Proceedings of the 17th International Symposium on Mathematical
Theory of Networks and Systems, Kyoto, Japan, July 24-28, 2006

Distributed Model Predictive Consensus

Björn Johansson, Alberto Speranzon, Mikael Johansson, Karl Henrik Johansson

***Abstract*— In this paper we consider the problem of designing a distributed control strategy such that a linear combination of the states of a number of vehicles coincide at a given time. The vehicles are described by linear difference equations and are subject to convex input constraints. It is demonstrated how primal decomposition techniques and incremental subgradient methods allow us to find a solution in which each vehicle performs individual planning of its trajectory and exchanges critical information with neighbors only. We explore various communication, computation, and control structures, and demonstrate the performance of the algorithms by numerical examples.**

I. INTRODUCTION

The problem of cooperatively controlling a system comprising a large number of autonomous vehicles has attracted substantial attention in the control and robotics communities. An emerging problem is the so called consensus problem, see for example [8], [9], [5], [11] and the references therein. It consists of designing distributed control strategies such that the output of the vehicles asymptotically converges to a common value. Typically the vehicles are modelled by identical first-order systems and they communicate over a fixed or time-varying communication network [8], [9], [5], [3]. In this paper we study the consensus problem in a different setting. In particular, we assume that the vehicles are described by general linear dynamics, possibly different for each vehicle, that the inputs are constrained and that a linear combination of the states needs to converge to a common value, the consensus point, after a fixed time. The consensus point is *not* specified in advance, but it is negotiated by the vehicles so that a cost index is optimized.

In order to accommodate all the constraints a distributed model predictive control (MPC) strategy is used to design the controller and to determine an optimal consensus point. In distributed MPC, a static finite-horizon optimization problem is decomposed into a set of subproblems, each solved by an individual agent. The coordination of the subproblems is, generally, achieved by an active communication, or sensing, among the agents [4].

Distributed MPC for coordinating swarms of mobile vehicles was recently proposed in the literature. Distributed MPC strategies for steering vehicles to a stable formation are studied in [6], [7], [2]. In [6], [7] the authors propose a scheme where the equilibrium is given a priori, while in [2] the models and constraints of the other agents are needed in order for each agent to solve its optimal control problem. In this paper we consider less restrictive setups.

The main contribution of this paper is to propose a decentralized control strategy that yields consensus in fixed time. In this solution, vehicles can be described by arbitrary linear difference equations and be subject to convex input constraints, while only suggested consensus values need to be communicated between vehicles. Contrary to related proposals, the vehicles do not need any model of the dynamics of its team mates, nor exchange complete planned trajectories. We explore various communication, computation, and control structures, and demonstrate the performance of the algorithms by numerical examples.

The paper is organized as follows. In Section II we formally define the vehicle models and the distributed model predictive control problem. In Section III it is shown how primal decomposition techniques and incremental subgradient methods allow us to find a distributed solution to this consensus problem, in which each vehicle performs individual planning of its trajectory and exchanges critical information with neighbors only. We explore different computation and control structures needed in order to cope with disturbances and changes in the system configuration in Section IV. The performance of these different structures are demonstrated by numerical examples in Section V. Finally, Section VI concludes the paper.

II. PROBLEM FORMULATION

Consider $N > 1$ vehicles whose dynamics are described by the following discrete time state equations

$$\begin{aligned} x_i(t+1) &= A_i x_i(t) + B_i u_i(t) \\ y_i(t) &= C_i x_i(t)\,, \qquad\qquad i = 1,\dots,N\,, \end{aligned}$$

where $A_i \in \mathbb{R}^{n_i \times n_i}$, $B_i \in \mathbb{R}^{n_i \times p_i}$ and $C_i \in \mathbb{R}^{s_i \times n_i}$. We assume that the inputs are constrained according to

$$u_i(t) \in \mathcal{U}_i := \{v : \underline{u}_i \le v \le \overline{u}_i\}, \tag{1}$$

where $\underline{u}_i, \overline{u}_i \in \mathbb{R}^{p_i}$ and the inequalities are elementwise.

Let $T > 0$ be a finite and fixed time. We want to find a sequence of inputs $u_i(0), \dots, u_i(T-1)$, with $i = 1, \dots, N$ and $u_i(t) \in \mathcal{U}_i$ for all $t = 0, \dots, T-1$, such that

$$y_i(T) = \theta, \tag{2}$$

where $\theta \in \Theta$ is the consensus point and Θ is a given convex and compact set. Namely we are seeking a control sequence and a consensus point so that in fixed time we reach such a consensus point, meaning that all the outputs are equal at time T.

All the authors are with the Automatic Control Lab, School of Electrical Engineering, Royal Institute of Technology (KTH), 100 44 Stockholm, Sweden. Email: {bjorn.johansson, albspe, mikaelj, kallej}@ee.kth.se.

In order for (2) to hold, the outputs need to have the same dimensions, i.e., we require that $s_i = s$ for all i. We assume that the following cost function is associated to the i-th system,

$$
\begin{aligned}
V_i(y_i(t), u_i(t), \theta) = (y_i(t) - \theta)^\intercal\, Q_i\, (y_i(t) - \theta) \\
+\, u_i(t)^\intercal R_i u_i(t)
\end{aligned} \tag{3}
$$

where $Q_i \in \mathbb{R}^{s \times s}$ and $R_i \in \mathbb{R}^{p_i \times p_i}$ are positive definite symmetric matrices (i.e., we penalize deviations from the consensus point and the use of control effort). We then formulate the following control problem.

Problem: Let $T > 0$ be fixed. Determine the control vectors $u_i(t)$, $t = 0, \ldots, T-1$, for all $i = 1, \ldots, N$ and the consensus point $\theta \in \Theta$, that solve the following optimization problem

$$
\begin{aligned}
\underset{u,x,y,\theta}{\text{minimize}} \quad & \sum_{i=1}^{N}\sum_{t=1}^{T} V_i(y_i(t), u_i(t), \theta) \\
\text{s.t.} \quad & x_i(t+1) = A_i x_i(t) + B_i u_i(t) \\
& y_i(t) = C_i x_i(t) \\
& y_i(T) = \theta \\
& x_i(0) = x_i^0 \\
& \theta \in \Theta \\
& u_i(t) \in \mathcal{U}_i
\end{aligned} \tag{4}
$$

In order to make the problem well posed the following assumptions need to be satisfied. First, the dynamical systems are assumed to be controllable and observable. Second, the meeting time T is large enough so that all θ in the set Θ is feasible, i.e., all θ in the set Θ are possible consensus points. Third, for all $\theta \in \Theta$ and $i = 1, ..., N$, there exists a sequence $u_i(0), ..., u_i(T-1)$ in the relative interior of $\mathcal{U}_i$ such that $y_i(T) = \theta$. This condition means that it should be possible to reach θ without saturating the control signal. Note that the optimal control signal may still be saturated.

If we introduce

$$
\mathbf{x}_i = \begin{pmatrix} x_i(1) \\ x_i(2) \\ \vdots \\ x_i(T) \end{pmatrix}, \quad \mathbf{u}_i = \begin{pmatrix} u_i(0) \\ u_i(1) \\ \vdots \\ u_i(T-1) \end{pmatrix}
$$

then we can rewrite the optimization problem (4) as follows

$$
\begin{aligned}
\underset{\mathbf{u_1}, \ldots, \mathbf{u_N}, \theta}{\text{minimize}} \quad & \sum_{i=1}^{N} \mathbf{V}_i(\mathbf{u}_i, \theta) \\
\text{s.t.} \quad & \mathbf{H}_i(\mathbf{E}_i x_i^0 + \mathbf{F}_i \mathbf{u}_i) = \theta \\
& \theta \in \Theta \\
& \mathbf{u}_i \in \mathcal{U}_i^T,
\end{aligned} \tag{5}
$$

where $\mathcal{U}_i^T = \prod_{k=1}^{T} \mathcal{U}_i$ and the constraints $y_i(T) = \theta$ have been rewritten as

$$
\begin{aligned}
y_i(T) = C_i x_i(T) &= \underbrace{\begin{pmatrix} 0 & \ldots & 0 & C_i \end{pmatrix}}_{\mathbf{H}_i} \mathbf{x}_i \\
&= \mathbf{H}_i(\mathbf{E}_i x_i^0 + \mathbf{F}_i \mathbf{u}_i) = \theta
\end{aligned}
$$

using the fact that

$$
\mathbf{x}_i = \underbrace{\begin{pmatrix} A_i \\ A_i^2 \\ \vdots \\ A_i^T \end{pmatrix}}_{\mathbf{E}_i} x_i^0 + \underbrace{\begin{pmatrix} B_i & 0 & \ldots & 0 \\ A_i B_i & B_i & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A_i^{T-1} B_i & A_i^{T-2} B_i & \ldots & B_i \end{pmatrix}}_{\mathbf{F}_i} \mathbf{u}_i \,.
$$

The cost function of (5) is given by

$$
\begin{aligned}
\mathbf{V}_i(\mathbf{u}_i, \theta) = \sum_{t=1}^{T} V_i(y_i(t), u(t), \theta) = (\mathbf{C}_i(\mathbf{E}_i x_i^0 \\
+ \mathbf{F}_i \mathbf{u}_i) - \mathbf{1}_T \otimes \theta)^\intercal \mathbf{Q}_i (\mathbf{C}_i(\mathbf{E}_i x_i^0 \\
+ \mathbf{F}_i \mathbf{u}_i) - \mathbf{1}_T \otimes \theta) + \mathbf{u}_i^\intercal \mathbf{R}_i \mathbf{u}_i
\end{aligned}
$$

where $\mathbf{1}_T$ is the T-dimensional unity vector $(1, 1, \ldots, 1)^\intercal$, $\mathbf{Q}_i = \mathbf{I}_T \otimes Q_i$, $\mathbf{R}_i = \mathbf{I}_T \otimes R_i$, and $\mathbf{C}_i = \mathbf{I}_T \otimes C_i$ where $\mathbf{I}_T$ is the T-dimensional identity matrix and $\otimes$ is the Kronecker matrix product.

When dealing with the coordination of a set of mobile vehicles, the optimization problem (5) becomes interesting if the computations can be distributed among the vehicles and the amount of information that the vehicles need to exchange is limited. The problem is even more interesting if each vehicle is constrained to communicate only with a few neighboring vehicles. In the following we develop a coordination algorithm in which vehicles only need to exchange their current values of θ, but still the algorithm leads to that the vehicles' outputs converge to the optimal consensus point.

III. Distributed Negotiation

In this section we show how the optimal consensus point, θ, can be computed in a distributed way. The key idea is to transform (5) using primal decomposition in combination with incremental subgradient methods (*e.g.*, [1]).

Since $\mathbf{H}_i(\mathbf{E}_i x_i^0 + \mathbf{F}_i \mathbf{u}_i) = \theta$, we can eliminate the dependence from θ in $\mathbf{V}_i(\mathbf{u}_i, \theta)$. Thus we have

$$
\begin{aligned}
\mathbf{V}_i(\mathbf{u}_i) = \Big(\mathbf{C}_i(\mathbf{E}_i x_i^0 + \mathbf{F}_i \mathbf{u}_i) - \mathbf{1}_T \otimes \big(\underbrace{\mathbf{H}_i(\mathbf{E}_i x_i^0}_{\theta} \\
+ \mathbf{F}_i \mathbf{u}_i)\big)\Big)^\intercal \mathbf{Q}_i \Big(\mathbf{C}_i(\mathbf{E}_i x_i^0 + \mathbf{F}_i \mathbf{u}_i) \\
- \mathbf{1}_T \otimes \big(\underbrace{\mathbf{H}_i(\mathbf{E}_i x_i^0 + \mathbf{F}_i \mathbf{u}_i)}_{\theta}\big)\Big) + \mathbf{u}_i^\intercal \mathbf{R}_i \mathbf{u}_i \,.
\end{aligned}
$$

We can then define $q_i(\theta)$ as follows

$$
\begin{aligned}
q_i(\theta) = \underset{\mathbf{u}_i}{\text{minimum}} \quad & \mathbf{V}_i(\mathbf{u}_i) \\
\text{s.t.} \quad & \mathbf{H}_i(\mathbf{E}_i x_i^0 + \mathbf{F}_i \mathbf{u}_i) = \theta \\
& \mathbf{u}_i \in \mathcal{U}_i^T \,.
\end{aligned} \tag{6}
$$

Then the optimization problem (5) can be written as

$$\begin{aligned} \underset{\theta}{\text{minimize}} \quad & \sum_{i=1}^{N} q_i(\theta) \\ \text{s.t.} \quad & \theta \in \Theta\,, \end{aligned} \tag{7}$$

Solution to an optimization problem over u_i

since the only coupling between the vehicles is θ. We will later use subgradients to find the consensus point and therefore we give the following definition.

Definition 3.1: ([1]) Let $f : \mathbb{R}^n \to \mathbb{R}$ be a convex function. We say that a vector $\lambda \in \mathbb{R}^n$ is a subgradient of f at point $x \in \mathbb{R}^n$ if

$$f(z) \geq f(x) + \lambda^{\mathsf{T}}(z - x)$$

for all $z \in \mathbb{R}^n$.

Proposition 3.2: The cost function $q_i(\theta)$ defined in (6) is a convex function and a subgradient λ_i is given by the Lagrange multipliers corresponding to the constraint $\mathbf{H}_i(\mathbf{E}_i x_i^0 + \mathbf{F}_i \mathbf{u}_i) = \theta$.

Proof: We start by showing that a subgradient is given by the Lagrange multipliers corresponding to the constraint $\mathbf{H}_i(\mathbf{E}_i x_i^0 + \mathbf{F}_i \mathbf{u}_i) = \theta$. By Lagrangian relaxation we can define

$$L(\mathbf{u}_i, \theta, \lambda_i) = \mathbf{V}_i(\mathbf{u}_i) - \lambda_i^{\mathsf{T}}(\mathbf{H}_i(\mathbf{E}_i x_i^0 + \mathbf{F}_i \mathbf{u}_i) - \theta)\,,$$

where λ_i are Lagrange multipliers. We also introduce the dual function

$$d(\lambda_i, \theta) = \min_{\mathbf{u}_i \in \mathcal{U}_i^T} \mathbf{V}_i(\mathbf{u}_i) - \lambda_i^{\mathsf{T}}(\mathbf{H}_i(\mathbf{E}_i x_i^0 + \mathbf{F}_i \mathbf{u}_i) - \theta)\,.$$

Since the constraint $\mathbf{H}_i(\mathbf{E}_i x_i^0 + \mathbf{F}_i \mathbf{u}_i) = \theta$ is linear in $\mathbf{u}_i$ and there exist a solution to this equation (by assumption) within the relative interior of $\mathcal{U}_i^T$ and the function $\mathbf{V}_i$ is convex and the set $\mathcal{U}_i^T$ is convex, strong duality follows from Theorem 6.4.4 (p. 373) in [1]. Now $q_i(\theta)$ can be expressed as

$$q_i(\theta) = \max_{\lambda_i} d(\lambda_i, \theta)\,.$$

Consider two feasible points, $\theta^\dagger$ and $\theta^\ddagger$, and let $\lambda_i^\dagger$ be the Lagrange multipliers corresponding to the relaxed constraint for $\theta^\dagger$, then

$$\begin{aligned} q_i(\theta^\ddagger) &= \max_{\lambda_i} \Big\{ \min_{\mathbf{u}_i \in \mathcal{U}_i^T} \{\mathbf{V}_i(\mathbf{u}_i) - \lambda_i^{\mathsf{T}}(\mathbf{H}_i(\mathbf{E}_i x_i^0 \\ &\qquad + \mathbf{F}_i \mathbf{u}_i) - \theta^\ddagger)\} \Big\} \\ &\geq \min_{\mathbf{u}_i \in \mathcal{U}_i^T} \Big\{ \mathbf{V}_i(\mathbf{u}_i) - (\lambda_i^\dagger)^{\mathsf{T}}(\mathbf{H}_i(\mathbf{E}_i x_i^0 \\ &\qquad + \mathbf{F}_i \mathbf{u}_i) - \theta^\ddagger) \Big\} \\ &= \min_{\mathbf{u}_i \in \mathcal{U}_i^T} \Big\{ \mathbf{V}_i(\mathbf{u}_i) - (\lambda_i^\dagger)^{\mathsf{T}}(\mathbf{H}_i(\mathbf{E}_i x_i^0 \\ &\qquad + \mathbf{F}_i \mathbf{u}_i) - \theta^\dagger) \Big\} + (\lambda_i^\dagger)^{\mathsf{T}}(\theta^\ddagger - \theta^\dagger) \\ &= q(\theta^\dagger) + (\lambda_i^\dagger)^{\mathsf{T}}(\theta^\ddagger - \theta^\dagger) \end{aligned}$$

Hence, by the definition of a subgradient, $\lambda_i^\dagger$ is a subgradient of $q_i(\cdot)$ at $\theta^\dagger$. Now $q_i(\theta^\ddagger)$ can be expressed as

$$\begin{aligned} q_i(\theta^\ddagger) &= \max_{\lambda_I} \Big\{ \min_{\mathbf{u}_i \in \mathcal{U}_i^T} \{\mathbf{V}_i(\mathbf{u}_i) - \lambda_i^{\mathsf{T}} \mathbf{H}_i(\mathbf{E}_i x_i^0 \\ &\qquad + \mathbf{F}_i \mathbf{u}_i)\} + \lambda_i^{\mathsf{T}} \theta^\ddagger \Big\} \\ &= \max_{\lambda} \{ g(\lambda_i) + \lambda_i^{\mathsf{T}} \theta^\ddagger \} \end{aligned}$$

where $g(\lambda_i)$ is some function and $g(\lambda_i) + \lambda_i^{\mathsf{T}} \theta^\ddagger$ is convex in $\theta^\ddagger$. Since $q_i(\theta^\ddagger)$ is the pointwise maximum of a family of convex functions, $q_i(\theta^\ddagger)$ is convex. ■

To find the consensus point θ, we next use incremental and randomized subgradient methods. Proposition 3.2 provides the subgradients corresponding to $q_i(\theta)$.

A. Incremental Subgradient Algorithms

We present in the following two algorithms that compute the optimal consensus point, θ, using two different communication strategies. These algorithms are based on the incremental subgradient methods from optimization theory [1].

Subgradient methods work in a way that is similar to gradient methods, i.e., the update is made in the opposite direction of the subgradient (for minimization). The update equation is

$$\theta_{k+1} = \mathcal{P}_\Theta\{\theta_k - \alpha_k \lambda_k\} \tag{8}$$

where $\mathcal{P}_\Theta\{\cdot\}$ denotes the Euclidean projection on the set Θ and α_k is the stepsize. The subgradient at time step k, λ_k, is in the standard approach computed for the total cost $\sum_i q_i(\theta)$. In this paper we describe two algorithms where the generic vehicle i computes the subgradient corresponding to the function q_i at time step k, $\lambda_{i,k}$, based on the information received by the neighbor vehicles.

In the first algorithm we assume that the generic vehicle i communicates with vehicle $(i+1) \bmod N$, that is to say the neighbor of vehicle $(i+1) \bmod N$ is vehicle i. Starting with an arbitrary initial condition $\theta_0 \in \Theta$, the first vehicle computes the subgradient $\lambda_{1,0}$ corresponding to $q_1(\theta_0)$. Using (8) an update of the consensus point θ_1 is computed and communicated to the next vehicle. This vehicle then computes in the same way the next update of the consensus point, θ_2. The algorithm then proceeds iteratively. A pseudocode version of the algorithm is summarized in Algorithm 1.

In the second algorithm, at each time step k the vehicle that has performed the last update of the consensus point, according to (8), randomly selects another vehicle among all the available vehicles, and sends the update to the selected vehicle. The advantage of the method is that there is no need for a particular communication structure, even if at each time step every vehicle must be able to communicate with any other vehicle. The algorithm is summarized in Algorithm 2.

We then have the following proposition.

Algorithm 1 Cyclic Incremental Algorithm

Initialize θ_0 and α_0
$k := 0$
loop
  for $i := 1$ to N **do**
    Compute a subgradient, $\lambda_{i,k}$, corresponding to $q_i(\theta_k)$
    $\theta_{k+1} := \mathcal{P}_\Theta\{\theta_k - \alpha_k \lambda_{i,k}\}$
    $k := k + 1$
    $\alpha_k := \alpha_0/k$
  end for
end loop

Algorithm 2 Randomized Algorithm

Initialize θ_0 and α_0
$k := 0$
loop
  Choose $i \in \{1, ..., N\}$ accordingly to uniform probability mass function
  Compute a subgradient, $\lambda_{i,k}$, corresponding to $q_i(\theta_k)$
  $\theta_{k+1} := \mathcal{P}_\Theta\{\theta_k - \alpha_k \lambda_{i,k}\}$
  $k := k + 1$
  $\alpha_k := \alpha_0/k$ varying step size
end loop

Proposition 3.3: Algorithms 1 and 2 converge in the sense that $\lim_{k\to\infty} \theta_k = \theta^*$, where θ^* is the solution to the optimization problem (7).
The proof follows from Theorem 8.2.6 (p. 480) and Theorem 8.2.13 (p. 496) in [1], since the set Θ is convex and compact (so the norms of all possible subgradients have an upper bound), and the stepsize a_k is square summable over k but not summable over k.

IV. Implementation

In this section we present three classes of control strategies that address the consensus problem. Figure 1 show the logical flow of these classes, cf., [10, p. 386]. The distributed MPC strategies discussed previously belong to first class, while the second and third class are further improvements to address various implementation aspects.

The main assumption we make here is that the most costly phase in the consensus algorithms is the motion of the vehicles to the consensus point. We thus consider the communication among the vehicles to be cheap. Such assumption is plausible when the vehicles are, for example, large size autonomous robots or unmanned aerial vehicles.

The logical flow of the first scheme is summarized in Figure 1(a). In the negotiation phase, the optimal consensus point is computed in a distributed way using Algorithm 1 or Algorithm 2. After the distributed negotiation, the corresponding control action is applied to the vehicle in open loop during the execution phase. If there are no disturbances the system will reach the consensus point at time T. The main advantage of the scheme proposed is that it is possible to formally guarantee that the optimal consensus point is computed in a distributed way. Moreover only a small amount of information, the current consensus point, needs to be exchanged at each step. Such strategy, being completely open loop, is very sensitive to disturbances. Another disadvantage is that it requires the negotiation phase to converge to the optimum consensus point before any control action can be applied, or in more practical terms it would require a long time before the consensus point is relatively close the optimum.

In Figure 1(b), a second control strategy is proposed. In this case, as the previous strategy, the negotiation phase yields a consensus point that is optimal in the absence of disturbances. The controller that drives the vehicles towards the consensus during the execution phase uses the negotiated consensus point as fixed reference. Each vehicle can then use a receding horizon (MPC) control strategy for reaching the consensus point: in each step we recompute the optimal control sequence for reaching the consensus point at time T, apply the first component, sense the current state and recompute the control sequence.

The third control strategy is shown in Figure 1(c). The negotiation phase, in this case, is carried out at each time step and we assume that the negotiation is stopped after that all the vehicles have communicated only β times, namely we assume that the negotiation is interrupted at $k = \beta N$. In this case we then have N different reference signals, one for each vehicle. Similarly to above, in the execution the optimal control sequence for reaching the consensus point is computed and the first component is applied. The negotiation phase is then repeated.

Similarly to classical MPC, the control strategies in Figure 1(b) and Figure 1(c) can cope with disturbances due to the receding horizon operation. The main advantage of the strategy in Figure 1(c) is that the negotiation is not carried out to the optimum and thus we do not need to wait for the incremental subgradient algorithms to converge. This solution can also cope with changes in the vehicle dynamics and/or input constraints. Indeed the vehicle affected can include these changes locally in its optimization algorithm. As we will see later, the strategy can also handle the situation when the number of vehicles in the consensus problem increases or decreases. In this case the new vehicle can be included in the negotiation and the consensus point can be recomputed.

Compared to the open-loop solution of Figure 1(a), the control strategies of Figure 1(b) and Figure 1(c) are much harder to analyze formally and an in-depth theoretical investigation of these approaches are outside the scope of this paper. Still, since they are relevant from a practical perspective, we will demonstrate the potential of such strategies via simulations.

V. Numerical Examples

In this section we explore the performance of the three control strategies through numerical examples. The setup is

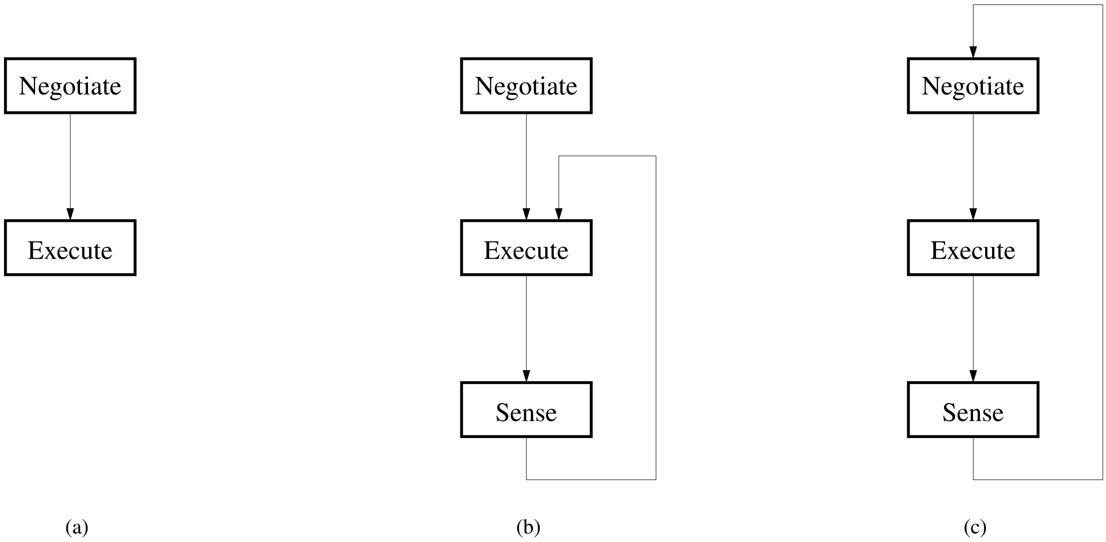


Fig. 1. The logical flow of the different control strategies. (a) open loop strategy: θ is negotiated once at the beginning, then the corresponding control action is applied. (b) set-point strategy: θ is negotiated once at the beginning and the corresponding control action is then computed at every time step and applied to the system. (c) renegotiation strategy: θ is renegotiated at every time step. The corresponding control action is then computed and applied to the system.

that a number of vehicles with double integrator dynamics and input constraints should reach the same coordinates at time T.

A. Disturbance Free Scenario

The first case is the ideal case, where we assume that there is no noise, the number of vehicles is constant, and the dynamics of the vehicles do not change. There is no need for feedback and the optimal consensus point is negotiated in the beginning, and then the corresponding control actions are applied to the systems in open-loop. In this specific example there are four vehicles with double integrator dynamics, identical control signal constraints, but different initial positions and velocities. The vehicles should meet after 20 time samples. The trajectories are shown in Figure 2(a). As we expect the vehicles meet after 20 samples. With the same setup we also introduce the extension that the vehicles can meet in a formation. This is done by adding an individual bias to the consensus point. The configuration is identical with the ideal case except that the system now should meet in a square formation. As can be seen in Figure 2(b), the vehicles meet in a square formation after 20 samples.

B. Noisy Scenario

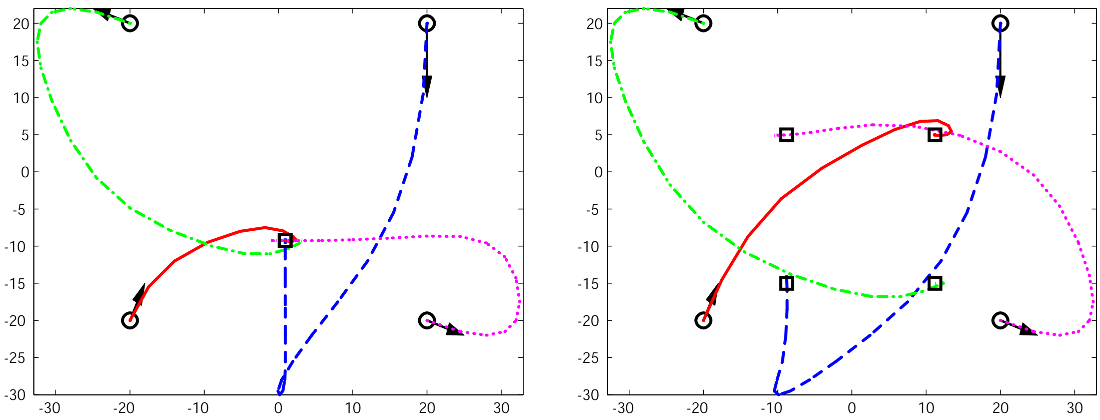
In the second case, we add Gaussian noise with standard deviation 0.5. Two variants are compared: the open-loop variant (Figure 1(a)) and the setpoint variant (Figure 1(b)). In the setpoint variant, the consensus is negotiated before the vehicles start moving, and then the consensus is used as a setpoint. The control signals are recalculated at every time step, yielding a closed loop control. The trajectories of the open loop variant are shown in Figure 3(a), and as expected the vehicles do not reach consensus. Figure 3(b) demonstrates the results for the setpoint variant: the vehicles are very close to achieving consensus in 20 samples despite the persistent disturbances.

C. Scalability Scenario

The third case starts with three vehicles and adds a fourth vehicle after 10 samples. The total time of the simulation is 30 samples. Also in this scenario, two variants are compared: the setpoint variant discussed above and the renegotiation variant (Figure 1(c)). In the setpoint variant the consensus point is negotiated between the three vehicles in the beginning. When the fourth vehicle is added, it is given θ as a setpoint. Figure 4(a) shows the trajectories of the setpoint variant. The vehicles reach consensus as expected but the initial condition of the fourth system does not influence the consensus point at all, irrespectively of how hard it is to control and how expensive its control efforts are. In the renegotiation algorithm, the cyclic algorithm is executed with 10 iterations at each step. The trajectories are shown in Figure 4(b), and as can be seen the final consensus is closer to the added vehicle in this case compared with the previous algorithm. Another advantage is that the vehicles can start moving before the consensus point is completely agreed upon. However, this is also a drawback since if the current θ is far from optimal, then the vehicles can start moving in the wrong direction. The behavior depends on the setup, e.g., the dynamics and initial conditions, and warrants further theoretical investigations.

VI. Conclusions

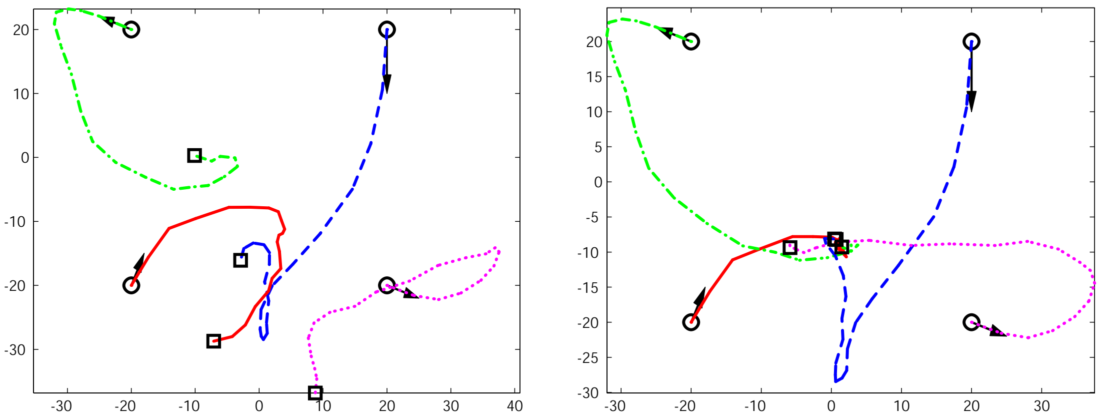
We have formulated a consensus problem where the output of a number of different vehicles should coincide after a specified time. The dynamics of the vehicles can



(a) The vehicles meet at θ at time T

(b) The vehicles meet at a formation, parameterized as the consensus point θ + individual offsets, at time T

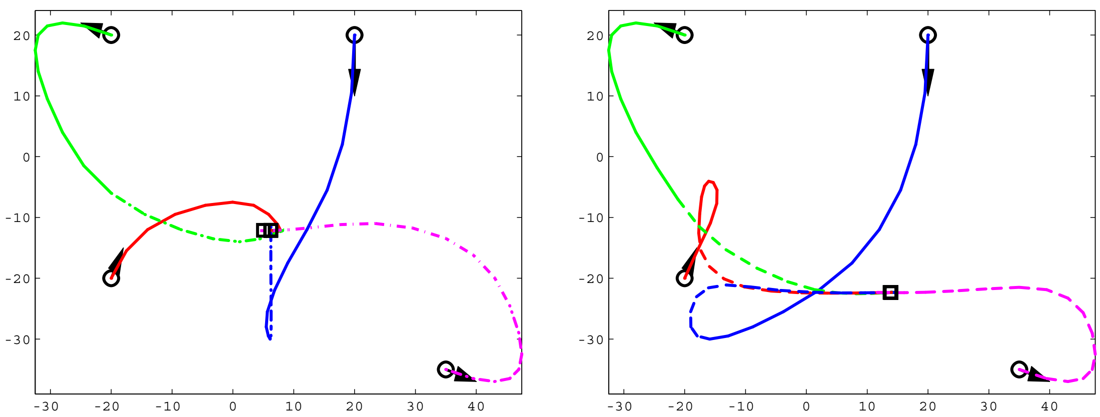
Fig. 2. The trajectories of four vehicles with double integrator dynamics. The circles are the starting points and the squares are the ending points. The arrows show the initial velocities.



(a) Open loop.

(b) Closed loop with θ as setpoint.

Fig. 3. The trajectories of four vehicles with double integrator dynamics and noise added to the states at each sample. The circles are the starting points and the squares are the ending points. The arrows show the initial velocities.



(a) Closed loop with θ negotiated once, then θ is the setpoint for all systems.

(b) θ is renegotiated at every time step.

Fig. 4. The trajectories of four vehicles with double integrator dynamics. The system starts with three vehicles and after 10 samples a fourth vehicle is added to the system. The circles are the starting points and the squares are the ending points. The solid lines denote the trajectories before the fourth system is added, and the dashed lines denote the trajectories after that the fourth system has been added. Finally, the arrows show the initial velocities.

be arbitrary as long as it is linear and input constraints are convex. We have shown that it is possible to find the consensus point in a distributed way where the only information needed to be communicated is the current consensus point suggestion. Moreover, we have proposed three different control schemes suited for more realistic scenarios and we have explored the performance by numerical simulation. The control schemes are very flexible and can handle several difficulties in multi-vehicle coordination. The drawback is that the general behavior is difficult to analyze: we have only been able to analyze the simplest of these schemes. Natural extensions include investigating the convergence properties of the more advanced schemes, extending the set-up to include collision avoidance, and performing more detailed simulation studies.

REFERENCES

[1] D.P. Bertsekas, A. Nedić, and A.E. Ozdaglar. *Convex Analysis and Optimization*. Athena Scientific, 2003.

[2] F. Borrelli, T. Keviczky, K. Fregene, and G. J. Balas. Decentralized receding horizon control of cooperative vehicle formations. In *Proceedings of the 44th IEEE Conference on Decision and Control / European Control Conference*, 2005.

[3] R. Carli, F. Fagnani, A. Speranzon, and S. Zampieri. Symmetries in the coordinated rendezvous problem. In P. J. Antsaklis and P. Tabuada, editors, *NESC: Networked Embedded Sensing and Control*, Lecture Notes in Control and Information Sciences. Springer, 2005. To appear.

[4] E. Componogara, D. Jia, B. H. Krogh, and S. Talukdar. Distributed model predictive control. *IEEE Control Systems Magazine*, February 2002.

[5] J. Cortés, S. Martínez, and F. Bullo. Robust rendezvous for mobile autonomous agents via proximity graphs in arbitrary dimensions. *IEEE Transaction on Automatic Control*, 2004. Submitted.

[6] W. B. Dunbar and R. M. Murray. Receding horizon control of multi-vehicle formations: A distributed implementation. In *Proceedings of the IEEE Conference on Decision and Control*, 2004.

[7] W. B. Dunbar and R. M. Murray. Distributed receding horizon control with application to multi-vehicle formation stabilization. *Automatica*, 2005. Submitted.

[8] A. Jadbabaie, J. Lin, and A. S. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control*, 48(6):988–1001, 2003.

[9] R. Olfati-Saber and R. M. Murray. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on Automatic Control*, 49(9):1520–1533, 2004.

[10] S. Skogestad and I. Postlethwaite. *Multivariable Feedback Control*. John Wiley & Sons, New York-Chichester-Brisbane, 2005.

[11] S. L. Smith, M. E. Broucke, and B. A. Francis. A hierarchical cyclic pursuit scheme for vehicle networks. *Automatica*, 41(6):1045–1053, 2005.