Consensus Based Decoupling in Hybrid Optimization of Cooperative Transportation Planning

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

* *Institute of Automatic Control Engineering, Technische Universität München, Germany*
** *Control and System Theory, Universität Kassel, Germany*

Abstract: A recently introduced new class of multi-agent transportation planning is considered. Several agents have to cooperate in order to transfer a passive agent from its initial configuration to a goal configuration. Docking events refer to discrete decisions, whereas the individual agents are governed by nonlinear continuous dynamics. Modeling this problem in a hybrid optimal control framework allows computing a solution using a centralized, hierarchical optimization algorithm. Relatively well performing solutions can be found with much lower computational time by decoupling the discrete-event level from the continuous processes. This is accomplished by introducing a feedback controller based on a consensus protocol into the hierarchical optimization structure. The scalability and optimality of this approach is examined for varying information topologies in the consensus protocol and controller gains.

Keywords: Hybrid systems modeling and control; Multi-agent systems; Cooperative systems

1. INTRODUCTION

The multi-agent system (MAS) approach found its way into a wide field of applications. The key characteristic is that local (agent) properties in a MAS may influence any other local agent's properties and thus condition the global behavior by its interconnected dynamics. Global behaviors and switchings can be described by discrete variables and transitions between them, which follow underlying continuous dynamic processes, see Olfati-Saber (2007). Therefore, hybrid systems are widely used for modeling MAS (see Fierro et al. (2001)) as they can capture both logic-based decisions and continuous control in one model. For technical application, the performance of executing a task by a MAS is of fundamental interest. Usually, the overall task performance can not be represented by the sum of the agents' performances in their individual task execution. Those systems operate in a cooperative manner, whereat an optimal cooperative control strategy for a given task depends on global knowledge, and hence, it has to be solved in a centralized scheme, see Parker (1993).

For such performance-oriented and distributed hybrid control problems, exact methods for optimal planning and scheduling have been seen to suffer from the "curse of dimensionality". This relates to dynamic optimization solved by backward induction as means of a divide-and-conquer principle, where every possible information set is explored first, before a decision about a Pareto-optimal execution plan is taken. Computational intractability due to combinatorial explosion is the consequence.

Hence, classical hybrid optimal control which computes centralized strategies by hierarchical optimization algorithms (Shaikh and Caines (2007)) are applicable only for small-scale cooperative problems with the number of agents being in the order of 10^1. Tools to reduce computational complexity such as Lagrangian relaxation for switched linear systems or heuristic methods as genetic algorithms and evolutionary programming are available. For the latter tools, however, quality of the resulting global execution plans can hardly be controlled. Recently, Passenberg et al. (2010) introduced an algorithm, that combines the exploitation of logic-based schedules with the underlying exploration of continuous state trajectories in one scheme. The idea is to transform the globally optimal control problem to a problem depending on switching points, which comply with a given geometric structure. Hence, the two functional layers of exploitation on the discrete-event level and optimization of individual continuous agent trajectories can be combined in one functional layer. However, for practical and especially large-scale problems switching manifolds are often unknown and can hardly be computed in a tractable way, concerning the time scale of the MAS evolution. Therefore, a near-optimal approximation of a switching structure is of interest.

The purpose of the current work is two-fold. First, in addition to extensively studied generic tasks for MAS, see Murray (2007) for reference, here a novel cooperative tasking scenario is considered: several agents have to cooperate in order to transfer a passive agent from its initial configuration to a goal configuration. On a discrete-event level it has to be decided in which order agents will take part in transportation by sequentially docking to the passive object. The continuous problem generates state trajectories as well as the configurations and times of docking; overall, this yields a three-layer hierarchical algorithmic structure described in Mangesius et al. (2010). Secondly, a new approximative method to reduce the computational complexity in the hybrid optimal control prob-

lem is introduced: near optimal switching configurations are computed time-efficiently by merging two algorithmic layers.
Sec. 2 introduces the MAS setting, its modeling, and the optimal control problem. Sec. 3 briefly explains the hierarchical optimization procedure, highlights the coupling of discrete and continuous dynamics on a middle layer, and provides the theoretical account for decoupling using a consensus protocol. Sec. 4 presents numerical results, and Sec. 5 concludes the paper with an outlook to future research.

2. TRANSPORTATION PROBLEM

In the MAS transportation problem, illustrated in Fig. 1, the agents $A_1, \dots, A_N$ have to cooperate in order to transport the targeted agent TA from its initial to a given target position. The agent TA is passive and can only be transferred in the 2D Euclidean configuration space by one or many small agents docking to and actively moving it. Realizing the cooperative task with maximum performance requires to determine the best suited sequence for the agents to reach TA as well as the most suitable docking positions and docking times. The set $\mathcal{D}_A(t)$ indexes the docked (active) agent(s) transferring TA at time t. Conversely the indices of those vehicles that have not yet reached TA are collected in the set $\mathcal{M}_A(t)$.

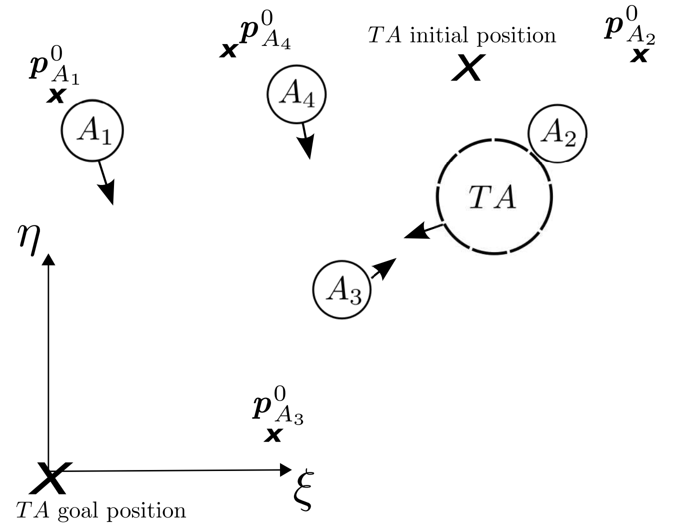

Fig. 1. Scenario: agents $A_1, \dots, A_4$ cooperate to transport object TA from an initial to a goal position. The arrows show the heading of the agents moving in a plane spanned by the axes ξ and η. The initial positions of the vehicles are denoted by $\boldsymbol{p}^0_{A_1}, \dots, \boldsymbol{p}^0_{A_N}$.

2.1 Hybrid System Model

In a hybrid system model, the transportation planning scenario with multiple agents is mathematically described by collections of dynamical systems that evolve in continuous-variable state spaces and are subject to continuous control and discrete transitions. A continuously-controlled autonomous-switching hybrid dynamical system formally is given by the tuple

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

(see Branicky et al. (1998)), with components modeled for the MAS setting as follows:

$Q=\{q_1, \dots, q_{N_q}\}$ is the finite *set of discrete states* with the discrete states $q(t) \in Q$. A discrete state is assigned to every possible set of indices of docked agents in $\mathcal{D}_A(t)$. The MAS evolves continuously within each q, and docking events refer to switchings between two discrete states at a certain switching time t_k. Therefore the MAS evolves piecewise continuously through sequenced discrete states. The times t_k are the boundaries of sequenced time intervals τ_k, which are ordered in the set

$$\Lambda = \{[t_0; t_1[, [t_1; t_2[, \dots, [t_{L-1}; t_e]\}. \tag{2}$$

$\Sigma = \{\Sigma_q\}_{q \in Q}$ is the *collection of controlled dynamical systems*, where each $\Sigma_q = [X_q, \boldsymbol{F}_q, U_q]$ is a continuously controlled dynamical system. The continuous state vectors $\boldsymbol{x}_q(\text{t})$ of dimension $n_{\boldsymbol{x}_q} \in \mathbb{N}^+$ are defined on the continuous state spaces $X_q \subseteq \mathbb{R}^{n_{\boldsymbol{x}_q}}$. Accordingly, the continuous control inputs $\boldsymbol{u}_q(\text{t})$ of dimension $n_{\boldsymbol{u}_q} \in \mathbb{N}^+$ are defined on the continuous input spaces $U_q \subseteq \mathbb{R}^{n_{\boldsymbol{u}_q}}$. The time-invariant vector fields $\boldsymbol{F}_q : \mathbb{R}^{n_{\boldsymbol{x}_q}} \times \mathbb{R}^{n_{\boldsymbol{u}_q}} \to \mathbb{R}^{n_{\boldsymbol{x}_q}}$ provide the continuous dynamics of $\mathbb{H}$.
The state space X_q for the MAS is defined as the product of the configuration spaces of the freely evolving agents and the TA (in case the latter is transported), so that

$$X_q = C^{A_1} \times C^{A_2} \times \dots \times C^{A_{|\mathcal{M}_A|}} \times C^{TA}. \tag{3}$$

Here, unicycle dynamics are chosen as nonlinear model for agent motion:

$$\boldsymbol{x}_{A_i} = (\boldsymbol{p}^T, \Theta)^T_{A_i}, \quad \text{and} \tag{4}$$

$$\dot{\boldsymbol{x}}_{A_i} = \begin{pmatrix} v\cos(\Theta) \\ v\sin(\Theta) \\ \omega \end{pmatrix}_{A_i}, \quad C^{A_i} = \mathbb{R}^2 \times [-\pi; \pi] \tag{5}$$

The position vector $\boldsymbol{p}_{A_i} = (\xi, \eta)^T_{A_i}$ describes the configuration of agent i in the 2D Euclidean plane; the control inputs are $\boldsymbol{u}_{A_i} = (v, \omega)^T$, where ω directly sets the angular velocity of the orientation and v sets the speed. The agent TA moves passively, i.e. it has to be transferred by some agent(s) and hence its control is:

$$\boldsymbol{u}_{TA} = f(b(t))\boldsymbol{u}_{A_i}. \tag{6}$$

For $f(b)$ applies that $f(b) = \tanh(\frac{2}{N_A}b)$ with $b(t) = |\mathcal{D}_A(t)|$, i.e. the cardinality of the set of docked agents can be seen as an activation gain: the continuous dynamics of TA changes to higher velocity with every additionally agent docking to TA.

The state $\boldsymbol{x}_q \in X_q$ specifies the configurations of all agents and is equal to

$$\boldsymbol{x}_q = (\boldsymbol{x}^T_{A_1}, \boldsymbol{x}^T_{A_2}, \dots, \boldsymbol{x}^T_{A_{|\mathcal{M}_V(t)|}}, \boldsymbol{x}^T_{A_{TA}})^T, \tag{7}$$

where the dimension of $\boldsymbol{x}_q$ is $n_{\boldsymbol{x}_q} = \sum_{i=1}^{|\mathcal{M}_A(t)|} \dim(C^{A_i}) + \dim(C^{TA}) = (|\mathcal{M}_A(t)| + 1) \times 3$.
The differential equation system for the MAS finally is

$$\dot{\boldsymbol{x}}_q = \boldsymbol{F}_q(\boldsymbol{x}_q, \boldsymbol{u}_q), \tag{8}$$

with $\boldsymbol{F}_q$ and $\boldsymbol{u}_q$ in accordance with the state vector in (7).

$\mathcal{A}= \{\mathcal{A}_q\}_{q \in Q}$ 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 this paper, $\mathcal{A}_q$ is used to model the docking process when an agent reaches TA:

$$\mathcal{A}_q = \{ \boldsymbol{x}_q \mid \boldsymbol{x}_{A_i} = \boldsymbol{x}_{TA},\, i \in \mathcal{M}_A \} \,. \tag{9}$$

Within $\mathcal{A}_q$, the position in which docking occurs is denoted by $\boldsymbol{x}_q^S$.

$\mathcal{G} = \{\mathcal{G}_q\}_{q \in Q}$ is the *autonomous jump transition map* with hybrid transition functions $\mathcal{G}_q : \mathcal{A}_q \times Q \to X_q \times Q$. Hence, $\mathcal{G}_q$ contains all possible discrete transitions $q \to q^+$ and updates of the continuous states.

A set of logic rules sets bounds on the number of discrete states and transitions between them. In the following, a MAS is considered where docking occurs sequentially and

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

Therefore, the number of discrete states in a feasible run is $L = N_V + 1$. With these rules a discrete structure arises that can be identified as an acyclic directed graph. A possible ratio for the last assumption is that a preceding analysis has already determined the set of agents that should contribute to the transportation task. Undocking or multiple dockings would result in a more complex discrete structure - technically, a consideration of additional discrete transitions would only result in an increased computational effort, due to increasing combinatorial complexity, without adding direct value to the basic idea.

Let $\Phi_q = (q(t_0), q(t_1), q(t_2), \ldots, q(t_{L-1}))$ be the discrete state schedule, $\Phi_{\mathcal{S}} = (\boldsymbol{x}^S_{q^0}, \boldsymbol{x}^S_{q^1}, \ldots, \boldsymbol{x}^S_{q^{L-1}})$ the sequence of switching configurations, and $\Phi_{\boldsymbol{x}}$ and accordingly $\Phi_{\boldsymbol{u}}$ refer to the collection of continuous state trajectories and control input trajectories. Then, $\Phi_{\boldsymbol{\sigma}} = (\boldsymbol{\sigma}_0(\tau_0), \boldsymbol{\sigma}_1(\tau_1), \ldots, \boldsymbol{\sigma}_{L-1}(\tau_{L-1}))$, $\tau_i \in \Lambda$ according to (2) is a hybrid state trajectory with a hybrid state being defined as $\boldsymbol{\sigma}(t) = (\boldsymbol{x}_q(t), q(t))$, $\boldsymbol{\sigma} \in \bigcup_{q \in Q} X_q \times \{q\}$.

2.2 *Optimal Control Problem*

The objective is to convey the agent TA from its initial configuration to a final configuration in an energy and time optimal manner involving all agents. The following hybrid optimal control problem is considered: Find a discrete state schedule Φ_q, a sequence of switching points $\Phi_{\mathcal{S}}$, and continuous control inputs $\Phi_{\boldsymbol{u}}$ such that the hybrid state trajectory $\Phi_{\boldsymbol{\sigma}}$ of the hybrid dynamical system $\mathbb{H}$ satisfies the hybrid boundary value problem

$$\boldsymbol{\sigma}(t_0) = (\boldsymbol{x}_q^0, q(t_0)), \quad \boldsymbol{\sigma}(t_e) = (\boldsymbol{x}_q^e, q(t_e)), \quad t_e \ \text{ free}, \tag{10}$$

and minimizes the real-valued cost function

$$J = \sum_{k=0}^{L-1} \left\{ \int_{t_k}^{t_{k+1}} \boldsymbol{u}^T_{q(t_k)} \boldsymbol{R} \boldsymbol{u}_{q(t_k)} + \mu_k \, \mathrm{d}\tau \right\}. \tag{11}$$

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

Finally, let the solution of the optimization problem be denoted by $\Phi_q^*, \Phi_{\mathcal{S}}^*, \Phi_{\boldsymbol{u}}^*$.

3. SOLUTION APPROACHES FOR THE HYBRID MULTI-AGENT CONTROL PROBLEM

The hybrid optimal control problem (HOCP) is solved centrally by separating the optimization of the continuous and the discrete dynamics, similar to Shaikh and Caines (2007). The ensuing hierarchical structure constitutes a three-layer approach as illustrated in Fig. 2: on the upper level the optimal discrete state sequence is determined, whereas the optimal continuous trajectories are computed in the inner layer as solutions of two-point boundary value problems (TPBVP). An additional intermediate level is introduced that accounts for the coupling of optimal discrete and optimal continuous dynamics via (sub-)optimal autonomous switching configurations, i.e. it determines the docking positions.

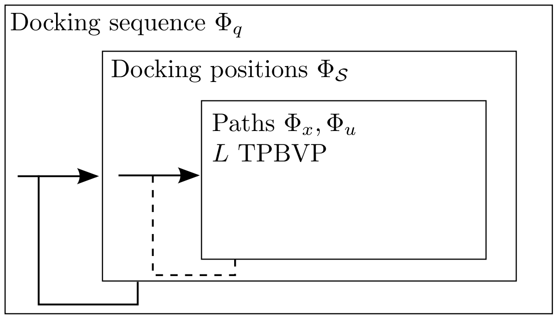

Fig. 2. Structure of the hierarchical solution algorithm; the dashed feedback line connecting inner and middle layer solutions indicates two versions of computing docking positions: optimization based and consensus approximations.

3.1 *Coupling between Discrete and Continuous Dynamics*

The continuous trajectories of a run of $\mathbb{H}$ consists of L sequenced solution trajectories obtained from the solution of embedded TPBVPs. Each of these subproblems requires initial and final boundaries. Consequently, a sequence of L pairs of boundary configurations is associated to each discrete state schedule:

$$\{[(x^0, q^0); \boldsymbol{x}^{\mathcal{S}}_{q^0}], [\boldsymbol{x}^{\mathcal{S}}_{q^0}; \boldsymbol{x}^{\mathcal{S}}_{q^1}], \ldots, [\boldsymbol{x}^{\mathcal{S}}_{q^{L-1}}; (x^e, q^e)]\} \hookleftarrow \Phi_q \tag{12}$$

An optimal overall transportation plan requires local optimality, i.e. each local TPBVP is solved optimally , denoted as TPBVP*. However, in view of global optimality, the linear combination of cost functionals of L locally optimal continuous problems has to be considered. In that sense, let $\tilde{J}$ denote a locally optimal cost index. Then, the globally best discrete state sequence that represents the Pareto-optimal solution of the optimal control problem can be obtained from rewriting (11) such that

$$\begin{aligned} \Phi_q^* &= \arg\inf\{\, \tilde{J}(\Phi_{\boldsymbol{\sigma}})\} \\ \tilde{J}(\Phi_{\boldsymbol{\sigma}}) &= \min_{\Phi_{\mathcal{S}} \hookleftarrow \Phi_q} \sum_{q^k \in \Phi_q} \tilde{J}(\mathrm{TPBVP}^*(\boldsymbol{x}^{\mathcal{S}}_{q^k})) = \\ &= \tilde{J}(\Phi^*_{\mathcal{S}}), \quad \forall \Phi_q \in \mathcal{G} \end{aligned} \tag{13}$$

From this representation, it can be seen that the optimal discrete state schedule is linked with the optimal continuous trajectories constituting TPBVP* via the docking positions which represent the free variables in (13).

3.2 Brute-Force Enumerative Hierarchical Optimization

In the brute-force approach, each of the loops in Fig. 2 uses the optimal value of the cost function from the embedded problem at the next lower level as performance index for appropriate variations of the respective argument of its own level. In order to adequately measure changes in the cost index due to variations in an outer level, all optimal solutions of the embedded inner sub-problems need to be recalculated. By that, an embedded subproblem can be considered as a 'black-box' where the input (varied argument) and the output (corresponding cost index) are the only measurable quantities from the perspective of the related outer optimization layer.
First, the algorithm generates all possible docking sequences Φ_q that are in accordance with the transition map $\mathcal{G}$. Furthermore, the algorithm has to be initialized with one feasible discrete state sequence Φ_q and $L-1$ docking positions $\Phi_{\mathcal{S}}$. As a consequence, L TPBVP are considered in the lower layer and are solved for the initialized docking positions. By gradient-descent applied to the vector of docking positions $\Phi_{\mathcal{S}}$, the docking configuration is varied until the optimal sequence of docking positions is found. The overall cost index as sum of the cost indices of the L TPBVP has to be computed for each variation of the docking positions. The information stored with each optimized feasible run Φ_q is a structure HR $= (\Phi_q, \Phi_{\boldsymbol{u}}, \Phi_{\boldsymbol{x}}, \Phi_{\mathcal{S}}, t_e, J)$ and is passed to the upper level. The algorithm continues with a re-initialization using the next discrete state schedule Φ_q and an initial sequence $\Phi_{\mathcal{S}}$, 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.

The hierarchically operating, centralized gradient-descent algorithm, together with the combinatorial complexity of this enumerative approach, leads to computational times in the scale of hours for already small numbers of cooperating agents , e.g. $N_A = 3$, as demonstrated in Mangesius et al. (2010).

3.3 Decoupled Hybrid Optimization by Approximation of Sub-Optimal Docking Positions Using A Consensus Feedback Controller

Other than in the brute-force methodology, the ex ante knowledge about docking positions allows to decouple the discrete and continuous dynamics and thereby enables the construction of a weighted transition graph. By that, shortest path algorithms can be applied in a tree search as an instance of dynamic programming. This enables relatively time efficient computation of solutions, which perform good in comparison to the optimal solution trajectories. Within such sub-optimal strategies only one free agent tries to reach TA per discrete state, what allows for distributed planning coordinated by one supervisor. Near-optimal cooperative strategies are then approximated starting from these trajectories.
A consensus feedback controller is used to approximate docking positions, whereat the controller gains and the receding horizon of the static information topology in the nonlinear consensus protocol are the adjustable parameters.
The degree of optimality of an approximated single docking position not only depends on the one agent to dock next but also on one or more future docking decisions. To reflect this, the receding horizon parameter r is introduced, which together with Φ_q defines the ordered set of agents forming a cooperative group that is subject to the consensus feedback controller. The goal within this group is to agree on a common information state, i.e. $|\boldsymbol{x}_{Ai} - \boldsymbol{x}_{Aj}| \to 0,\ \forall i \neq j$ as $t \to \infty$, that is the rendez-vous configuration.
Let the docking schedule Φ_q and the number r of agents influencing the rendez-vous approximation of one docking configuration be given. Then, an undirected and connected information graph encodes the information topology that captures the state information exchange between the agents, and it is represented by the normed graph Laplacian

$$L_r = D_r^{-1}(D_r - Adj_r). \tag{14}$$

Here, D_r is the degree matrix of the information graph and Adj_r the adjacency matrix; both matrices are symmetric and of dimension $(r+1)\times(r+1)$. For the case of a two agent receding horizon the Laplacian is

$$L_r = \begin{pmatrix} 1 & -0.5 & -0.5 \\ -0.5 & 1 & -0.5 \\ -0.5 & -0.5 & 1 \end{pmatrix}. \tag{15}$$

Let $\boldsymbol{p}^T = (\boldsymbol{p}_{TA}^T, \boldsymbol{p}_{A_1}^T, \ldots, \boldsymbol{p}_{A_{r-1}}^T)$ be the stack vector of Euclidean position measures, then the system level error operation on Euclidean positions $\delta\boldsymbol{p}$ is given by

$$\delta\boldsymbol{p} \equiv (L_r \otimes I_2)\boldsymbol{p}. \tag{16}$$

Here, $\otimes$ denotes the Kronecker product, which assures separated computation of errors in ξ and η, while explicitly taking into account the parameterization with the information topology. The entries of the normed Laplacian thereby represent the weights in the averaging process of calculating the individual errors. To compute the overall heading error associated to the control input of one agent, the relative angle between two agents A_i and A_j is given as

$$\Psi_{A_{ij}} = \arctan\left(\frac{\eta_j - \eta_i}{\xi_j - \xi_i}\right). \tag{17}$$

In accordance with Dimarogonas and Kyriakopoulos (2007), $\arctan(\frac{0}{0}) = 0$ is used. Then, with the stacked vector $\boldsymbol{\Theta}^T = (\Theta_{TA}, \Theta_{A_1}, \ldots, \Theta_{A_r})$ and the $(r+1)\times(r+1)$ heading error matrix Υ, which is build element-wise by

$$[\Upsilon]_{ii} = \Theta_{A_i} \quad \text{and} \quad [\Upsilon]_{ij} = \Psi_{A_{ij}},\ \ \forall i,j \in \{1,\ldots,r\}, \tag{18}$$

the heading error operation is defined as the contraction

$$\delta\boldsymbol{\Theta} \equiv L_r : \Upsilon = tr(L\Upsilon^T). \tag{19}$$

Finally, let $\boldsymbol{t}_{A_i} = (\cos\Theta_{A_i}, \sin\Theta_{A_i})^T$ denote the ith agent's unit heading vector. Similar to Listmann et al. (2009), a control strategy that achieves a consensus on the Euclidean distance measure and the headings between the agents $\{A_i\}$ within a receding horizon is

$$v_{A_i} = -g_{A_i}\delta\boldsymbol{p}_{A_i}^T\boldsymbol{t}_{A_i} \tag{20}$$

$$\omega_{A_i} = -g_{A_i}\delta\Theta_{A_i}. \tag{21}$$

Here, g denotes a controller gain, and the δ-operator applied to the position and heading coordinates represents the respective overall error between one agent and all other agents within a receding horizon.
The receding action horizon is adjusted after convergence to the rendez-vous configuration along the provided docking sequence Φ_q until $N_A - 1$ docking positions are approximated.
Having a set of docking positions approximated, sequenced TPBVP can be formulated according to (12). As these continuous sub-problems are decoupled from one another, they can be solved in a decentralized manner, i.e. by each agent computing its own optimal trajectory. This method is denoted decentralized planning.
Based on the decentralized individual solutions a set of most promising global plans is determined. For those trajectories, a cooperative strategy is approximated, where all agents that have not yet reached TA already move towards a future docking configuration as illustrated in scheme Fig. 1. Therefore, intermediate boundary values are introduced for agents $\{A_i\}, \;\; \forall i \in \mathcal{M}_A(t)$. This technique is referred to cooperated planning. By contrast, cooperative transportation plans computed by full hierarchical optimization are related to the brute-force planning approach.

4. NUMERICAL RESULTS

In the following, (sub-) optimal solutions obtained from applying the consensus based decoupling method are discussed in terms of their performance and the adjustable parameters in the consensus protocol. The scalability of this technique is presented in comparison to brute-force enumerative planning.

4.1 Approximation of Best-Performing Plans and the Influence of the Consensus Parameters

In the examined four-vehicle setup, the fixed initial conditions are $\boldsymbol{\xi}_A^0 = (10, 9, 0, 5)^T, \boldsymbol{\eta}_A^0 = (10, 0, 7, 10)^T, \boldsymbol{\Theta}_A^0 = (-\frac{\pi}{2}, -\pi, \frac{\pi}{2}, 0)^T$, and $\boldsymbol{x}_{TA}^0 = (7, 8, \pi)^T, \boldsymbol{x}_{TA}^e = (0, 0, \pi)^T$.
In Tab. 1 the effects of a varying information topology in approximating well-performing docking positions are listed. In the left half of the table the best performing docking sequences are given for the decentralized planning approach. The performance of the approximated solutions increases until an information topology involving three agents in addition to the TA is used for docking position approximation. The lowest cost index is obtained for the docking sequence A_4-A_1-A_2-A_3. Cooperative task execution for this best plan is approximated and depicted in Fig. 3. For comparison, the costs of the same docking schedule computed in the brute-force manner results in a costs index of $J = 68.3$. For $r = 4$ the performance slightly decreases, but when examining the three best performing schedules with the cooperated planning method the best plan is represented by the schedule A_4-A_2-A_1-A_3, accompanied by a performance increase of almost 30 percent.

For this latter planning approach and the globally best plan, the effect of a varying receding horizon is examined again, see the right half of Tab. 1. Here, the increase in cost efficient execution of the transportation plan with increased r becomes more evident. Comparing the cost

Table 1. Varying information topology and best approximated solution plans. Left: distributed planning; Right: cooperated planning; $Nv = 4$, $f(b)$-gain

r	Sequence	Cost	Sequence	Cost
1	A_1-A_4-A_2-A_3	97.5	A_4-A_2-A_1-A_3	88.9
2	A_4-A_2-A_1-A_3	95.2	A_4-A_2-A_1-A_3	86.3
3	A_4-A_1-A_2-A_3	91.3	A_4-A_2-A_1-A_3	75.0
4	A_4-A_2-A_1-A_3	92.3	A_4-A_2-A_1-A_3	65.7

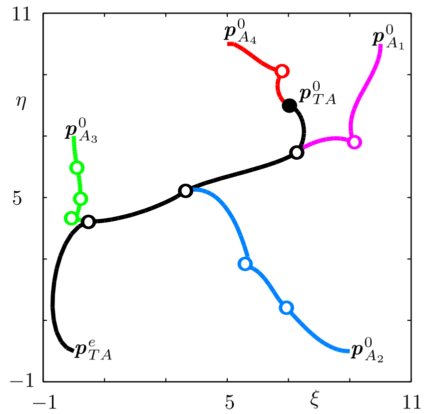

Fig. 3. Cooperative transportation plan approximated from the best performing solution resulting from decentralized planning. The docking sequence is A_4-A_1-A_2-A_3 with the transportation cost $J = 76.7$. The white filled circles represent intermediate boundaries.

index stemming from the most and least expensive docking position approximations, again an overall performance increase of almost one third can be observed. In Tab. 2 the influence of the controller gain on the rendez-vous point approximation is demonstrated for three cases. In the unit gain scenario, all control inputs in the feedback controller are weighted equally, and thus, besides the receding horizon, the docking position approximation depends only on the type of kinematics, i.e. the unicycle type. In contrast to that, $f(b)$-related gains consider by weighting the TA-inputs with the factor $f(b(t))$ the switching TA dynamics in case of an agent docking to it, see (6). For the third case, the TA controls are weighted with $f(b(t))^2$; hence, the resulting docking approximation is related to the objective function (11), where the square of the control inputs is integrated over time. Herein, the most cost efficient decoupling ensues from using the $f(b)^2$-gain in the consensus protocol. Moreover, for each examined docking plan, a performance improvement is achieved by applying the gain strategies in the order 'unit-$f(b)$-$f(b)^2$'.

In reference to the decoupling approaches presented in Mangesius et al. (2010), it is to note that the feedback consensus decoupling approach applied to the three-agent scenario leads to the same best transportation plan.

Table 2. Controller gain and costs of best performing global plans for $N_A = 4$, $r = 4$ and coordinated planning

Sequence	unit-gain	$f(b)$-gain	$f(b)^2$-gain
A_1-A_4-A_2-A_3	79.4	74.2	73.6
A_4-A_1-A_2-A_3	76.7	72.2	71.6
A_4-A_2-A_1-A_3	72.8	65.7	64.0

4.2 Scalability

In Fig. 4 the computational effort of the three methods decentralized, cooperated, and brute-force planning is shown. It is to note that the combinatorial complexity of enumeration on the upper level is not considered in this illustration, but only the effort to find the optimal solution for one discrete state schedule. The solution process of the decentralized approach scales linearly with an increasing number of agents. The effort in cooperated planning strongly increases, whereas it remains in the scale of the decentralized approach. Therefore, the combination of decentralized planning and reduction to a small set of well performing docking sequences upon which cooperative planning is applied is reasonable. The brute-force approach quickly becomes intractable already for a small number of agents. Due to that, the computational effort of this approach is plotted as logarithmic value. The reason for such expensive computation is the exponential increase of the computational time when adopting hierarchical optimization: For each variation of a variable on the middle layer a complete sequence of TPBVP has to be re-optimized. In that sense, the decoupling approach by approximating the docking positions avoids the effect of exponential increase of effort. Exploring the solution

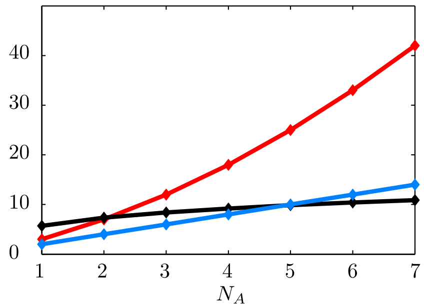

Fig. 4. Computational effort to solve the overall transportation planning problem normed by the effort to solve one TPBVP optimally; red - cooperated, blue - decentralized and black - log-scaled brute-force scheme

space of a three-agent scenario, and approximating a cooperative plan that corresponds to the best discrete sequence requires approximately 7 minutes. The same procedure for the four-agent setup lasts about 36 minutes, whereas in the later case the solution space comprises 24 discrete state sequences to examine (in contrast to 6 for the three-agent case).

5. IMPLICATIONS AND FUTURE OUTLOOK

In this new type of multi-agent problem, a relation between optimal discrete and continuous dynamics is established via autonomous jump sets and optimal switching configurations therein, see (13). The approximation of switching configurations ensues without explicitly computing values of encapsulated objective functions; however, essentially invariant features of the global minimization of the value function (11) are reflected in the near-optimal switching position approximation. These features are the agent kinematics and dynamics (5) and (6) as well as the transition graph $\mathcal{G}$, which is related to the information graph within the receding horizon. This scheme differs from standard (static) heuristics, such as purely geometric measures, by being implemented in a dynamics setting reaching agreements between the inner two hierarchical levels, see Fig. 2. Hence, a dynamics related approximation is achieved by merging of two layers. In contrast to optimization based approximations, see Mangesius et al. (2010), near-optimal switching configurations can be computed by simple time integration.

Numerical simulation results show gradual performance increase by gradually adding structural information into the consensus controller. This clearly indicates that a Pareto-optimal solution of the collective dynamics has a special invariant structure, and thus it has to lie in a constraint solution set. Furthermore, the design of a computation scheme for real-time application has to include balance between exploration of the continuous solution space and exploitation of the discrete solution space in one functional layer; Hence, future research aims at incorporating the top-level supervisor in a hybrid consensus scheme, that produces a near-optimal agreement between discrete state scheduling and locally optimal trajectories in a distributed fashion.

REFERENCES

Branicky, M., Vivek, B., and Mitter, S. (1998). A unified framework for hybrid control: Model and optimal control theory. *IEEE Trans. Automat. Contr.*, 43(1), 31–45.

Dimarogonas, D. and Kyriakopoulos, K. (2007). On the rendezvous problem for multiple nonholonomic agents. *IEEE Trans. on Autom. Contr.*, 52(5), 916–922.

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)*.

Listmann, K., Msalawala, V., and Adamy, J. (2009). Consensus for formation control of nonholonomic mobile robots. In *IEEE International Conference on Robotics and Automation*.

Mangesius, H., Sobotka, M., and Stursberg, O. (2010). Solution of a multi-agent transport problem by hybrid optimization. In *10th International Workshop on Discrete Event Systems*.

Murray, R. (2007). Recent research in cooperative control of multi-vehicle systems. *ASME Journal of Dynamic Systems, Measurement and Control*, 129(5), 571–583.

Olfati-Saber, R. (2007). Evolutionary dynamics of behavior in social networks. In *46th IEEE Conference on Decision and Control*.

Parker, L.E. (1993). Designing control laws for cooperative agent teams. In *Proceedings of the IEEE International Conference on Robotics and Automation*.

Passenberg, B., Sobotka, M., Stursberg, O., Buss, M., and Caines, P. (2010). An algorithm for discrete state sequence and trajectory optimization for hybrid systems with partitioned state space. In *49th IEEE Conference on Decision and Control*.

Shaikh, M. and Caines, P. (2007). On the hybrid optimal control problem: Theory and algorithms. *IEEE Trans. Automat. Contr.*, 57(6), 1587–1603.