Min-max Time Consensus Tracking Over Directed Trees

Ameer K. Mulla, Sujay Bhatt H. R., Deepak U. Patil, and Debraj Chakraborty

***Abstract*— In this paper, decentralized feedback control strategies are derived for min-max time consensus tracking of multi-agent systems with bounded inputs that are communicating over directed graphs. Each agent is a linear time invariant system with distinct and rational eigenvalues. The graph contains a directed spanning tree rooted at an agent which generates the reference trajectory for the other agents. This spanning tree is locally identified and the tail agent of each edge tries to match its states with that of the head of the edge in min-max possible time. Using recent Gröbner basis based methods by the authors, these local min-max time problems are solved by deriving Nash equilibrium *feedback* strategies for time optimal pursuit-evasion games. The local min-max strategies lead to global consensus tracking in *min-max* time under some conditions.**

***Index Terms*— Digraph, Pursuit evasion games, Nash equilibrium, Switching surfaces, Gröbner basis, Spanning trees.**

I. INTRODUCTION

Complex large-scale dynamical systems usually consist of a number of subsystems each of which needs to be locally controlled to perform some collective task. Control of such multi-agent systems has been an area of intense recent research and various scenarios like formation control, rendezvous problem, surveillance, consensus problem, synchronization have been widely studied (e.g. see [1] and references therein). The problem of *consensus tracking* of multi-agent systems [2] considers the situation where a collection of agents, through local interactions are required to agree upon a common reference trajectory generated by one of the agents (called as the Root). In this work, the communication topology is assumed to be given by a directed graph with each agent as a node of the graph and a directed edge is present in the direction that allows transmission of information about the states (see [3] for a justification of directed information sharing in multi-agent systems). It is assumed that this directed graph contains a directed spanning tree rooted at the Root agent [4], [5], [6]. Further it is assumed that each agent is linear time invariant system with distinct rational eigenvalues and the input to each agent has a fixed upper bound. The contributions of this work are threefold:

1) Given a directed graph containing a directed spanning tree as above, a distributed algorithm, implementable with directed information sharing, is proposed to identify the spanning tree.

2) Each edge of the spanning tree is considered separately as an evader-pursuer pair where the agent at the tail of the edge (pursuer) tries to match the states of the agent at the head of the edge (evader) in minimum possible time. However the tail has no information about the future actions of the head, and hence has to guard against possible worst case maneuvers from the head agent. Hence min-max time pursuit *feedback* strategies are computed for each tail agent using extensions of recently proposed Gröbner basis based methods for computation of time optimal feedback strategies [7], [8]. Though irrelevant for the consensus tracking problem, these *closed loop* strategies are also shown to achieve Nash equilibrium for the corresponding pursuit evasion game.

3) When the solution of the local pursuit strategies is applied to each agent (except the Root), the agents on the tree achieve consensus in min-max time, which is guaranteed to be finite provided that each agent is stable.

The local min-max time problem (see (2) above) is a restatement of the classical pursuit evasion differential game where, the pursuer tries to minimize the time of capture (defined here as a matching of the states), while the evader tries to maximize it. Pursuit-evasion game was first considered by [9] and [10], and later on developed extensively by [11], [12], [13], [14] and many others. Positional differential games to obtain feedback solutions were considered in [15]. These methods required the computation of the value function for solving the differential game. Optimal feedback control strategies were then constructed using the computed value function. However, it is well known that in a time-optimal pursuit-evasion game with bounded controls, the value function is non-smooth [16]. Various authors have got around this difficulty of synthesizing feedback controls by considering the notion of weak solutions, commonly known as viscosity solutions [17], [16] and $v-$and $u-$stable functions [15]. Numerical methods have been employed to compute the feedback strategies by solving Two-point boundary value problem (TPBVP) [18]. However, recent developments in computing time-optimal feedback control for linear systems, using techniques from algebraic geometry [8], [7], suggests a novel way to compute the feedback strategies. The state-feedback control laws proposed in [8], [7] for LTI systems with rational, distinct and non-zero eigenvalues, make use of the implicit representation of switching surfaces computed using Gröbner basis based elimination method. By using these switching surfaces, the method proposed in this paper eliminates the need for solving TPBVP and also circumvents the viscosity solution approach by constructing the controls without the need for explicit construction of the value function, at the cost of assuming rational and distinct eigenvalues.

This paper is partially supported by Department of Science and Technology Grant SB/S3/EECE/051/2013.

Authors are with the Department of Electrical Engineering, Indian Institute of Technology, Bombay, Mumbai, India, e-mail: {ameerkm,sujaybhatt,deepakp,dc}@ee.iitb.ac.in.

Though the proposed method successfully computes Nash equilibrium feedback policies for both the pursuer and evader, the consensus tracking formulation only requires the implementation of the min-max pursuer strategies by the tail agents of each directed edges in the spanning tree. For example in Figure 1b, agent 3 tries to capture agent 1 in min-max time, while agent 1 and agent 2 independently try to capture agent 0 in min-max possible time.

Game-theoretic methods for the consensus problem were also used in [19]. However, unlike our case, a quadratic cost is used as a value function for each agent and the objective is to achieve consensus while minimizing the cost.

In systems theory,consensus problem was first proposed in [20], where an estimation problem with multiple observers was considered. It was followed by [21], [22], where agreement problem in distributed decision making was dealt with. The consensus problem for single integrators was initially formulated in [2], [23], [6]. A consensus protocol for double integrators with saturated inputs was considered in [24]. The consensus problem for general linear systems in state-space form with *unbounded* inputs was addressed in [25]. Various communication topologies like directed or undirected, fixed or switching were addressed in the literature and the details of these can be found in [26], [1], [27] and the references therein.

All these papers deal with designing control laws for asymptotic convergence. In [28], gradient descent flow is used to design a discontinuous protocol to achieve consensus in finite time. In [29], finite time consensus problem for multi-agent systems with single integrators is addressed, while [30] consider systems with double integrator agent dynamics. The interested reader can refer to [31] and [32] for a review of all the recent work.

The remaining paper is organized as follows. In section II, the problem of consensus tracking is formulated. Closed-loop feedback strategies are derived for pursuit-evasion games in section III. Then, in section IV using these feedback strategies, decentralized local control laws are proposed for consensus tracking.

II. Problem Formulation

A. Basic Definitions and Assumptions

Consider a directed graph (*digraph*) $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with a finite set of nodes $\mathcal{V}$ and a set of edges $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. A graph $\mathcal{G}_s(\mathcal{V}_s, \mathcal{E}_s)$ is called a *subgraph* of the given directed graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ if $\mathcal{V}_s \subseteq \mathcal{V}$ and $\mathcal{E}_s \subseteq \mathcal{E}$. When there is a connection from node $a_i \in \mathcal{V}$ to $a_j \in \mathcal{V}$, we say that the digraph $\mathcal{G}$ has an edge $(a_i, a_j) \in \mathcal{E}$, where a_i is referred as *parent* node of a_j. Alternatively, node a_i is called the *head* and node a_j is called the *tail* of the edge. A *directed path* is a sequence of edges $(a_{i_k}, a_{j_k}) \in \mathcal{E}$ with $k = 1, \cdots, r$ such that $a_{i_{k+1}} = a_{j_k}$ and $a_{i_k} \neq a_{i_l}$ for $k \neq l$. A *directed cycle* is a directed path such that $a_{i_r} = a_{i_1}$. For $a_i, a_j \in \mathcal{V}$, the node a_i is said to be connected to the node a_j when there exists a directed path from a_i to a_j. The *distance* from a_i to a_j (defined only when a_i is connected to a_j) is the number of edges in the shortest path from a_i to a_j. A *rooted directed tree* is a digraph such that there exists a node (called Root) and a directed path from that node to all other nodes in the digraph. A digraph $\mathcal{G}$ is said to contain a *directed spanning tree*, if there exists a rooted directed tree $\mathcal{G}_d = (\mathcal{V}_d, \mathcal{E}_d)$ such that $\mathcal{V}_d = \mathcal{V}$ and $\mathcal{E}_d \subseteq \mathcal{E}$.

Consider $N + 1$ agents with identical dynamics communicating over a directed graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$. These agents are defined by linear time invariant differential equations:

$$\dot{x}_i(t) = Ax_i(t) + Bu_i(t) \qquad \text{for } i = 0, 1, \cdots, N \tag{1}$$

where $x_i \in \mathbb{R}^n$ are the states and $u_i \in \mathbb{R}$ is the input of the i^{th} agent. The 0^{th} agent (also called the Root) generates the reference trajectory for the other N agents. The following assumptions are made:

i) The pair (A, B) is controllable and the eigenvalues of A are distinct and rational.
ii) The given digraph contains a directed spanning tree.
iii) Root input is constrained by $|u_0| \leq \alpha$ while the inputs of the other agents obey $|u_i| \leq \beta$ for $i = 1, \cdots, N$, where $\alpha < \beta$.

It should be noted that, the proposed method also works for a chain of n-integrators (see remark 10).

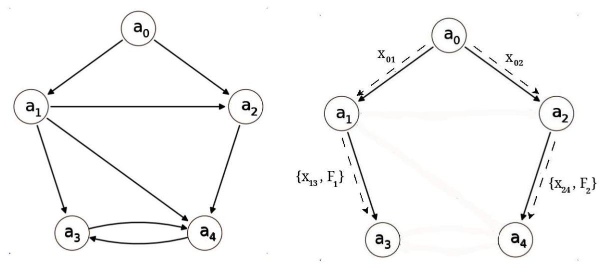
B. Information Flow

We compute a decentralized control strategy so that the other N agents track the trajectory defined by the Root a_0 in min-max time. We treat each edge of a directed graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ as a pair (a_i, a_j) of agents in which, head node i.e., i^{th} agent a_i is the leader and tail node i.e., j^{th} agent a_j is the follower. The node a_i acts as a leader for those agents on which the edges originating from a_i are incident and as a follower for those agents from which the edges incident on a_i are originating. This situation is explained in the Figure 1a. Here, node a_1 acts as a leader for the agents a_3 and a_4, and as a follower for agents a_0. It is possible that multiple head nodes are incident on a single tail node. In this case the follower node has multiple leader nodes to follow. For example in the Figure 1a, node a_4 has two leader nodes a_1 and a_2. Also, in some cases there can be a directed cycle (see for example nodes a_3 and a_4 in Figure 1a). In order to resolve the issue of multiple leaders and directed cycles, we first propose a decentralized method to select a directed spanning tree $\mathcal{G}_d = (\mathcal{V}_d, \mathcal{E}_d)$ from the given directed graph. Then we solve the problem of consensus tracking for the directed spanning tree $\mathcal{G}_d$ thus selected.

Consider an edge $(a_i, a_j) \in \mathcal{E}_d$ directed from agent a_i to agent a_j. Let the dynamics of this leader-follower pair be given by

$$\begin{aligned} a_i : \dot{x}_i(t) &= Ax_i(t) + Bu_i(t) \\ a_j : \dot{x}_j(t) &= Ax_j(t) + Bu_j(t) \end{aligned} \tag{2}$$

where $x_i(t)$, $x_j(t)$ are the states and $u_i(t)$, $u_j(t)$ are the inputs of leader and follower respectively. For the edge (a_i, a_j), the leader a_i has no information about x_j or u_j. Whereas, the agent a_j is receiving the relative state information $x_{ij} = x_i - x_j$ from the leader a_i and is required to reach the trajectory of a_i in the shortest possible time. The follower a_j also receives a Capture Flag (say a binary value $F_i \in \{0, 1\}$) when the leader a_i captures (i.e. matches the states of) the Root agent. Let $F_i = 0$ indicate that a_i has not yet captured the Root, and $F_i = 1$ otherwise. The information flow is indicated by Figure 1b. The leader in each edge communicates its input bound to its follower by raising the flag. Agents a_1 and a_3 form the leader-follower pair (a_1, a_3). Agent a_1 communicates $\{x_{13} = x_1 - x_3, F_1\}$ to a_3.



(a) Digraph $\mathcal{G}$ (b) Information flow

Fig. 1: Digraph and Information flow along a directed tree

C. Problem Statements

Under the above assumptions, the problems to be solved in this paper can be stated precisely as follows:

Problem 1. Find a decentralized scheme to select a directed spanning tree $\mathcal{G}_d = (\mathcal{V}_d, \mathcal{E}_d)$ from the given directed graph $\mathcal{G}$.

We consider the worst case scenario wherein leader a_i acts in the worst possible way by choosing the control strategy u_i, to maximally delay consensus with the follower a_j, who in turn tries to catch its leader (evader) in the minimum possible time. Hence we solve following problem for each edge $(a_i, a_j) \in \mathcal{E}_d$.

Problem 2. For a given pair of agents (a_i, a_j), compute the control laws $u_i^*(t)$ and $u_j^*(t)$ respectively such that for some some time $T_{ij} \in [0, \infty)$, $x_i(t) = x_j(t)$ and the payoff function, $J = \int_0^{T_{ij}} 1dt$, satisfies the following condition for *Nash equilibrium*:

$$J(x_{ij}(0), u_i^*, u_j) \leq J(x_{ij}(0), u_i^*, u_j^*) \leq J(x_{ij}(0), u_i, u_j^*)$$

for all u_i, u_j.

The decentralized min-max time consensus tracking problem is stated next.

Problem 3. Find the decentralized control strategy for each agent such that the collection tracks the trajectory generated by the Root agent in finite time.

III. Time-Optimal Pursuit-Evasion Games

In this section, we derive the feedback strategies for both pursuer and evader and these strategies are shown to constitute a Nash equilibrium. Any pair (u_p^*, u_e^*) of strategies correspond to *Nash equilibrium*, if for all u_p and u_e, the following condition is satisfied:

$$J(x(0), u_p^*, u_e) \leq J(x(0), u_p^*, u_e^*) \leq J(x(0), u_p, u_e^*)$$

The pursuer and evader states are governed by the following linear time invariant differential equations:

$$\text{Evader:} \quad \dot{x}_e(t) = Ax_e(t) + Bu_e(t) \quad |u_e(t)| \leq \alpha \tag{3}$$
$$\text{Pursuer:} \quad \dot{x}_p(t) = Ax_p(t) + Bu_p(t) \quad |u_p(t)| \leq \beta \tag{4}$$

The admissible controls of both the pursuer and evader are given by measurable functions $u_p : \mathbb{R}^+ \to P$ and $u_e : \mathbb{R}^+ \to E$ respectively where $E = [-\alpha, \alpha]$ and $P = [-\beta, \beta]$. Using the pursuer (4) and the evader dynamics (3), we define following *difference system* which captures the dynamics of the relative states i.e. $x(t) = x_p(t) - x_e(t)$ between the pursuer and the evader:

$$\dot{x}(t) = Ax(t) + Bu_{ep}(t) \tag{5}$$

where $u_{ep}(t) = u_p(t) - u_e(t)$ and $x(t) = x_p(t) - x_e(t)$.

First, open loop strategies for the pursuer and evader are derived as in [13] and [33]. Using these open loop strategies and the method of computing the switching surfaces using Gröbner basis methods which was presented by the authors in [7] and [8], *feedback strategies* for the time-optimal pursuit evasion game are derived.

A. Open loop Strategies

1) Min-max : For any given pursuer strategy u_p, let the evader choose a control u_e^* to maximize the cost $J = T(u_p, u_e)$ i.e., $T(u_p, u_e^*) = \max_{|u_e(t)| \leq \alpha} T(u_p, u_e)$. The pursuer then chooses a strategy u_p^* to capture the evader in min-max possible time i.e., u_p^* is selected to satisfy

$$J^+ = \min_{|u_p(t)| \leq \beta} \max_{|u_e(t)| \leq \alpha} T(u_p, u_e)$$

Using difference system dynamics given by equation (5), we form the Hamiltonian H given by $H = \lambda^T(Ax + B(u_p - u_e)) + 1$.

Theorem 4. *[13] The necessary conditions on u_p and u_e for the stationarity of $J = T(u_p, u_e)$ are given by*

$$\dot{\lambda} = -\frac{\partial H}{\partial x} = -A^T\lambda \qquad \lambda(0) = \lambda_0 \tag{6}$$
$$H^* = \min_{u_p} \max_{u_e} (\lambda^T(Ax + B(u_p - u_e)) + 1) \tag{7}$$

The equation (6) describes the co-state dynamics and its solution is $\lambda(t) = e^{-A^T t}\lambda_0$. Thus, the optimal inputs u_e^* and u_p^* for pursuer and evader satisfying the above necessary conditions are given by

$$u_e^*(t) = \arg\max_{u_e} H(u_p, u_e) = -\alpha sign(\lambda_0^T e^{-At} B) \tag{8}$$
$$u_p^*(t) = \arg\min_{u_p} H(u_p, u_e^*) = -\beta sign(\lambda_0^T e^{-At} B) \tag{9}$$

From (8) and (9), it is inferred that under the maximizing strategy (8) played by the evader, the pursuer should play (9) to minimize the capture time.

2) Max-min: As opposed to min-max, for any given evader strategy u_e, we now allow the pursuer to choose a control u_p^* to minimize the cost $J = T(u_p, u_e)$ i.e., $T(u_p^*, u_e) = \min_{|u_p(t)|\leq\beta} T(u_p, u_e)$. Then the evader plays a strategy u_e^* to maximize capture time i.e., u_e^* is selected to satisfy

$$J^- = \max_{|u_e(t)|\leq\alpha} \min_{|u_p(t)|\leq\beta} T(u_p, u_e)$$

Using similar arguments as in the min-max case, the optimal inputs for pursuer and evader respectively for the max-min situation [33] are given by

$$u_e^*(t) = -\alpha sign(l_0^T e^{-At} B) \quad (10)$$
$$u_p^*(t) = -\beta sign(l_0^T e^{-At} B) \quad (11)$$

where l_0^T is the corresponding multiplier. From (10) and (11), it is inferred that under the minimizing strategy (11) played by the pursuer, the evader should play (10) to maximize the capture time.

B. Feedback Strategies

Let the relative position dynamics between the pursuer and evader be given by equation (5). Now consider the following problem:

Problem 5. Compute a control $u(t)$ for transferring the state of (12) from any initial state $x(0) \in X_0 \subset \mathbb{R}^n$ to the origin in minimum possible time t_{min}.

$$\dot{x}(t) = Ax(t) + Bu_{ep}(t) \quad x(0) \in X_0,\ |u_{ep}(t)| \leq \beta - \alpha \quad (12)$$

The set X_0 is a set of states which can be steered to the origin using admissible controls. This set is called as *Null-controllable region* (see [34], [35], [7], [8] for details about characterization and computation of such regions) and is defined as follows:

$$X_0 = \bigcup_{0\leq t<\infty} \{x | x = \int_0^t e^{-A\tau} Bu(\tau)d\tau,\ \forall |u(t)| \leq \beta - \alpha\} \quad (13)$$

Proposition 6. *The computation of the pursuer optimal control u_p and evader optimal control u_e in open loop, is equivalent to problem 5. Moreover, $J^+ = J^-$ i.e.,*

$$t_{min} = \max_{|u_e(t)|\leq\alpha} \min_{|u_p(t)|\leq\beta} T(u_p, u_e) \quad (14)$$
$$= \min_{|u_p(t)|\leq\beta} \max_{|u_e(t)|\leq\alpha} T(u_p, u_e)$$

Proof: It is clear that from equations (8),(9) and (10),(11) the pursuer and evader strategies have same switching function (for both min-max and max-min objectives). Therefore, we conclude that when both evader and pursuer are playing optimal strategies, the input to (5) is constrained by $|u_{ep}| = |u_p - u_e| \leq \beta - \alpha$. Hence, the order in which players initiate the game doesn't matter and as a result (14) follows. □

Lemma 7. *Let the initial condition of the difference system (5) be $x(0) \in X_0$. There exists $T < \infty$ such that $x_p(t) = x_e(t)$ for all time $t \geq T$, if and only if $\beta > \alpha$.*

Proof: Suppose $\beta = \alpha$. Since both pursuer and evader are using the same switching function, the input for the difference system (5) is zero (by proposition 6). Hence, the difference system becomes $\dot{x}(t) = Ax(t)$ where $x(t) = x_p(t) - x_e(t)$. Now, if the pursuer (or evader) dynamics is stable (i.e., all eigenvalues of A are in left half of the complex plane) then the state of the difference system reaches the origin asymptotically i.e., T is not finite. Hence there does not exist any $T < \infty$ such that $x_p(t) = x_e(t)$ for all time $t \geq T$.

Similarly, if $\beta > \alpha$ then the input u_{pe} for the difference system (5) is constrained as $|u_{pe}| \leq \beta - \alpha$ (see proposition 6). Therefore, the set X_0 is a non-empty open convex set containing the origin [34], [35]. By definition of X_0, there exists an admissible control which transfers the initial condition $x(0) \in X_0$ to the origin. Thus, there also exist a time optimal control which transfers the initial condition $x(0)$ to the origin in minimum time $T < \infty$ [33]. □

Assuming rational, distinct and nonzero eigenvalues for the matrix A, computation of switching surfaces and construction of the state-feedback control was presented in [7], [8]. These assumptions are relaxed to include a zero eigenvalue in [36]. Due to space constraints, we omit the computation of switching surfaces for systems with zero eigenvalues and present briefly the computation for rational, distinct and non-zero eigenvalues only.

1) Computation of switching surfaces : In this section, for simplicity, we will drop the subscript from the control $u_{pe}(t)$ and denote it by $u(t)$. Without loss of generality it is assumed that $|\alpha - \beta| = 1$. The control $u(t)$ must drive the states of (5) to the origin at some time $t \in [0, \infty)$. It is well known that optimal control which transfers the state of system to the origin in minimum time is bang-bang i.e., it switches between $+1$ and -1 at no more than $n-1$ switching instants [33], [34]. Thus, we define following functions characterizing those states which can be steered to the origin using a bang-bang ($u(t) = \pm 1$) input with $(k-1)$-switches:

$$\left.\begin{aligned} F_k^+(t_1, ..., t_k) &= \Big(-\int_0^{t_1} + \int_{t_1}^{t_2} - \\ &\quad ... +(-1)^k \int_{t_{k-1}}^{t_k}\Big) e^{-A\tau} B d\tau \\ F_k^-(t_1, ..., t_k) &= -F_k^+(t_1, ..., t_k) \end{aligned}\right\} \quad (15)$$

Define the set $V_k := \{(t_1, t_2, ..., t_k) : 0 \leq t_1 \leq t_2 \leq ... \leq t_k < \infty\}$ for $k = 1, ..., n$. Then the set of states, which can be steered to the origin in $(k-1)$-switches with $u = +1$ to begin with, is denoted by M_k^+ and is defined as follows:

$$M_k^+ = \{x : x = F_k^+(v), \forall v \in V_k\} \quad (16)$$

Similarly, starting with $u = -1$,

$$M_k^- = \{x : x = F_k^-(v), \forall v \in V_k\} \quad (17)$$

Thus, the set of all states which can be steered to the origin in $(k-1)$-switches is defined as follows: $M_k = \{x :$

$x = F_k^{\pm}(v), \ \forall v \in V_k\}$ i.e., $M_k = M_k^+ \cup M_k^- \ \forall k = 1, ..., n$. Now, the set of initial conditions which can be steered to origin with only bang-bang inputs (with at most $n-1$ switches) is equal to X_0 i.e., $M_n = X_0$. The set X_0 is divided symmetrically into two parts M_n^+ and M_n^- by M_{n-1}. Further, M_{n-1} is divided into two parts M_{n-1}^+ and M_{n-1}^- by M_{n-2} and so on [34]. In general, we write $M_k = M_k^+ \cup M_k^-, \ \forall k = 1, ..., n$ and $X_0 = M_n^+ \cup M_n^-$. Thus, the structure of set X_0 obeys the inclusion relation $M_0 \subset M_1 \subset ... \subset M_n$.

Remark 8. We further assume that A is diagonal because of following reasons. For any real similarity transformation $\widehat{x} = Tx$, the corresponding set $\widehat{M}_k^+ = \{\widehat{x} = Tx : \ x \in M_k^+\}$ and similarly $\widehat{M}_k^- = \{\widehat{x} = Tx : \ x \in M_k^-\}$ [34]. Thus, it is enough to compute M_k^+ and M_k^- for the diagonalized system. The corresponding $\widehat{M}_k^+$ and $\widehat{M}_k^-$ for all similar systems can then be computed accordingly.

It is clear from the description of the set X_0 (or M_n) that optimal control $u(t)$ is such that, if $x(t) \in M_n^+$ then $u(t) = 1$ and if $x(t) \in M_n^-$ then $u(t) = -1$. We also know that M_n is divided into two parts M_n^+ and M_n^- by M_{n-1}. Therefore, to decide whether the state $x(t)$ is in M_n^+ or M_n^-, we use M_{n-1} (which henceforth will also be referred as the *Switching Surface*). However, we have the parametric representation of M_{n-1} which depends on the switching instants $t_1, ..., t_{n-1}$ and is not useful for state-based switching. Hence, we eliminate the dependence of the set M_k (defined by (16) and (17)) on $t_1, ..., t_k$ and convert the parametric representation into an implicit representation which depends only on the state variables $(x_1, ..., x_n)$ as follows. Assuming A to be diagonal without loss of generality (see remark 8 above), each component of state x $(x_i, \ i = 1, .., n)$ can be written as some other function denoted as $f_{ki}^{\pm}$, with arguments $e^{-\lambda_i t_1}, ..., e^{-\lambda_i t_k}$ for all $i = 1, ..., n$. Here f_{ki}^+ corresponds to F_k^+ and f_{ki}^- to F_k^-. Thus, $x_i = f_{ki}^{\pm}(e^{-\lambda_i t_1}, ..., e^{-\lambda_i t_k}) \ \forall i = 1, ..., n; \ \lambda_i \in \lambda(A)$. Under the assumption that eigenvalues of A are rational, the denominator of λ_i is denoted by d_i $(i = 1, ..., n)$. Let the least common multiple of denominators d_i's be $l = lcm(d_1, ..., d_n)$. Substituting $z_i = e^{-\frac{t_i}{l}} \ \forall i = 1, ..., k$, converts the parametric representations of x_i's (with t_i's as parameters) into a representation in terms of z_i's. Observe that if the eigenvalues of A are of same signs, then the substitution will express x_i's as polynomials in z_i's. On the other hand, if the eigenvalues of A have mixed signs, then we get rational parametric representations of x_i's. After the substitution, set M_k is given by

$$M_k^+ = \{(x_1, ..., x_n) : x_i = \frac{N_{ki}^+(z_1, ..., z_k)}{D_{ki}^+(z_1, ..., z_k)} \quad (18)$$

$$\forall i = 1, ..., n, 0 < z_k \leq z_{k-1} \leq ... \leq z_1 \leq 1\} \quad (19)$$

$$M_k^- = -M_k^+$$

where N_{ki}^+ and D_{ki}^+ are polynomial numerators and denominators of f_{ki} respectively.

To eliminate $z_j, \quad j = 1, ..., n-1$ from the parametric representation of switching surface M_{n-1},which is given by (18) for $k = n-1$, we will follow the standard implicitization steps (please refer to [37]) described next:

1) Form an ideal $J_{n-1}^+ = \langle D_{n-1,1}^+ x_1 - N_{n-1,1}^+, ..., D_{n-1,n}^+ x_n - N_{n-1,n}^+, 1 - D_{n-1,1}^+ D_{n-1,2}^+ ... D_{n-1,n}^+ y \rangle$.
2) Compute Gröbner basis G_{n-1}^+ of J_{n-1}^+ w.r.t. lexicographic ordering as $y \succ z_1 \succ z_2 \succ ... \succ z_{n-1} \succ x_1 \succ ... \succ x_n$.
3) The element $g_{n-1}^+ \in G_{n-1}^+ \cap \mathbb{Q}[x_1, ..., x_n]$ defines the smallest variety containing the parametric representation $x_i = f_{n-1,i}^+$.

We also need to eliminate the dependence of condition (19) on $z_1, ..., z_{n-1}$ and express it in terms of $x_1, x_2, ..., x_n$ only. To do this we solve each z_j in terms of $x_1, x_2, ..., x_n$ and then impose inequality (19). Let us denote the Gröbner basis obtained by using ordering $.. \succ .. \succ ... \succ z_j^+ \succ x_1 \succ ... \succ x_n$ by $G_{z_j}^+$. Let the elements of $G_{z_j}^+$ which are polynomials in the variables $(z_j, x_1, ..., x_n)$ be $g_1^+, g_2^+, ..., g_m^+$. Now, it might be possible that z_j^+ appears linearly in at least one polynomial among $g_i^+ = 0$ $(i = 1, ..., m)$, and hence z_j^+ can be expressed as a closed form expression of $(x_1, .., x_n)$). Otherwise, since the state values are known, we substitute $(x_1, .., x_n)$ in to any one (usually the simplest) of $g_i^+ = 0$ $(i = 1, .., m)$. This single variable polynomial is then solved numerically for z_j^+. Lastly, we check whether any one set of solutions z_j^+ $(j = 1, ..., n-1)$ satisfies the corresponding inequality (19). The method works similarly for each z_j^+, $j = 1, ..., n-1$.

Now that we have a complete description of the switching surface in terms of state variables, algorithm 1 defines the optimal strategy for both the pursuer and the evader.

Algorithm 1 Switching Logic

Given: {current $x(t)$ and $u(t)$ }

If $\{x(t) \in M_{n-1}\}$ **Then** $\{ \ u(t+\delta) = -u(t)\}$
Else $\{u(t+\delta) = u(t)\}$

Remark 9. Note that the expensive Gröbner basis computation is performed off-line. The only computation that is required to be done on-line is evaluating the polynomials and in the worst case, solving univariate polynomial equations.

2) Computation of Capture Time: Recall that X_0 is divided in two parts viz. $M_n^+ = \{x_i = f_{ni}^+(z_1^{p_i}, ..., z_n^{p_i}) \forall i = 1, ..., n : 0 \leq z_n \leq z_{n-1} \leq ... \leq z_1 \leq 1\}$ and M_n^-. To compute the time required for transferring the initial relative position to the origin, we must first compute z_n given the initial relative position $x(0) \in X_0$. For that we use following procedure:

- Select $g_n^+ \in G_n^+[z_n, x_1, ..., x_n]$ and solve $g_n^+ = 0$ for z_n.
- Select $g_{n-1}^+ \in G_n^+[z_n, z_{n-1}, x_1, ..., x_n]$ and using values of z_n obtained in previous step, solve $g_{n-1}^+ = 0$ for z_{n-1}.
- $\vdots$

- Finally, select $g_{n-1}^{+} \in G_n^{+}[z_n, z_{n-1}, ..., z_1, x_1, ..., x_n]$ and using values of $z_n, z_{n-1}, ..., z_2$ obtained in previous step, solve $g_1^{+} = 0$ for z_1.

Now, as the initial relative state may lie in either of the two sets M_n^{+} or M_n^{-}, repeat the above procedure for G_n^{-} and identify combinations of $z_1, \ z_2, \ \cdots, \ z_n$ such that the inequality $0 < z_n \leq z_{n-1} \leq \cdots \leq z_1 \leq 1$ is satisfied. The value of $t_i = -l\log(z_i)$ gives the corresponding switching instants $t_1, \ t_2, \ \cdots, \ t_n$ and the optimal *capture time* is

$$T = t_n = -l\log(z_n)$$

Remark 10. It is well known that for a chain of integrators, the parametric description of switching surfaces is already in polynomial form [38], [33]. Hence the implicitization step and the capture time computation procedure is applicable to chain of integrators also.

IV. Consensus Tracking Problem

In this section, we first solve problem 3 for a given directed spanning tree using the methods developed for problem 2 in section III. Next a decentralized method to select a spanning tree from the directed graph is proposed (problem 1). The decentralized control scheme developed for directed trees is then applied to the selected spanning tree.

A. *Consensus Tracking Over Directed Trees*

Let the communication topology between agents be given by a directed tree over $N+1$ agents which has exactly N communication links. Thus, each agent a_i ($i \neq 0$) other than the Root has exactly one parent node i.e., only one incoming edge. The Root generates the reference trajectory for other agents to follow. Agents other than the Root, choose their control action based on the information received from their respective parent nodes. Those agents which are connected by a directed edge form a leader-follower pair. Since corresponding to each follower there is a unique leader, each pair of leader-follower can be treated separately. We propose to apply the min-max policy computed in section III to each agent except for the Root agent. For e.g., in Figure 1b agent a_1 applies the min-max policy for the a_0-a_1 pair, while agent a_3 applies the min-max policy for the a_1-a_3 pair. Similarly agent a_2 applies the min-max policy for a_0-a_2 pair. It is evident from Lemma 7, that the available input bound for each agent plays a crucial part in the local finite time capture and consequently in global finite time consensus.

1) Root Agent as Target (Capture Flag $F_i = 1$): As described in section II, the Root dynamics is given by (1) for $i = 0$, and it generates a reference trajectory by using input u_0 which is constrained as $|u_0| \leq \alpha$. The input of other N agents is constrained as $|u_i| \leq \beta$ for $i = 1, \cdots, N$, where $\alpha < \beta$.Hence, if any agent connected directly to the root agent is trying to capture the root, the input difference $\beta - \alpha > 0$ leads to the difference system 5 having a non-empty Null-controllable region X_0 (see equation (13)) about the origin [7]. It is also known that the set X_0 is a bounded convex set containing the origin for unstable systems [34], [8], [7] and $\mathbb{R}^n$ for stable systems. According to Lemma 7, $\beta - \alpha > 0$ leads to finite time capture if the initial relative state $x_{0i}(0) = x_0(0) - x_i(0)$ between the Root a_0 and agent a_i satisfies $x_{0i}(0) \in X_0$.

Example 11. Consider five agents communicating over the digraph given in Figure 2b. Input to Root a_0 is constrained to be in the interval $E = [-1, 1]$ and the inputs to other agents are constrained to be in the intervals $P_i = [-3, 3]$ for $i = 1, 2, 3, 4$. As can be seen from the figure, agents a_1 and a_2 are directly connected to a_0 and therefore the inputs to difference systems formed by (a_0, a_1) and (a_0, a_2) are restricted to the interval $[-2, 2]$. By proposition 6, the agents a_1 and a_2 catch up with the Root in finite min-max time.

2) Any agent other than root as the target (Capture flag $F_i = 0$): Consider Figure 2b in continuation of the last example. Initially, before agent a_1 captures agent a_0, the difference system formed by (a_1, a_3) has input zero. In other words, both the agents a_1 and a_3 might have input β with the effect that the min-max capture time under such condition might be infinity, as argued in Lemma 2b. However, in this situation, such a policy for agent a_3 is still the min-max policy, and one that should be followed to take advantage of possible deviations of a_1 from the worst case policy.

The (a_1, a_3) pair will continue this status quo until agent a_1 captures a_0. At that instant, agent a_1 passes on the new capture flag value $F_1 = 1$ and the input bound α to agent a_3, thus informing that from that instant onwards, the dynamics of agent a_1 will be approximately same as agent a_0. Now, as agent a_3 knows that the input bound for agent a_0 is α, it computes the optimal policy using switching surfaces corresponding to $\beta - \alpha$. In effect, thereafter, agent a_3 will target the Root agent a_0, and the dynamics will evolve according to the $F_i = 1$ case described above.

Initially all agents other than the Root start in $F_i = 0$ mode. However, as agents in each level catch up with the root, their capture flags change and this situation propagates down to the lowest layers in the tree. Obviously, the description pertains to the worst case scenario. It may well be the case that, even in Capture Flag $F_i = 0$ situation, some of the pursuing agents capture their targets before their Capture Flags change to $F_i = 1$. However, it should be noted that consensus cannot be always achieved for unstable systems with bounded inputs.

3) Sufficient condition for consensus : Recall that if the matrix A is unstable then, the set X_0 is a proper subset of $\mathbb{R}^n$ i.e., $X_0 \subset \mathbb{R}^n$. Now, in Figure 2b for example, it is possible that before the status quo between the pair $a_1 - a_3$ gets resolved, the relative state, $x_{13} = x_1 - x_3$, between the follower agent a_3 and the leader agent a_1 on the edge (a_1, a_3), may exit out of the set X_0. In that case, consensus tracking of the leader agent a_1 by the follower agent a_3 is impossible. However, for stable systems, since the set $X_0 = \mathbb{R}^n$, the above situation does not arise. This leads to the following lemma:

Lemma 12. *If all the eigenvalues of matrix A are negative* *then there exists $T < \infty$ such that $x_i(t) = x_j(t)\ \forall\ (a_i, a_j) \in \mathcal{E}_d$ for all time $t \geq T$.*

B. Spanning tree selection

We consider the graphs with directed spanning trees. Distributed methods to select a spanning tree from a given graph were considered in [39], [40], [41]. Here, as the root is already known, we propose a simple *local* method to select the rooted spanning tree.

In a general directed graph, each agent may have more than one parent. If $\mathcal{G}$ contains a directed cycle, arbitrary selection of a parent as the leader does not guarantee that the resultant graph will be a spanning tree. Therefore, each agent has to choose a leader among its parents, using only *local* information, so that the resultant subgraph is a directed spanning tree. In some cases the improper selection of a leader may lead to the resultant subgraph being disconnected. For example, in Figure 1a, if the agent a_3 selects a_4 as the leader and vice versa and rejects all other nodes connected to it then the resultant subgraph becomes disconnected. We propose the following algorithm (Algorithm 2) to make the given graph acyclic using only local information.

Algorithm 2 Selection of acyclic graph

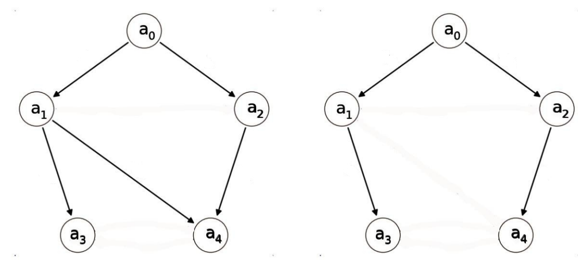
Step-0: The Root agent ignores all the incoming edges and raises a *edge-select* flag.
Step-1: Then the agents which have an incoming edge from the Root ignore the edges coming from all the agents other than the Root and raise their edge-select flags indicating they are connected to the Root. These agents are said to be at level-1 of the hierarchy.
Step-2: The agents having incoming edges from level-1 agents ignore the edges coming from all the agents other than those at level-1 and raise their edge-select flags indicating they are connected to the Root. These agents are said to be at level-2 of the hierarchy.
⋮
Step-j: The agents having incoming edge from level-($j-1$) agents reject the edges incoming from all the remaining agents and raise their edge-select flags indicating they are connected to the Root. These agents are said to be at level-j of the hierarchy.

Algorithm 2 continues until all the agents have raised their edge-select flags. Agents only retain those parents which are at same *distance* from the Root. Since agents ignore incoming edges from the agents on same or higher level, no loops are formed and therefore cycles are eliminated. However, in the resultant subgraph some agents may still have multiple parents. Agent a_i chooses as its parent, the agent a_j, which is at the least distance $\|x_{ji}\|_2$ from it.

Example 13. Consider a system with 5 agents communicating over a directed graph $\mathcal{G}$ as shown in Figure 1a. The dynamics of each agent is given by

$$\dot{x}_i(t) = \begin{bmatrix} -1 & 0 \\ 0 & -2 \end{bmatrix} x_i(t) + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u_i(t) \qquad \text{for } i = 0, \cdots, 4$$

$|u_0(t)| \leq 1$ and $|u_i(t)| \leq 3$ for $i = 1, 2, 3, 4$.



(a) Graph after Algorithm 2 (b) Directed spanning tree

Fig. 2: Communication Graph

After execution of Algorithm 2, the digraph reduces to Figure 2a. In this graph, a_4 has two incoming edges one from a_1 and a_2. Agent a_4 has to choose one of the two parents as the leader. In this example agent a_4 chooses a_2 as its leader and the resultant directed spanning tree is shown in Figure 2b. The switching surface for all agents $k = 1, 2, 3, 4$ after their respective leaders have raised the capture flags, is as follows :

$$S_k = \begin{cases} \frac{1}{4}x_{0k1}^2 + x_{0k2} - x_{0k2} & \text{for } x_{0k2} > 0 \\ -(\frac{1}{4}x_{0k1}^2 - x_{0k2} + x_{0k2}) & \text{for } x_{0k2} < 0 \end{cases}$$

where $x_{0k} = (x_{0k1}, x_{0k2})$ are the states of the difference system defined by agents (a_0, a_k). The control inputs are chosen according to algorithm 1. A sine wave input is given to a_0 and the simulation results are plotted in Figure 3.

V. Conclusions

Feedback strategies are derived for time optimal pursuit evasion games using the switching surfaces computed using Gröbner basis methods. For a directed graph, a procedure to locally select a spanning tree is proposed and the problem of min-max time consensus tracking over the selected tree for linear time invariant systems with *bounded inputs* is solved.

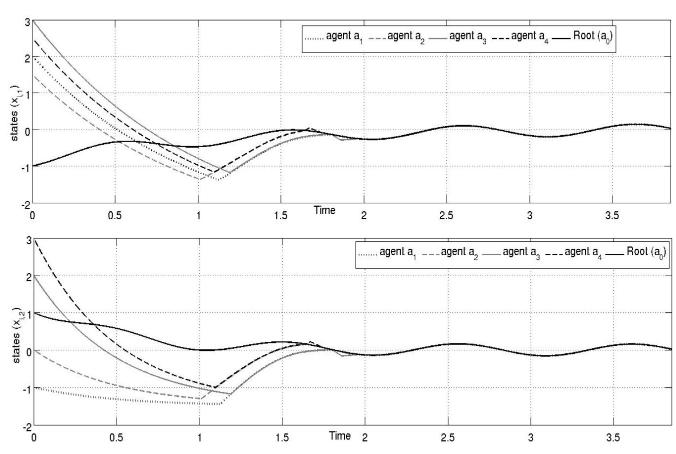


Fig. 3: States v/s Time graph

The method proposed works identically for a chain of n-integrators. The method is currently applicable to controllable and diagonalizable systems having distinct and rational eigenvalues. Relaxing these assumptions is the subject of future research.

REFERENCES

[1] W. Ren and R. Beard, *Distributed consensus in multi-vehicle cooperative control: theory and applications*. Springer, 2008.

[2] A. Jadbabaie and A. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Transactions on Automatic Control*, vol. 48, pp. 988–1001, June 2003.

[3] W. Ren and E. Atkins, "Distributed multi-vehicle coordinated control via local information exchange," *International Journal of Robust and Nonlinear Control*, vol. 17, no. 10-11, pp. 1002–1033, 2007.

[4] K. Moore and D. Lucarelli, "Forced and constrained consensus among cooperating agents," in *Networking, Sensing and Control, 2005. Proceedings. 2005 IEEE*, pp. 449–454, 2005.

[5] W. Ren, "Multi-vehicle consensus with a time-varying reference state," *Systems & Control Letters*, vol. 56, no. 7-8, pp. 474 – 483, 2007.

[6] J. Fax and R. Murray, "Information Flow and Cooperative Control of Vehicle Formations," *IEEE Transactions on Automatic Control*, vol. 49, pp. 1465–1476, Sept. 2004.

[7] D. Patil and D. Chakraborty, "Switching surfaces and null-controllable region of a class of lti systems using groebner basis," in *IEEE 51st Annual Conference on Decision and Control (CDC), 2012*, pp. 1279–1284, Dec. 2012.

[8] D. Patil and D. Chakraborty, "Computation of time optimal feedback control using groebner basis," *accepted in IEEE Transactions on Automatic Control*, 2013.

[9] R. Isaacs, *Differential Games*. Dover Publication Inc., 1965.

[10] D. Kelendzheridze, "On the theory of optimal pursuit," in *Soviet Math. Dokl*, vol. 2, pp. 654–656, 1961.

[11] L. S. Pontryagin, "Linear differential games of pursuit," *Matematicheskii Sbornik*, vol. 154, no. 3, pp. 307–330, 1980.

[12] Y.-C. Ho, A. Bryson, and S. Baron, "Differential games and optimal pursuit-evasion strategies," *IEEE Transactions on Automatic Control,*, vol. 10, no. 4, pp. 385–389, 1965.

[13] A. E. Bryson Jr and Y.-c. Ho, "Applied optimal control: optimization, estimation, and control," 1969.

[14] O. Hajek, *Pursuit Games: An Introduction to the Theory and Applications of Differential Games of Pursuit and Evasion*. Dover Books on Mathematics Series, Dover Publications, 2008.

[15] N. N. Krasovskii, A. I. Subbotin, and S. Kotz, *Game-theoretical control problems*. Springer-Verlag New York, Inc., 1987.

[16] M. Bardi, M. Falcone, and P. Soravia, "Numerical methods for pursuit-evasion games via viscosity solutions," in *Stochastic and differential games*, pp. 105–175, Springer, 1999.

[17] P. E. Souganidis, "Two-player, zero-sum differential games and viscosity solutions," in *Stochastic and differential games*, pp. 69–104, Springer, 1999.

[18] J. Shinar, "Solution techniques for realistic pursuit-evasion games," tech. rep., Israel Institute of Technology, 1980.

[19] E. Semsar-Kazerooni and K. Khorasani, "Multi-agent team cooperation: A game theory approach," *Automatica*, vol. 45, pp. 2205–2213, Oct. 2009.

[20] V. Borkar and P. Varaiya, "Asymptotic agreement in distributed estimation," *IEEE Transactions on Automatic Control*, pp. 0–5, 1982.

[21] J. Tsitsiklis, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *IEEE Transactions on Automatic Control*, no. September, 1986.

[22] J. N. Tsitsiklis, "Problems in decentralized decision making and computation.," tech. rep., DTIC Document, 1984.

[23] R. Olfati-Saber and R. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on Automatic Control*, vol. 49, pp. 1520–1533, Sept. 2004.

[24] A. Abdessameud and A. Tayebi, "On consensus algorithms for double-integrator dynamics without velocity measurements and with input constraints," *Systems & Control Letters*, vol. 59, pp. 812–821, Dec. 2010.

[25] S. Tuna, "Conditions for synchronizability in arrays of coupled linear systems," *IEEE Transactions on Automatic Control*, no. i, pp. 1–22, 2009.

[26] R. Olfati-Saber, J. Fax, and R. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007.

[27] W. Ren, R. Beard, and E. Atkins, "A survey of consensus problems in multi-agent coordination," in *American Control Conference*, pp. 1859–1864, 2005.

[28] J. Cortés, "Finite-time convergent gradient flows with applications to network consensus," *Automatica*, vol. 42, pp. 1993–2000, Nov. 2006.

[29] Q. Hui, W. Haddad, and S. Bhat, "Finite-time semistability and consensus for nonlinear dynamical networks," *IEEE Transactions on Automatic Control*, vol. 53, no. 8, pp. 1887–1900, 2008.

[30] X. Wang and Y. Hong, "Finite-time consensus for multi-agent networks with second-order agent dynamics," *IFAC World Congress*, pp. 15185–15190, 2008.

[31] Y. Cao, W. Yu, W. Ren, and G. Chen, "An Overview of Recent Progress in the Study of Distributed Multi-Agent Coordination," *IEEE Transactions on Industrial Informatics*, vol. 9, pp. 427–438, Feb. 2013.

[32] W. Ren and Y. Cao, *Distributed coordination of multi-agent networks: emergent problems, models, and issues*. Springer, 2011.

[33] L. Pontryagin, V. Boltyanskii, R. Gamkrelidze, and E. Mischenko, *The Mathematical Theory of Optimal Control*. Interscience Publishers, 1962.

[34] O. Hajek, "Terminal manifolds and switching locus," *Theory of Computing Systems*, vol. 6, no. 3, pp. 289–301, 1971.

[35] T. Hu and Z. Lin, *Control Systems with Actuator Saturation: Analysis and Design*. Birkhäuser, 2001.

[36] D. U. Patil, A. K. Mulla, D. Chakraborty, and H. Pillai, "Computation of feedback control for time optimal state transfer using groebner basis," *in preparation*, 2013.

[37] D. Cox, J. Little, and D. O'Shea, *Ideals, Varieties and Algorithms:An Indroduction to Computational Algebraic Geometry and Commutative Algebra*. Springer, 2007.

[38] U. Walther, T. T. Georgiou, and A. Tannenbaum, "On the computation of switching surfaces in optimal control: A gröbner basis approach," *IEEE Transactions on Automatic Control*, vol. 46, no. 4, 2001.

[39] B. Awerbuch, "Optimal distributed algorithms for minimum weight spanning tree, counting, leader election, and related problems," in *Proceedings of the nineteenth annual ACM symposium on Theory of computing*, pp. 230–240, ACM, 1987.

[40] R. G. Gallager, P. A. Humblet, and P. M. Spira, "A distributed algorithm for minimum-weight spanning trees," *ACM Transactions on Programming Languages and systems (TOPLAS)*, vol. 5, no. 1, pp. 66–77, 1983.

[41] P. Humblet, "A distributed algorithm for minimum weight directed spanning trees," *IEEE Transactions on Communications*, vol. 31, no. 6, pp. 756–762, 1983.