.

The solution obtained from the centroid technique is shown in Fig. 7. In comparison to the optimal reference solution in Fig. 6, the accumulated costs are, of course, higher (as the solution is sub-optimal), and also the sequence of vehicle docking differs. The same holds for the solution obtained from the direct consensus approach depicted in Fig. 8.

In comparison to the direct consensus approach (Fig. 8), solutions of the centroid technique (Fig. 7) are computationally less expensive, resulting from cheaper vehicle

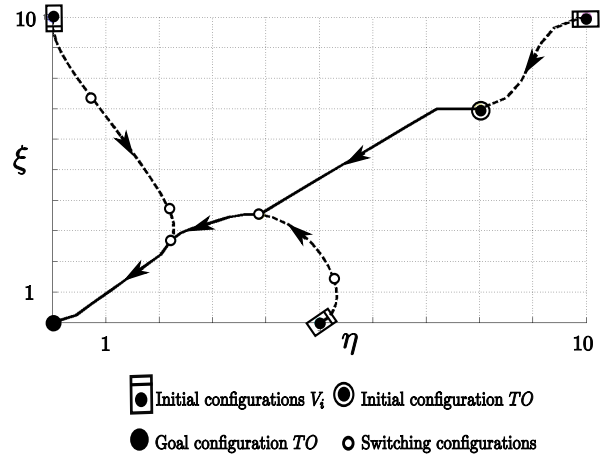


Fig. 6. Optimal solution with enumerative optimization:  $J = 38.52$ .

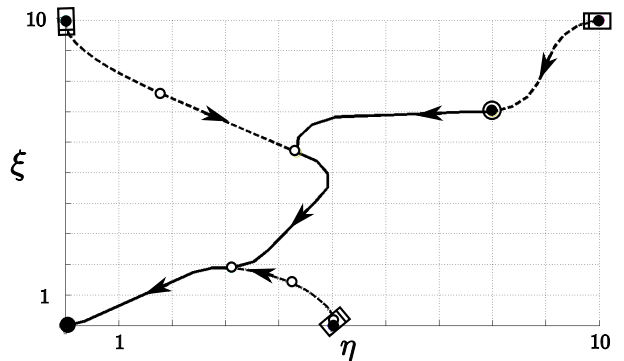


Fig. 7. Optimized solution with the centroid technique:  $J = 42.10$ .

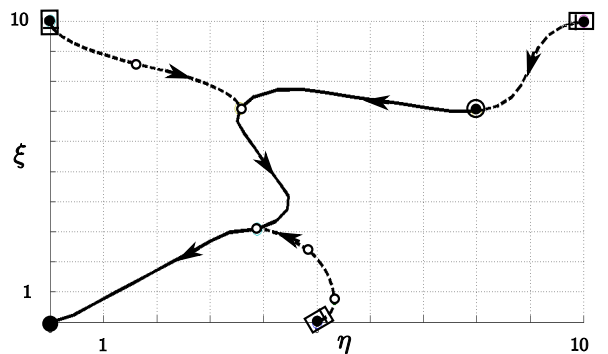


Fig. 8. Optimized solution with the direct consensus approach:  $J = 46.12$ .

trajectories, see Table 1. This effect may surprise, because the computed consensus docking configurations are assumed to represent the closest configuration between the interacting vehicle and  $TO$ , while referring to a distance measure that considers the real-valued output of the objective function. Indeed, this assumption holds for the first docking, whereas in the subsequent locations higher costs are aggregated. The calculation of consensus configurations considering only the currently involved vehicles does not contribute significantly to the performance in a global context. Thus, locally this approach leads to

optimal configurations, whereas the performance is worse for the complete evolution.

Table 1. Performance on a 1.6 GHz Intel Centrino platform with 1.25 GB RAM.

Method	Time [h]	Cost	Sequence
enumerative	14	38.52	$V_1, V_2, V_3$
centroid technique	0.16	42.10	$V_1, V_3, V_2$
direct consensus	0.25	46.12	$V_1, V_3, V_2$
finite horizon consensus	0.5	41.28	$V_1, V_2, V_3$

Due to this issue, the finite horizon consensus approach is proposed and applied in order to combine the advantages of the other two techniques. In Fig. 9, the corresponding optimized trajectories are depicted. The similarity of these sub-optimal trajectories with the optimal ones in Fig. 6 is significant. The numerical calculations to obtain this result took approximately 30 minutes (compared to 14 hours for the enumerative approach). Note, that the first two solution approaches led to the docking sequence  $V_1 - V_3 - V_2$ , and only the finite horizon consensus method produces the docking sequence  $V_1 - V_2 - V_3$  of the reference solution. However, with respect to the costs of the overall evolution, both docking sequences lead to relatively similar values. Finally, note that the centroid method is the fastest technique and produced solutions in approximately 10 minutes.

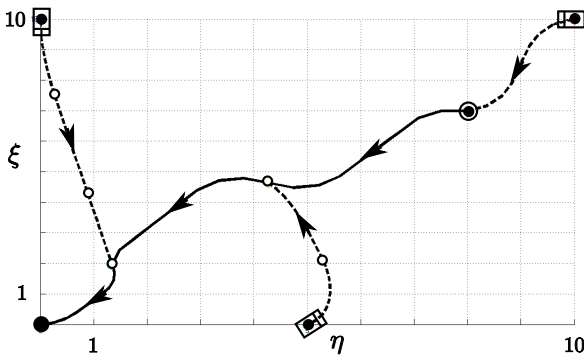


Fig. 9. Optimized solution with the finite horizon consensus approach:  $J = 41.28$ .

## 5. CONCLUSION

A method to model and optimize cooperative transportation planning in a multi-agent setting has been proposed. For this application, the discrete and the continuous dynamics are strongly coupled at the level of optimal docking sites and times. In comparison to a brute-force enumerative approach for optimizing the motion of the agents, a considerable decrease of computational time can be achieved by allowing sub-optimal solutions for the docking sites and times. With three different methods for approximating the sites and times, similar results are obtained which are not considerably worse with respect to the costs of the overall execution than obtained for the reference solution, while the time of computation is reduced by more than one order of magnitude. Among the proposed techniques, the finite horizon optimization-based consensus approach was found to provide the best compromise of

optimization result and time of computation for the investigated example. Further investigations will predominantly address the scalability of the method.

## ACKNOWLEDGEMENTS

Daniel Engelhardt's contribution to preliminary results is gratefully acknowledged.

## REFERENCES

- Alur, R., Courcoubetis, C., Halbwachs, N., Henzinger, T., Ho, P.H., Nicollin, X., Olivero, A., Sifakis, J., and Yovine, S. (1995). The algorithmic analysis of hybrid systems. *Theor. Comput. Sci.*, 138(1), 3–34.
- Bertsekas, D.P. (2006). *Dynamic Programming and Optimal Control*. Prentice-Hall.
- Branicky, M., Borkar, V., and Mitter, S. (1998). A unified framework for hybrid control: Model and optimal control theory. *IEEE Trans. Automat. Contr.*, 43(1), 31–45.
- Dunbar, W. and Murray, R. (2006). Distributed receding horizon control for multi-vehicle formation stabilization. *Automatica*, 42(4), 549–558.
- Earl, M. and D'Andrea, R. (2002). Modeling and control of a multi-agent system using mixed integer linear programming. In *Proc. IEEE Conf. on Decision and Control (CDC)*.
- Fierro, R., Das, A., Kumar, V., and Ostrowski, J. (2001). Hybrid control of formations of robots. In *Proc. IEEE Conf. on Robotics and Automation (ICRA)*.
- Gokbayrak, K. and Cassandras, C. (2000). Hybrid controllers for hierarchically decomposed systems. In *Hybrid Systems: Computation and Control*, volume 1790 of *LNCS*, 117–129. Springer.
- Jadbabaie, A., Lin, J., and Morse, A. (2003). Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Trans. Automat. Contr.*, 48(6), 988–1001.
- Li, W. and Cassandras, C. (2006). A cooperative receding horizon controller for multivehicle uncertain environments. *IEEE Trans. Automat. Contr.*, 51(2), 242–257.
- Lynch, N., Segala, R., and Vaandrager, F. (2003). Hybrid I/O automata. *Inf. Comput.*, 185(1), 105–157.
- Murray, R.M. (2007). Recent research in cooperative control of multi-vehicle systems. *ASME Journal of Dynamic Systems, Measurement and Control*, 129(5), 571–583.
- Reinl, C., Glocker, M., and von Stryk, O. (2009). Optimal control of cooperative multi-vehicle systems. *AT-Automatisierungstechnik*, 57(6), 296–305.
- Shaikh, M. and Caines, P. (2007). On the hybrid optimal control problem: Theory and algorithms. *IEEE Trans. Automat. Contr.*, 52(9), 1587 – 1603.
- Stursberg, O. (2004). Dynamic optimization of processing systems with mixed degrees of freedom. In *7th IFAC Symposium on Dynamics and Control of Process Systems*.
- W.H.Press, Teukolsky, S., Vetterling, W., and Flannery, B. (1992). *Numerical Recipes in C, Second Edition*. Cambridge University Press.
- Xu, X. and Antsaklis, P. (2004). Optimal control of switched systems based on parameterization of the switching instants. *IEEE Trans. Automat. Contr.*, 49(1), 2–16.