Minimal-time uncertain output final value of unknown DT-LTI systems with application to the decentralised network consensus problem

Ye Yuan, Guy-Bart Stan, Ling Shi, Mauricio Barahona and Jorge Gonçalves

***Abstract*— For an unknown discrete-time linear time-invariant (DTLTI) autonomous system, this paper characterises the minimal number of steps necessary to compute the asymptotic final value of an output observed with uncertainty. We show that this minimal number of steps can also be obtained directly from a graphical representation of the DTLTI system using Mason's rule. Moreover, we provide heuristic algorithms to compute the final value in a minimal amount of time with uncertain observations. The general structure of these algorithms is as follows. Step one, by introducing a one-step prediction error metric, we characterise the minimal length of recursion for the outputs of the considered DTLTI system. Step two, by constructing a new data set "close" to the original uncertain output data set satisfying certain conditions, we estimate the final value of the original output set by computing the final value associated with this new data set. Step three, we characterise the difference between the estimated final values obtained from different estimated data sets. Furthermore, we also consider systems with time-delays and investigate how the delays affect the minimal number of steps required to compute the final value. These results find applications in minimal-time network consensus problems with minimal and uncertain (e.g., noisy) information.**

I. Introduction

Linear systems theory has played a key role in many technology advancements in various areas like aerospace, communications, networks and computer engineering. Previous study [6] has shown that without noise or uncertainty in such a system, the final value of the observed output can be computed in minimal time, solely using a finite, minimal number of past values of the observed output.

The algorithm proposed in [6] finds application in various current open problems, one of which is the minimal-time decentralised consensus problem [2]. In the context of decentralised consensus [7], [10], the problem is to ensure that each agent's state reaches consensus asymptotically and to compute this consensus value. However, from a practical point of view, requiring infinite (or arbitrarily long) time to obtain the consensus value is unsatisfactory. In [2], an algorithm for computing the consensus value in finite time is proposed. This algorithm is based on the accumulation of several trajectories of the output, each one corresponding to different initial conditions of the considered system.

Ye Yuan and Jorge Gonçalves are with the Control Group, Department of Engineering, University of Cambridge. Guy-Bart Stan is with the Centre for Synthetic Biology and Innovation, Department of Bioengineering, Imperial College London, Mauricio Barahona is with the Department of Bioengineering, Imperial College London. Ling Shi is with Department of Electronic and Computer Engineering, the Hong Kong University of Science and Technology.

This work was supported in part by EPSRC grant number EP/G066477/1 and by Microsoft Research through the PhD Scholarship Program.

Using our results we improve the previous results presented in [6] by providing an algorithm which is able to compute in *minimal time* the final consensus value of an arbitrarily chosen state, solely using successive observations of the past values of this state. This, in other words, means that we can use the past information of an arbitrarily chosen state in the considered network to predict the future consensus value shared asymptotically by the whole network.

In this paper, we furthermore propose the following extensions to this minimal time consensus value computation result:

Firstly, using Mason's rule for linear networks [17], we relate the degree of the minimal polynomial of the matrix pair $[A, C]$ to the information flow in the graphical representation of such a system. Tight upper and lower bounds on the minimum time required to compute the consensus value are computed by using some results stemming from structural controllability/observability theory for LTI systems [19] (see Sections II and III).

Secondly, standing on the practical side, we consider the case of imperfect output observations, i.e., observations containing additive measurement noise and/or quantisation errors. Algorithms are proposed to estimate the minimal number of steps needed to compute an estimated final value in presence of these imperfect observations (see Section IV).

Thirdly, we consider the case of imperfect communication channels including time delays and provide an extension of the proposed results to discrete-time LTI systems with multiple time delays (see Section V).

A. Notation

For a matrix $A \in \mathbb{R}^{M\times N}$, $A[i,j] \in \mathbb{R}$ denotes the element in the i^{th} row and j^{th} column, $A[i,:] \in \mathbb{R}^{1\times N}$ denotes its i^{th} row, $A[:,j] \in \mathbb{R}^{M\times 1}$ denotes its j^{th} column and $A[i_1 : i_2, j_1 : j_2] \in \mathbb{R}^{(i_2-i_1+1)\times(j_2-j_1+1)}$ denotes the submatrix of A defined by the rows i_1 to i_2 and the columns j_1 to j_2. For a column vector $\alpha \in \mathbb{R}^{N\times 1}$, $\alpha[i]$ denotes its i^{th} element. Similarly for a row vector $\beta \in \mathbb{R}^{1\times N}$, $\beta[i]$ denotes its i^{th} element. Let $\boldsymbol{e}_r^T = [0,\dots,0,1_{r^{th}},0,\dots,0] \in \mathbb{R}^{1\times N}$. $\mathbf{1}$ denotes the vector $[1,\dots,1]^T$ vector with appropriate dimension.

B. Definitions

Definition 1 (Polynomial of a matrix): If $p(t) = \beta_k t^k + \beta_{k-1}t^{k-1} + \cdots + \beta_1 t + \beta_0$ is a given polynomial, then one can define $p(A) = \beta_k A^k + \beta_{k-1}A^{k-1} + \cdots + \beta_1 A + \beta_0 I_N$ for any $A \in \mathbb{R}^{N\times N}$. The polynomial is monic if $\beta_k = 1$.

ISBN 978-963-311-370-7

Definition 2 (Minimal polynomial of a matrix): The minimal polynomial associated with a matrix $A \in \mathbb{R}^{N\times N}$ is denoted by $q(t)$ and is defined as the minimal degree monic polynomial which satisfies $q(A) = \mathbf{0}$.

The Cayley-Hamilton theorem [4] guarantees that for any matrix $A \in \mathbb{R}^{N\times N}$, there is a degree N matrix polynomial with value 0 at A. We may find another polynomial with smaller degree and value zero at A and will discuss this in detail later. It is easy to show that the minimal polynomial of a given matrix is unique [1]. Based on Definition 2, we define the minimal polynomial of a matrix pair $[A, C]$ as follows:

Definition 3 (Minimal polynomial of a matrix pair): The minimal polynomial associated with a matrix pair $[A, C]$ where $A \in \mathbb{R}^{N\times N}, C \in \mathbb{R}^{p\times N}$ is denoted by $q_c(t)$ and is defined as the minimal degree monic polynomial $q_c(\cdot)$ which satisfies $Cq_c(A) = \mathbf{0}$.

Remark 1: The minimal polynomial of $[A, C]$, $q_c(t)$, is not necessarily the same as that of A, $q(t)$. Moreover, it can be shown that $q_c(t)$ divides $q(t)$ for the same A [2]. We denote the degree of the minimal polynomial of $[A, C]$ by $D_c + 1$. It is easy to see that this degree $D_c + 1$ is varying with respect to the observation matrix C [6].

II. Problem formulation and preliminary results

Consider the discrete-time LTI system

$$\begin{aligned} x_{k+1} &= Ax_k \\ y_k &= Cx_k \end{aligned} \tag{1}$$

where $x_k = \begin{bmatrix} x_k[1] & x_k[2] & \dots & x_k[N] \end{bmatrix}^T \in \mathbb{R}^N$, $A \in \mathbb{R}^{N\times N}$, $C \in \mathbb{R}^{p\times N}$ ($p < N$). In particular, in the case of the network consensus problem, we adopt the following assumption:

Assumption 1: In order to guarantee asymptotic consensus, the matrix A in system (1) is assumed to be nonnegative (see [1]) and A has one and only one eigenvalue at 1 and all the other eigenvalues nonzero, real and distinct. Furthermore, A is also assumed to be row-stochastic, i.e., $A\mathbf{1} = \mathbf{1}$. In this paper, we assume that the elements and dimension of A and the initial state x_0 are unknown.

In terms of decentralised network consensus problem, it is natural to define the *minimal information* available to each agent. In this paper, we assume that the only information available to an agent is its own value at different time steps. In particular, we do not assume availability of output measurements from the neighbours of a given agent [9]. Therefore, we consider the most difficult case corresponding to $C = \boldsymbol{e}_r^T$ for arbitrary $r \in \{1, 2, \cdots, N\}$. In that situation, D_c and $q_c(\cdot)$ defined in Section I-B are changed to D_r and $q_r(\cdot)$ respectively for the sake of notational simplicity and coherence.

The main purpose of this paper is to compute, in a minimal amount of time, the *final value*, $\phi = \lim_{k\to\infty} x_k[r]$, based on noisy information. An algorithm to obtain the final value of system (1) based on the accumulation of successive non-noisy observations is proposed in [6] and will be illustrated on an example in Section III. The minimal number of successive observations needed can be characterised in terms of the coefficients of the minimal polynomial of $[A, \boldsymbol{e}_r^T]$. We will review these results for perfect observation in this section and take into account uncertainties in the observations in the later Sections.

Proposition 1: Given a linear system (1) and an initial state x_0, there exist a $d \in \mathbb{N}$ and scalars $\alpha_0, ..., \alpha_d$ such that the following linear regression equation must be satisfied $\forall k \in \mathbb{N} \geq 0$,

$$x_{k+d+1}[r]+\alpha_d x_{k+d}[r]+\ldots+\alpha_1 x_{k+1}[r]+\alpha_0 x_k[r] = 0. \tag{2}$$

Remark 2: An algebraic characterisation of d is given in [6] based on the Jordan block decomposition. If we can compute the unknown coefficients in eq. (2) from data, then we can compute future outputs recursively using eq. (2) and past outputs.

Definition 4 (Minimal length of recursion (see [8])): We define

$$D_r + 1 = \max_{x_0\in\mathbb{R}^N} \min_{d\in\mathbb{N}} \{\text{d: eq. (2) holds for all } k\}$$

and call it the *minimal length of recursion*.

Remark 3: The minimal length of recursion is the same as the degree of the minimal polynomial of $[A, \boldsymbol{e}_r^T]$ (see [6]). Therefore Proposition 1 also indicates that for an arbitrary initial state x_0 and some scalars $\alpha_0, ..., \alpha_{D_r}$, the following equation always holds:

$$x_{k+D_r+1}[r]+\alpha_{D_r} x_{k+D_r}[r]+\ldots+\alpha_1 x_{k+1}[r]+\alpha_0 x_k[r] = 0. \tag{3}$$

As shown in [2], under the assumption that the minimal polynomial in (3) does not possess any unstable root [5], except for one single root located at 1, we can take the Z-transform of (3) and apply the final value theorem to compute the final value of $x_k[r]$ based on the coefficient of the minimal polynomial $q_r(\cdot)$:

$$\begin{aligned} \phi = \lim_{k\to\infty} x_k[r] &= \lim_{z\to 1}(z-1)X[r](z) \\ &= \frac{\begin{bmatrix} x_{D_r}[r] & x_{D_r-1}[r] & \dots & x_0[r] \end{bmatrix} S}{\begin{bmatrix} 1 & 1 & \dots & 1 \end{bmatrix} S}, \end{aligned} \tag{4}$$

in which

$$S = \begin{bmatrix} 1 \\ 1+\alpha_{D_r} \\ 1+\alpha_{D_r-1}+\alpha_{D_r} \\ \vdots \\ 1+\sum_{j=1}^{D_r}\alpha_j \end{bmatrix}.$$

In order to compute the final value, we need to know S. To identify the unknown coefficients in S, we resort to Hankel-type matrices

$$\boldsymbol{X}_r(k,k) \triangleq \begin{bmatrix} x_0[r] & x_1[r] & \dots & x_k[r] \\ x_1[r] & x_2[r] & \dots & x_{k+1}[r] \\ \vdots & \vdots & \ddots & \vdots \\ x_k[r] & x_{k+1}[r] & \dots & x_{2k}[r] \end{bmatrix}. \tag{5}$$

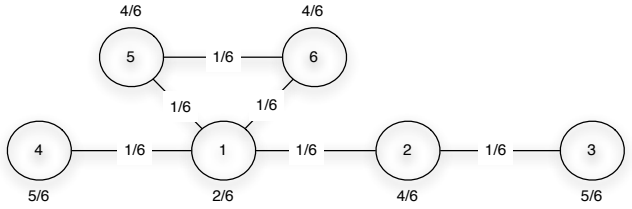

Fig. 1. Graph with constant edge weights of $1/N$.

From Proposition 1, we can see that, when increasing the dimension $k+1$ of this Hankel matrix $\boldsymbol{X}_r(k,k)$, it will eventually lose rank. When it does, at discrete-time step $2D_r+2$, where D_r is defined in eq. (3), one can compute its normalised kernel:

$$\boldsymbol{X}_r(k,k)\begin{bmatrix}\alpha_0 & \alpha_1 & \dots & \alpha_k & 1\end{bmatrix}^T = 0. \tag{6}$$

Furthermore, it can be shown (see [6]) that the normalised kernel obtained from eq. (6) corresponds to the coefficients in eq. (3).

III. An illustrative network-inspired example and its associated graphical interpretation

In this section, we use a network-inspired example (initially proposed in [2]) to illustrate the results stated in last section.

The network topology we are considering is represented in Fig. 1. Because the topology is undirected and connected, the final value of each node is the average of the initial state values (average consensus value). For the randomly chosen the initial state $x(0) = \begin{bmatrix}1.3389 & 2.0227 & 1.9872 & 6.0379 & 2.7219 & 1.9881\end{bmatrix}^T$, the final consensus value is thus 2.6828. We illustrate the results in Section II by focusing on node 1.

Step 1: We increase the dimension of $\boldsymbol{X}_1(k,k)$ until it loses rank. In particular, we have $\text{rank}(\boldsymbol{X}_1(3,3)) = \text{rank}(\boldsymbol{X}_1(4,4)) = 4$.

Step 2: The coefficients of the minimal polynomial (3) involving $x[1]$ can be obtained by computing the normalised kernel of $\boldsymbol{X}_1(4,4)$.

Step 3: The final consensus value can thus be computed using eq. (4) and proven to be equal to 2.6828.

If we repeat these steps for all nodes, we observe that node 1, 2, and 3 only need 9 successive values of their own state to compute the final value while node 4 needs 11 such values and node 5, and 6 need 13. It is not surprising that some nodes need less steps to compute the final consensus value of the network. We call such nodes, *dominant nodes*.

In the rest of this section, we establish a connection between the graphical representation of a network and the algebraic description of system (1) in Section II.

Consider the system in eq. (1), and a corresponding graph [13] $\mathbb{G} = (\mathbb{V}, \mathbb{E}, A/z)$, where $\mathbb{V}, \mathbb{E}$ denotes the vertex/edge set, A is the state-space matrix of the considered DTLTI system in eq. (1) and z is the Z-Transform operator. In the following, we show that the degree of the minimal polynomial of $[A, \boldsymbol{e}_r^T]$ can be determined using Mason's rule [17].

Let $\Phi = (I - A/z)^{-1}$. If we build the signal-flow network for A/z , then from Mason's rule we can obtain the gain from node i to node j directly from the graph as follows:

$$\Phi[i,j] = \frac{1}{\Delta}\sum_{\text{path } p\in\mathbb{G}} T_p\Delta_p, \tag{7}$$

where Δ is the determinant of the graph, which can be computed by

$$\Delta = 1 - \sum L_i + \sum L_iL_j + \dots + (-1)^m \sum \dots$$

T_p is the gain of the p^{th} forward path from node i to node j, L_i is the loop gain of each closed loop in the graph, and L_iL_j is the product of the loop gains of any two non-touching loops (i.e., loops with no common nodes). Δ_p is the cofactor value of Δ for the p^{th} forward path, with the loops touching the p^{th} forward path removed (i.e., the remaining graph when you have removed those parts of the graph that form loops while retaining the parts on the forward path).

The McMillan degree of $\boldsymbol{e}_r^T\Phi$ can be directly computed from the network using Mason's rule in eq. (7). Furthermore, it can be seen that the McMillan degree of $\boldsymbol{e}_r^T\Phi$ [4], i.e., the number of poles, is the same as degree of the minimal polynomial of $[A, \boldsymbol{e}_r^T]$ obtained from method in Section II .

Sometimes, the graph is rather complicated and therefore it is hard to compute the formula (7) from Mason's rule. In this case, one might resort to some basic graphical information [13], e.g., the diameter of the graph, the number of nodes in the graph, etc., to obtain a rough estimate of the minimal number of steps. Based on this idea, we propose the following upper and lower bounds.

Proposition 2: Consider the system in eq. (1). The degree of the minimal polynomial of $[A, \boldsymbol{e}_r^T]$, namely D_r+1, is lower bounded by d_r+1, where d_r is the longest path from node r to all other nodes, and upper bounded by N.

Proof: The upper bound can be obtained directly using the Cayley-Hamilton theorem, i.e., $D_r + 1 \le N$. Therefore, we only need to show the lower bound.

Suppose the minimal polynomial for $[A, \boldsymbol{e}_r^T]$ is $q_r(t) = t^{D_r+1} + \alpha_{D_r}t^{D_r} + \dots + \alpha_1 t + \alpha_0$. Since $\boldsymbol{e}_r^T q_r(A) = \boldsymbol{0}$, we have:

$$\begin{bmatrix}\alpha_0 & \alpha_1 & \cdots & \alpha_{D_r} & 1\end{bmatrix}\begin{bmatrix}\boldsymbol{e}_r^T \\ \boldsymbol{e}_r^TA \\ \vdots \\ \boldsymbol{e}_r^TA^{D_r+1}\end{bmatrix} = 0. \tag{8}$$

From the graphical perspective, the element $A^k[i,j]$ being 0 means there is no path from i to j with length k [13]. Meanwhile, note that the consensus is guaranteed if and only if the digraph is strongly connected (see [10]). Strong connectedness implies that there always exists an edge-following path from node r to any other node in the graph. Therefore, we can pick the longest path, say from node r to node s with length d_r. Based on these two facts, if $A^{d_r}[r,s]$ is nonzero, then eq. (8) implies that $D_r + 1 \ge d_r + 1$. ■

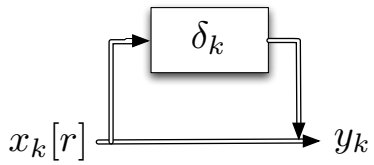

Fig. 2. A commonly used multiplicative uncertainty.

Remark 4: Proposition 2 proposes a fundamental limitation on the minimal number of steps (successive values of a node) needed to compute the final value based on the graphical definition of a network.

IV. Final value for data with uncertainty

This section is motivated by the following scenario. Assume that all the agents are "smart" in the sense that they have enough memory to store their own past observations and enough computational ability. Assuming that the topological structure of the network guarantees consensus, an agent observes its own state with uncertainty and tries, based on successive uncertain observation of its state, to compute the final consensus value of the network. At each time step, an agent may either propose an estimated final value computed from its own algorithm, or wait to get more data from the next time-step.

Generally, when uncertainties, i.e., unmodelled dynamics, noise or quantisation errors on observations, are taken into account, the minimal number of steps cannot be found with certainty. This is due to the output measurements being corrupted by uncertainties. As a consequence, the properties given above will not in general hold. From a network consensus perspective, this raises important questions: are the dominant nodes still dominant?, i.e., do they still need less steps to compute the final value? If so, what is their optimal strategy to compute it?

The currently dominant approach to modelling uncertainty is based on the idea of averaging. However, it requires a large number of data to study the distribution of the noise [16]. In this scenario, there are no advantages for these dominant nodes in knowing the final value in advance. Uncertainty might also lead to instability of the identified model in Section II, and, therefore, the estimated final value based on this identified model might be infinite. In this latter case, no information about the final value can be inferred. In this section, we propose an efficient heuristic algorithm to compute the final value using the same minimal number of successive values of observed state $x[r]$ with uncertainties as the number of successive values of perfect observations, i.e., $2D_r + 3$.

Consider a discrete-time LTI system (1) with uncertainty

$$\begin{aligned} x_{k+1} &= Ax_k, \\ y_k &= \boldsymbol{e}_r^T x_k(1+\delta'_k). \end{aligned} \tag{9}$$

Here, y_k represents the observation with uncertainty at time k of an arbitrarily chosen particular state $x_k[r] \in \mathbb{R}$ and δ'_k denotes the signal to noise ratio (SNR) for the randomly chosen state $x_k[r]$. We assume $|\delta'_k| \leq \delta$ for all k. We normalise this quantity by letting $\delta_k = \frac{\delta'_k}{\delta}$ for all k and, in this case, $|\delta_k| \leq 1$. This multiplicative uncertainty is widely used to model not only noise but also quantisation/transmission errors (see [11]).

Problem 1: Consider system (9) satisfying Assumption 1. At each discrete-time step, the only information available is the output y which is the sum of the observed state $x_k[r]$ and of a statistically unknown uncertainty (we formed them as multiplicative uncertainty here). We are interested in the following two problems:

1. How can one obtain the true length of recursion in eq. (3), i.e., $D_r + 1$? Furthermore how can one obtain the minimal number of steps $2D_r + 3$?
2. How can one quantify the error on the final value based on the data set with uncertainty and the data set without uncertainty.

In the rest of this Section, heuristic algorithms will be proposed to tackle this problem step by step. Considering uncertain data in Problem 1, we first introduce an easy-to-compute quantity to test whether the current step corresponds to the true minimal number of steps, namely the *one-step prediction error* defined in Section IV-A. We then compute the final value from outputs that are "close" to the original ones in Section IV-B and Section IV-C. For simplicity and clarity of exposition, we make the following additional Assumption:

Assumption 2: We assume that all the initial states x_0 are nonnegative, i.e., $x_0[i] \geq 0$ for all i.

Remark 5: This assumption is physically reasonable since, typically, the states in the network usually represents physical quantity like temperature, velocity, humidity, etc. Since A is nonnegative, this constraint on the initial state guarantees that all $x_k[i]$ are nonnegative for all k, i.

A. Identifying the length of recursion

In this section, we will propose a practical method to identify the minimal length of recursion. Before presenting the algorithm, a definition and an important theorem are introduced first.

Definition 5: Define a data set $Y = \{y_0,\ y_1, \ldots\}$ composed of successive scalar outputs ($y_i \in \mathbb{R}$) and its associated normalised kernel $\boldsymbol{\alpha} = \begin{bmatrix}\alpha_0 & \alpha_1 & \cdots & \alpha_k & 1\end{bmatrix}$ (or equivalently, its associated minimal polynomial $t^{k+1} + \alpha_k t^k + \cdots + \alpha_1 t + \alpha_0$) satisfying the iteration in eq. (3). The final value ϕ can be directly computed from eq. (4).

Remark 6: Once the cardinality of the data set Y is greater or equal to $k+1$ (the degree of its associated minimal polynomial), then all the elements in Y are fixed by recursion and therefore ϕ is fixed.

Definition 6: The Hankel matrix associated with the data set $Y = \{y_0,\ y_1, \ldots, y_{2k}\}$ is defined as follows:

$$\boldsymbol{Y}(k,k) \triangleq \begin{bmatrix} y_0 & y_1 & y_2 & \cdots \\ y_1 & y_2 & y_3 & \cdots \\ & & \ddots & \\ y_2 & y_3 & & \\ \vdots & \vdots & & y_{2k} \end{bmatrix}. \tag{10}$$

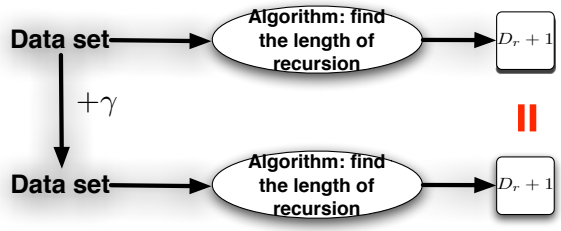

Fig. 3. If an algorithm can obtain the true length of recursion in eq. (2), then it can be shown that it can also obtain the length of recursion from shifted output data as stated in Theorem 1.

Theorem 1: Consider the system (9) satisfying Assumption 1 and Assumption 2, if an algorithm can solve the first problem of Problem 1 with $2D_r+3$ uncertain observations $Y = \{y_0, \ldots, y_{2D_r+2}\}$, then it can also solve the first problem of Problem 1 with $\tilde{Y} = \{y_0+\gamma, \ldots, y_{2D_r+2}+\gamma\}$ for $\gamma \in \Gamma_r \subset \mathbb{R}^+$ (characterised later).

Proof: First, recall that A is row-stochastic, therefore $\gamma A\mathbf{1} = \gamma\mathbf{1}$ and furthermore $x_{k+1}+\gamma\mathbf{1} = A(x_k+\gamma\mathbf{1})$. The final value computed when the shift is applied is thus equal to $\phi+\gamma$. For any known γ, we can subtract γ from the computed final value and recover the original final value. This means that the obtained final values are the same.

We then show that the minimal number of steps is unchanged when this shift is applied. To show this latter property we equivalently show that the following equality holds for all k and t

$$\text{rank}(\boldsymbol{X}_r(k,k)) = \text{rank}(\boldsymbol{X}_r(k,k)+\gamma\mathbf{1}\mathbf{1}^T). \quad (11)$$

By the assumptions made on A in Assumption 1 and Assumption 2, it is easy to show that the expression for $x_k[r]$ is

$$x_k[r] = \sum_{i=1}^{D_r+1} \lambda_i^k \mu_i, \quad (12)$$

where λ_i are distinct roots of the minimal polynomial $q_r(\cdot)$ (which is of degree D_r+1), and μ_i are parameters determined by the initial conditions of $x_k[r]$. Using the results in [3], it can directly be shown that the following equation holds

$$\boldsymbol{X}_r(k,k) = V(0,k)T_rV^T(0,k), \quad (13)$$

in which $\forall k \in \mathbb{N}$, $V(0,k) = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & \lambda_2 & \cdots & \lambda_{D_r+1} \\ \vdots & \ddots & \ddots & \vdots \\ 1 & \lambda_2^k & \cdots & \lambda_{D_r+1}^k \end{bmatrix}$ is a Vandermonde matrix containing the distinct roots of the minimal polynomial of $[A, \boldsymbol{e}_r^T]$ and $T_r = \text{diag}\{\mu_1, \cdots, \mu_{D_r+1}\} \in \mathbb{R}^{(D_r+1)\times(D_r+1)}$ is a diagonal matrix with μ_i on its diagonal. Without loss of generality, let $\lambda_1 = 1$, then

$$\boldsymbol{X}_r(k,k)+\gamma\mathbf{1}\mathbf{1}^T = V(0,k)(T_r+\gamma\boldsymbol{e}_1\boldsymbol{e}_1^T)V(0,k)^T. \quad (14)$$

Observe that since the final consensus value is nonnegative (from Assumption 2), then $\mu_1 \geq 0$. Therefore eq. (11) holds.

Thirdly, we show that the associated minimal polynomial is unchanged. This can be directly inferred from eq. (14).

Finally, since $y_k = \boldsymbol{e}_r^T x_k(1+\delta_k\delta)$ and $x_k \geq 0$, we have

$$y_k+\gamma = (x_k[r]+\gamma)\left(1+\delta_k(\delta\frac{x_k[r]}{x_k[r]+\gamma})\right). \quad (15)$$

Let $\epsilon \triangleq \max_k\{\delta\frac{x_k[r]}{x_k[r]+\gamma}\}$. Then eq. (15) writes $y_k+\gamma = (x_k[r]+\gamma)(1+\epsilon_k\epsilon)$, where $\epsilon_k = \delta_k(\frac{x_k[r]}{(x_k[r]+\gamma)\epsilon})$ and $|\epsilon_k| \leq 1$ for all k. From eq. (15), we see that the effect of the shift in the output set is equivalent to enhance the signal $x_k[r]$ while reducing the level of uncertainty (since $\epsilon \leq \delta$) under the Assumption 1 and Assumption 2. ■

Next, we will propose a heuristic algorithm following the ideas used in the algorithm described in Section II and Theorem 1.

Algorithm 1: **Algorithm to obtain the minimal length of recursion in eq. (2):**

Step 1: Initialise $k=0$, $\gamma=0$. At each time step $2k$, we compute an a priori defined quantity called the *one step prediction error* $e_{2k}(t)$ (a function of t) to test whether the true minimal length of recursion has been reached or not. Generally speaking, the smaller the prediction error, the higher probability that $k = D_r+1$.

Step 2: For $i = 1, \cdots, n$, let $\gamma = \gamma_i \in \Gamma_r$. Then, compute the values of e_{2k} evaluated at different γ_i and use these to compute $\hat{E}_{2k} = \sum_{i=1}^n f(e_{2k}(\gamma_i))$, where $n \in \mathbb{N}$ is an a priori determined natural number and f is a prior determined function will be discussed later.

Step 3: Compute $\frac{\hat{E}_{2k-2}}{\hat{E}_{2k}}$ (when $k \geq 1$). If there exists a sudden drop in this quantity, i.e., $\frac{\hat{E}_{2k-2}}{\hat{E}_{2k}} \geq a$, this indicates that $k = D_r+1$ with a high probability based on the analysis above, (where a is an a priori determined constant, e.g., 10, which defines the *level of confidence*). When such a sudden drop occurs, the algorithm stops. Otherwise we update $k := k+1$.

Remark 7: The *overall one step prediction error* is defined as

$$E_{2k} = \int_{\gamma\in\Gamma_r} f(e_{2k}(\gamma)), \quad (16)$$

where f is a function which can be chosen, e.g., $f(x) = x^2$. In particular, E_{2k} can be approximated by a summation of the function values at different points $\{\gamma_1, \ldots, \gamma_n\}$, i.e., $\hat{E}_{2k} = \sum_{i=1}^n f(e_{2k}(\gamma_i))$ where $\hat{E}_{2k}$ represents the approximation of E_{2k}. We can use $\hat{E}_{2k}$ as a criterion to test if the minimal number of successive observations, $2D_r+3$, has been reached. The choice of e_{2k}, Γ_r, n, f, a and γ_i plays a key role in this method [18]. How to choose them cooperatively in order to get the best estimation goes beyond the scope of this paper.

In particular, at time step k, we form the Hankel matrix of successively observed outputs, i.e., $\boldsymbol{Y}(k,k)$ in eq. (10), and predict the estimated value at time $2k+1$ using the following equations

$$\begin{aligned} \boldsymbol{Y}(k,k)\hat{\boldsymbol{\alpha}} &= \begin{bmatrix} y_{k+1} & y_{k+2} & \cdots & y_{2k+1} \end{bmatrix}^T \\ \hat{x}_{2k+2}[r] &= \begin{bmatrix} y_{k+1} & y_{k+2} & \cdots & y_{2k+1} \end{bmatrix} \hat{\boldsymbol{\alpha}}^T, \end{aligned} \quad (17)$$

where $[\hat{\boldsymbol{\alpha}}^T, 1]^T$ is an estimated normalised kernel in eq. (8). At time $2k+1$, we can compute the error between the true one and the estimated one in eq. (17):

$$e_{2k+2} = y_{2k+2} - \hat{x}_{2k+2}[r] = \frac{\det \boldsymbol{Y}(k+1,k+1)}{\det \boldsymbol{Y}(k,k)}. \quad (18)$$

The second equality in (18) holds because e_{2k+2} is the Schur complement of $\boldsymbol{Y}(k,k)$ with respect to $\boldsymbol{Y}(k+1,k+1)$. We can also define this quantity data sets shifted by γ:

$$e_{2k+2}(\gamma) = \frac{\det(\boldsymbol{Y}(k+1,k+1) + \gamma \mathbf{1}\mathbf{1}^T)}{\det(\boldsymbol{Y}(k,k) + \gamma \mathbf{1}\mathbf{1}^T)}. \quad (19)$$

Remark 8: To define Γ_r in this particular case, we can not apply too large a shift as this will lead to computational errors in eq. (19). A practical way to select Γ_r is to take $\Gamma_r = [0,\ x_0[r]\delta]$ to keep it at a level similar to that of the uncertainty.

Remark 9: Indeed, there are numerous ways to define this one step prediction error. We here chose an easy-to-compute one. The computational complexity of our chosen one step prediction error method is $O(N^2)$ which is much lower than the computational complexity associated with other definitions such as, e.g., a least square definition which has a complexity $O(N^3)$ [18]. Therefore, with the same computational resources, using our one step prediction error method we can compute more values $\hat{E}_{2k}$ in Step 2 of Algorithm 1. This allows us to obtain a better approximation of E_{2k} given the same computational resources.

B. Estimating data sets

After having successfully obtained the true minimal number of steps in the previous section, the final objective is to propose a method to estimate the difference between the estimated final value $\hat{\phi}$ and the exact final value ϕ. We showed that $2D_r+3$ successive values of the output y were needed to compute the coefficients of $q_r(t)$ or equivalently to compute the coefficients of the minimal polynomial of $[A, \boldsymbol{e}_r^T]$.

The objective pursued here is to obtain a second data set "close" (in terms of some norm) to the original one and to compute the final value corresponding to this data set to see how "close" the corresponding final value is with respect to the original one. First, we propose a possible stable data set "close" to the original data set, compute the corresponding final value, and characterise how this estimated data set differs from other possible data sets and how the estimated final value differs from the one computed from these other possible data sets. In the rest of this section, we will focus on the first step, while the second step will be considered in the next section. Recall that Assumption 1 requires that the estimated data set has the following properties:

1) It is "close" to the original data set $Y = \{y_0,\ y_1,\ \cdots,\ y_{2D_r+2}\}$;

2) The length of recursion is fixed as D_r+1 in eq. (3);

3) The estimated associate minimal polynomial $q_r(t)$ defined in Definition 5 should have one and only one root at 1 and all other roots within the unit circle.

Algorithm 2: **Algorithm to obtain a close and stable output data set:**

The first step is to change the output data set to $\{y_0 + \gamma',\ y_1+\gamma',\ \cdots,\ y_{2D_r+1}+\gamma',\ y_{2D_r+2}+\gamma' - e_{2D_r+2}(\gamma')\}$ where $\gamma' = \text{argmin}_{\gamma\in\Gamma_r}\{e_{2D_r+2}(\gamma)\}$. This data set satisfies the recursion in eq. (2) with a length of recursion D_r+1. In this case, we observe that the corresponding Hankel matrix to this data set (defined in eq. (10)), i.e., $\boldsymbol{Y}(D_r+1, D_r+1)$, is not full rank and thus we can compute the normalised kernel of this defective Hankel matrix $\boldsymbol{\alpha} = \begin{bmatrix}\alpha_0 & \alpha_1 & \dots & \alpha_{D_r} & 1\end{bmatrix}^T$, i.e.,

$$\boldsymbol{Y}(D_r+1, D_r+1)\boldsymbol{\alpha} = 0.$$

The second step is to compute the roots of $t^{D_r+1} + \alpha_{D_r}t^{D_r} + \ldots + \alpha_0 = 0$, namely, $\lambda_1, \ldots, \lambda_{D_r+1}$ (for simplicity, we assume that there are no repeated roots and no root at 1 since generally these happen with probability 0). Assume the ordering of the eigenvalues has been made such that $\lambda_1 \in \mathbb{R}$ satisfies $|\lambda_1 - 1| \le |\lambda_i - 1|$ for all $i \in \{1, 2, \cdots, D_r+1\}$. We then we construct a Vandermonde matrix

$$V(0, D_r) = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & \lambda_2 & \cdots & \lambda_{D_r+1} \\ \vdots & \ddots & \ddots & \vdots \\ 1 & \lambda_2^{D_r} & \cdots & \lambda_{D_r+1}^{D_r} \end{bmatrix}$$

and, without loss of generality, rearrange the columns of V such that $\lambda_1 = 1$, $\{|\lambda_i| \le 1 : i = 2, \cdots, k\}$ and $\{|\lambda_i| > 1 : i = k+1, \cdots, D_r+1\}$. This step guarantees that $q_r(t)$ has only one root processing at 1.

The third step is to compute the diagonal matrix T

$$\begin{aligned} T &\triangleq \text{diag}\{(V(0,D_r)^T)^{-1}\boldsymbol{Y}(D_r,D_r)(V(0,D_r))^{-1}\} \\ &\triangleq \begin{bmatrix} T_1 & 0 \\ 0 & T_2 \end{bmatrix}, \end{aligned}$$

where $T_1 \in \mathbb{R}^{k\times k}$ and $T_2 \in \mathbb{R}^{(D_r+1-k)\times(D_r+1-k)}$. We denote $\boldsymbol{H} = V(0,D_r)TV(0,D_r)^T$. This step approximates $\boldsymbol{Y}(D_r+1, D_r+1)$ to $\boldsymbol{H}$ by imposing the constraint that $\boldsymbol{H}$ should be Hankel.

The fourth step is to find a stable approximation of $\boldsymbol{H}$, i.e., another Hankel matrix $\boldsymbol{H}' = V'(0,D_r)T'V'(0,D_r)^T$ such that $\|\boldsymbol{H} - \boldsymbol{H}'\|$ is minimal, where $\|\cdot\|$ is some norm, e.g., the Frobenius norm and

$$V'(0, D_r) = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & \lambda_2' & \cdots & \lambda_{D_r+1}' \\ \vdots & \ddots & \ddots & \vdots \\ 1 & \lambda_2'^{D_r} & \cdots & \lambda_{D_r+1}'^{D_r} \end{bmatrix}$$

having all λ_i' within the unit circle. After some manipulation, we have

$$\|\boldsymbol{H} - \boldsymbol{H}'\| = \|V_1T_1V_1^T + V_2T_2V_2^T - V_1'T_1'V_1'^T - V_2'T_2'V_2'^T\|.$$

For the stable part, we choose $V_1' = V_1$ and $T_1' = T_1$, while for the unstable part, we can use well-established Nehari's theorem (see for example Section 8 in [4]) to find the closest stable approximation in terms of the $\mathcal{L}_\infty$ norm.

Finally, since $\boldsymbol{H}'$ is a Hankel matrix, we can find out the corresponding data set, namely, $Y' = \{y'_0, \ldots, y'_{2D_r}\}$ and the corresponding estimated normalised kernel $\hat{\boldsymbol{\alpha}}$. We then shift all the output back to a data set $\hat{Y} = \{y'_0 - \gamma', \ldots, y'_{2D_r} - \gamma'\}$. Therefore, we can compute the final value of this data set $\hat{\phi}$ from eq. (4) and take this value as an approximation for $\phi = \lim_k x_k[r]$.

Definition 7: Given the original data set obtained from system (9), i.e., $Y = \{y_0,\ y_1,\ \ldots, y_{2D_r}\}$, and assuming that $Y_1 = \{y_0^1,\ y_1^1,\ \ldots,\ y_{2D_r}^1\}$ is an estimated stable output data sets associated with a minimal polynomial $q_r^1(t)$ of degree $D_r + 1$, we define the *data distance* between the estimated output data set and the original data set as

$$d(Y_1, Y) = \|\boldsymbol{H^1} - \boldsymbol{H}\|_F = \sum_{k=0}^{2D_r} (y_k^1 - y_k)^2. \qquad (20)$$

Remark 10: Since finding the optimal solution satisfying the above mentioned conditions is NP hard, we are trying to meet the requirements step by step under the constraint that we only make small changes to the data set guided by the measurement of some norm. It is clear that the algorithm proposed here is not optimal and might produce a large data distance according to Definition 7. Using the approach described in [12] to obtain optimal bounds for the estimated state so as to minimise $d(\hat{Y}, Y)$ in eq. (20) is under investigation.

C. Characterising the final value error

In terms of how the approximation is affecting the computed final value, we look at the outputs obtained after the procedure described in Algorithm 2 of Section IV-B has been applied. Notice that there are many ways to obtain estimated output data sets. Since our solution is sub-optimal, one may easily obtain another estimated set of outputs from other sub-optimal procedures. In this section, we will quantify the difference in terms of both the estimated outputs and the computed final values obtained by different methods.

The data distance between two data set Y_1 and Y_2 (e.g., obtained by different methods) can be characterised by the following triangle inequality

$$|d(Y_1, Y) - d(Y_2, Y)| \le d(Y_1, Y_2) \le d(Y_1, Y) + d(Y_2, Y).$$

When it comes to the difference between estimated final values from different data sets, we are interested in those that are "close" in terms of data distance but different in terms of the associated minimal polynomial.

Assumption 3: Consider two data sets obtained by different methods, namely $Y_i = \{y_0^i, \ldots, y_{2D_r}^i\}$, $i = 1, 2$ with corresponding minimal polynomials $q_r^i(t)$ and final values ϕ_i. Assume that these data sets are such that $d(Y_1, Y_2) \le \epsilon$ with ϵ "small". Assume that these two minimal polynomials of degree $D_r + 1$ are such that $q_r^1(t) = (t - \lambda_j)s_r(t)$ and $q_r^2(t) = (t - \lambda'_j)s_r(t)$ for some $\lambda_j \in \mathbb{R}$, some $\lambda'_j \in \mathbb{R}$, and some polynomial $s_r(t)$.

Instead of directly characterising the final value error, i.e., $\phi_1 - \phi_2$, we use an indirect method as described in Fig. 4, i.e., we build virtual data sets $\hat{Y}_i = \{y_0^i, \ldots, y_{D_r-1}^i\}$

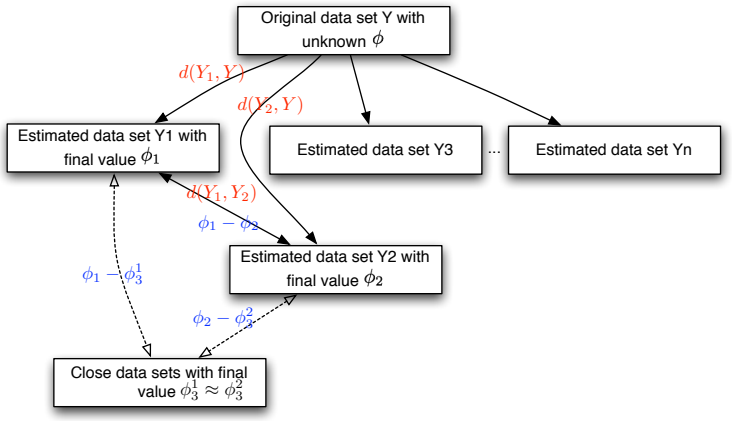

Fig. 4. Different output data sets and corresponding final values, and their associated distances.

which contain the first D_r elements of Y_i and for which the corresponding minimal polynomial is $s_r(t)$. The final value associated with the virtual data sets $\hat{Y}_i$ can be computed as

$$\phi_3^i = \frac{\begin{bmatrix} y_{D_r-1}^i & y_{D_r-2}^i & \ldots & y_0^i \end{bmatrix} P}{p_r(1)}, \qquad (21)$$

in which $P = \begin{bmatrix} 1 & \beta_{D_r-2} & \ldots & \beta_0 \end{bmatrix}^T$ and $p_r(t) = \frac{s_r(t)}{(t-1)} \triangleq t^{D_r-1} + \beta_{D_r-2}t^{D_r-2} + \ldots + \beta_0$. We then characterise the final value difference between ϕ_i and ϕ_3^i separately.

Theorem 2: Consider an estimated output data set $Y_1 = \{y_0^1, \ldots, y_{2D_r}^1\}$ (with associated minimal polynomial $q_r^1(t)$ and final value ϕ_1) and a virtual data set $\hat{Y}_1$ with final value ϕ_3^1 computed in eq. (21). The difference between the computed final values is $\phi_1 - \phi_3^1 = -\mu_j \frac{p_r(\lambda_j)}{p_r(1)}$,where $p_r(t) = \frac{q_r^1(t)}{(t-\lambda_j)(t-1)}$.

Proof: Let $c_r(t) = t^{D_r} + \eta_{D_r-1}t^{D_r-1} + \ldots + \eta_0$. It is easy to show that the minimal polynomial $q_r^1(t)$ satisfies $q_r^1(t) = (t-1)c_r(t) = (t-1)(t-\lambda_j)p_r(t),\ \forall t$. We have $\phi_1 = \frac{\begin{bmatrix} y_{D_r} & y_{D_r-1} & \ldots & y_0 \end{bmatrix} C}{c_r(1)}$, in which $C = \begin{bmatrix} 1 & \eta_{D_r-1} & \ldots & \eta_0 \end{bmatrix}^T$. Noting that $p_r(t) = c_r(t)/(t-\lambda_j)$, we obtain $C^T = \begin{bmatrix} P^T & 0 \end{bmatrix}^T + \lambda_j \begin{bmatrix} 0 & P^T \end{bmatrix}^T$. Therefore, the difference in the final value is given by

$$\begin{aligned}
\phi_1 - \phi_3^1 &= \frac{\begin{bmatrix} y_{D_r} & \ldots & y_0 \end{bmatrix} C}{c_r(1)} - \frac{\begin{bmatrix} y_{D_r-1} & \ldots & y_0 \end{bmatrix} P}{p_r(1)} \\
&= \frac{\begin{bmatrix} y_{D_r} & \ldots & y_1 \end{bmatrix} P}{c_r(1)} - \frac{\lambda_j \begin{bmatrix} y_{D_r-1} & \ldots & y_0 \end{bmatrix} P}{c_r(1)} \\
&\quad - \frac{(1-\lambda_j)\begin{bmatrix} y_{D_r-1} & \ldots & y_0 \end{bmatrix} P}{c_r(1)} \\
&= \frac{\begin{bmatrix} y_{D_r} & \ldots & y_1 \end{bmatrix} P}{c_r(1)} - \frac{\begin{bmatrix} y_{D_r-1} & \ldots & y_0 \end{bmatrix} P}{c_r(1)} \\
&= \mu_j \frac{(\lambda_j - 1)p_r(\lambda_j)}{p_r(1)(1-\lambda_j)} = -\mu_j \frac{p_r(\lambda_j)}{p_r(1)}. \qquad (22)
\end{aligned}$$

To show eq. (22) holds, we consider that for all roots of $c_r(t) = 0$ other than λ_j, say λ_i, $p_r(\lambda_i) = 0$. Therefore,

from eq. (3) and eq. (12), these roots λ_i will be eliminated as a result of multiplying $\begin{bmatrix} y_{D_r} & y_{D_r-1} & \dots & y_1 \end{bmatrix}$ and $\begin{bmatrix} y_{D_r-1} & y_{D_r-2} & \dots & y_0 \end{bmatrix}$ by P. This completes the proof. ■

Remark 11: Under Assumption 3, the difference between the final values is $\phi_1 - \phi_2 \approx (\phi_1 - \phi_3^1) - (\phi_2 - \phi_3^2)$. Both terms in brackets can be computed using Theorem 2 and $\phi_3^1 \approx \phi_3^2$ when $d(Y_1, Y_2) \leq \epsilon$.

Remark 12: Consider "close" output data sets without many common terms in their minimal polynomials, the procedure above can be applied several times recursively to obtain both the data distance (defined in Definition 7) and the difference between the estimated final values associated with data sets estimated from different subsets.

V. Final value for systems with time-delays

Another important aspect that generally needs to be taken into account is time-delays. This occurs in the communication channels, and we need to characterise how this affects the minimal number of steps needed to compute the final value for a single node. We hereby consider a DTLTI system with multiple delays, and assume the all these delays take integer values which are upper-bounded by, say, $\tau \in \mathbb{N}$. The corresponding model writes

$$x_{k+1}[i] = \sum_{j \in \{1,2,\cdots,N\}} a_{ij}(k) x_{k-\tau_{ij}}[j], \tag{23}$$

with $\tau_{ij} \in \mathbb{N}$ and $0 \leq \tau_{ij} \leq \tau$, $\forall i, j$. The corresponding system dynamics is given by:

$$x_{k+1} = A_1 x_k + A_2 x_{k-1} + \ldots + A_\tau x_{k+1-\tau}, \tag{24}$$

in which $A_1,\ A_2, \ldots,\ A_\tau \in \mathbb{R}^{N \times N}$ are the transition matrices corresponding to different delays.

Without loss of generality, we take one observation of this system, i.e., $y_k = \boldsymbol{e}_r^T x_k$. We rewrite the state-space equation as

$$\psi_{k+1} = \Xi \psi_k, \tag{25}$$

in which, $\psi_k = [x_k^T, x_{k-1}^T, \cdots, x_{k+1-\tau}^T]^T$, and

$$\Xi = \begin{pmatrix} A_1 & A_2 & \dots & A_{\tau-1} & A_\tau \\ I_N & & & & \\ & I_N & & & \\ & & \ddots & & \\ & & & I_N & \end{pmatrix},$$

where I_N is the identity matrix of dimension N. Denote by $g(z)$ the minimal polynomial of Ξ (see definition 2). As proven in [2], the minimal polynomial of a matrix divides its characteristic polynomial. Therefore, by direct calculation, $\det(zI - \Xi) = \det\left(z^\tau I - A_1 z^{\tau-1} - \ldots - A_{\tau-1} z - A_\tau\right)$. As a consequence, the degree of $g(z)$ is at most $\tau \times N$. Developments similar to those presented previously yield the following result.

Corollary 1: Consider the system in equation (23). Any arbitrarily chosen state y can compute its corresponding final value in finite time using at most $2\tau N + 1$ successive values of itself.

Proof: Similar to the ones in the preceding section. The only difference is that here we have a higher dimension transition matrix Ξ. ■

VI. Discussion and conclusion

Motivated by the finite and minimal time consensus problems [2], [6], we observe that, if every node can observe its own state perfectly, some nodes in the network require less steps than others to compute the final consensus value of the whole network. When imperfect communication and/or uncertain observation are taken into account, the question of knowing whether this property still holds is extremely important in practice. In particular, if such a property holds, what is the corresponding an optimal strategy to compute the final value?

In this paper, a first attempt to obtain a sub-optimal solution to this challenging problem is proposed since finding an optimal solution is NP hard. These results have important applications in the analysis of social network, minimal-time decentralised network consensus problem, prediction of stock values and many others.

References

[1] R. Horn and C. Johnson, *Matrix analysis*. Cambridge University Press 1999.

[2] S. Sundaram and C. N. Hadjicostis, "Finite-Time Distributed Consensus in Graphs with Time-Invariant Topologies," Proceedings of the American Control Conference, 2007.

[3] D. Boley, F. Luk and D. Vandevoorde, "Vandermonde Factorization of a Hankel Matrix," Scientific computing, pp. 27-39, Springer, 1997.

[4] K. Zhou, J. Doyle and K. Glover, "*Robust and Optimal Control*," Prentice Hall, 1996.

[5] E. I. Jury, "*Inners and stability of dynamic systems*," John Wiley and Sons, 1974.

[6] Y. Yuan, G. Stan, L. Shi and J. Gonçalves, "Decentralised final value theorem for discrete-time LTI systems with application to minimal-time distributed consensus," Proceedings of Conference on Decision and Control, 2009.

[7] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," IEEE Transactions on Automatic Control, vol. 49, no. 9, pp. 1520–1533, 2004.

[8] C. Ko and L. Shi, "Scheduling for Finite Consensus," In Proceedings of the American Control Conference, 2009.

[9] A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," IEEE Transactions on Automatic Control, vol. 48, no. 6, 2003.

[10] W. Ren and R. W. Beard, *Distributed Consensus in Multi-vehicle Cooperative Control: Theory and Applications*, Springer, 2007.

[11] W. Shuang and L. Qiu, "Stabilization of Networked Control Systems with Finite Data Rate," 2009 Asian Control Conference, 2009.

[12] Z. Zhang and I. Jaimoukha, "Fault detection and isolation for linear discrete-time systems using output measurement analysis," Proceedings of Conference on Decision and Control, 2009.

[13] C. Godsil and G. Royal, *Algebraic graph theory*, New York: Springer-Verlag, 2001.

[14] S. Beheshti and M. Dahleh, "Noisy data and impulse response estimation," IEEE Transaction on Signal Processing, vol. 58, no. 2, 2010.

[15] M. Fazel, "Matrix rank minimisation with applications," Ph.D. thesis, Stanford University, 2002.

[16] A. Megretski and J. Wyatt, *Linear algebra and functional analysis for signal and systems*, Lecture note, EECS, MIT.

[17] S. Mason and H. Zimmermann, *Electronic circuits, signals and systems.* John Wiley and Sons, INC.

[18] L. Ljung. *System Identification–Theory for the User*. Prentice Hall, 1999.

[19] K. Glover and L. Silverman "Characterisation of structural controllability," IEEE Transactions on Automatic Control, 21 (4). pp. 534-537, 1976.