

COMPARISON OF THE APPROXIMATION METHODS FOR TIME-DELAY SYSTEMS: APPLICATION TO MULTI-AGENT SYSTEMS

ÁRON FEHÉR^{*1,2} AND LŐRINC MÁRTON¹

¹Department of Electrical Engineering, Sapientia Hungarian University of Transylvania, C.1., Târgu Mureș, 547367, ROMANIA

²Department of Mathematics, University of Pannonia, Egyetem u. 10, Veszprém, 8200, HUNGARY

This paper presents a review of dominant pole and model-approximation algorithms for delayed systems that can be applied to multi-agent systems. A novel algorithm is proposed to determine an approximation method for multi-agent systems in the platoon configuration with a communication delay. Simulations are presented to show the applicability of the proposed algorithm.

Keywords: time delay, differential equations, asymptotic properties, multi-agent systems

1. Introduction

A Multi-Agent System (MAS) consists of multiple active agents (e.g. vehicles), passive agents (e.g. obstacles), in addition to cognitive agents and their environment [1]. Every active agent is at least partially autonomous and uses a distributed control algorithm [2]. The agents can communicate with each other. This communication structure is defined by a graph, where each vertex corresponds to one agent and each edge to a communication direction.

A platoon is a special MAS configuration, with a linear communication graph. The leading (first) agent implements a reference tracking algorithm. Every other agent implements a consensus with the adjacent agents [3].

With the increase in the number of agents and the physical distance between the agents, the communication delay cannot be neglected. While the stability of such systems is ensured by the consensus protocol [4], the delay will influence the transient behavior of the MAS [5].

The goal of this paper is to compare the existing approximation methods for the transient behavior analysis of MAS with communication delays and present a novel analysis method which can be applied to any MAS system which satisfies a smallness delay condition.

2. Modelling of MAS

A MAS is considered with agents that exhibit single-integrator dynamics [6]. The state-space model of an agent becomes $\dot{x}_i(t) = u_i(t)$, where $x_i \in \mathbb{R}$ denotes

the state of the i th agent and $u_i \in \mathbb{R}$ represents the input, $i = 1, 2, \dots, n$.

A MAS has an underlying communication graph, in which the vertex is an agent and the edge a communication path [7], so the agent i th in the system is the vertex v_i . Let N_i be the set of neighbors of v_i , so that N_i contains all vertices that are connected to v_i .

Consensus algorithm The consensus problem of a MAS is the procedure of gathering every state from the initial condition to a common steady-state. If the communication graph is connected, the consensus with regard to an agent can be reached with the input (*consensus protocol*)

$$u_i(t) = \sum_{j \in N_i} (x_j(t) - x_i(t)). \quad (1)$$

The adjacency matrix $A = (a_{ij}) \in \mathbb{R}^{n \times n}$ of a graph with n nodes is defined as

$$a_{ij} := \begin{cases} 1, & \text{if } i \neq j \text{ and } v_i \text{ are adjacent to } v_j \\ 0, & \text{otherwise} \end{cases}. \quad (2)$$

The degree matrix $D = (d_{ij}) \in \mathbb{R}^{n \times n}$ of a graph with n nodes shows the number of neighbors for each vertex and can be defined as

$$d_{ij} := \begin{cases} \deg(v_i), & \text{if } i = j \\ 0, & \text{otherwise} \end{cases}, \quad (3)$$

where $\deg(v_i)$ denotes the degree or the number of edges incident to vertex i .

*Correspondence: fehera@ms.sapientia.ro

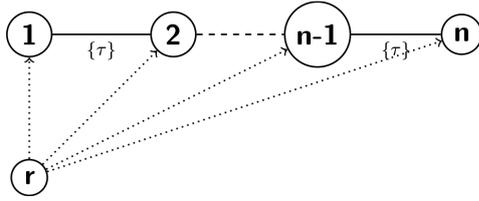


Figure 1: The communication topology of a vehicle platoon system consisting of n vehicles with time delay τ in the communication graph. The dashed line symbolizes more nodes in between, while the dotted line represents the reference input of the nodes.

With this notation the dynamics of the MAS with the consensus protocol is given by

$$\dot{\underline{x}}(t) = -L\underline{x}(t), \quad \underline{x}(0) = \underline{x}_0, \quad (4)$$

where L denotes the Laplacian matrix [8] which is constructed as $L = D - A$, D stands for the degree matrix, A represents the adjacency matrix of the graph, $\underline{x} = (x_1 \ x_2 \ \dots \ x_n)^\top \in \mathbb{R}^n$ denotes the state vector consisting of the n states of the MAS, and $\underline{x}_0 \in \mathbb{R}^n$ is a constant vector of the initial states.

According to [9] the eigenvalues of a MAS consisting of n agents with a connected communication graph can be ordered as

$$0 = \lambda_1 > \lambda_2 \geq \dots \geq \lambda_n. \quad (5)$$

The steady states (equilibria) of the MAS \underline{x}_{ss} are the elements of the null space of L . Given, according to the definition of the Laplacian matrix, $\sum_{j \in N_i} l_{ij} = 0$ [6] for every solution \underline{x} of (Eq. 4): $\lim_{t \rightarrow \infty} \underline{x}(t) = \underline{x}_{ss} = \frac{1}{n} \sum_{i=1}^n x_i(0) \mathbf{1}$, where $\mathbf{1} = (1 \ 1 \ \dots \ 1)^\top \in \mathbb{R}^n$.

MAS with delayed communication In the case of delayed communication, the consensus protocol is of the form

$$u_i(t) = \sum_{j \in N_i} (x_j(t - \tau) - x_i(t)), \quad (6)$$

where $\tau \geq 0$ denotes the constant communication delay which is present among adjacent agents.

According to Refs. [4] and [10], the MAS with communication delay (also referred to as Multi-Agent System with Delays (DMAS)) is

$$\begin{aligned} \dot{\underline{x}}(t) &= -D\underline{x}(t) + A\underline{x}(t - \tau) + R, \\ \underline{x}(\theta) &= \underline{x}_0 \in \mathbb{R}^n, \quad \theta \in [-\tau, 0], \end{aligned} \quad (7)$$

3. Platoon of vehicles

A platoon of vehicles is a special class of MAS with a communication topology shown in Fig. 1. The kinematic model of a vehicle can be written as

$$\dot{x}_i(t) = u_i(t), \quad (8)$$

where $x_i(t)$ denotes the position of the vehicle and $u_i(t)$ represents the control signal in the form

$$\begin{cases} u_1(t) = k_{p1}(r - x_1(t)) \\ u_i(t) = \xi_i(x_{i-1}(t - \tau) - d - x_i(t)), \quad \forall i = 2, \dots, n \end{cases} \quad (9)$$

where $i \in (1, n)$, $n \in \mathbb{N}$, $n > 1$, d denotes the prescribed inter-vehicle distance, $\xi_i > 0$, and $k_{p1} > 0$ are constant control gains. r is the position reference of the first vehicle.

Eqs. 8 and 9 can be written as a system of differential equations with delay:

$$\dot{\underline{x}}(t) = -\Phi\underline{x}(t) + \Gamma\underline{x}(t - \tau) + \underline{f}_1 r - \underline{f}_2 d, \quad (10)$$

where

$$\Phi = \text{diag}\{(k_{p1} \ \xi_2 \ \dots \ \xi_{n-1} \ \xi_n)\} \quad (11)$$

denotes the degree matrix with the state feedback,

$$\Gamma = \begin{pmatrix} 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ \xi_2 & 0 & 0 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & \xi_{n-1} & 0 & 0 \\ 0 & 0 & 0 & \dots & 0 & \xi_n & 0 \end{pmatrix} \quad (12)$$

represents the adjacency matrix, and

$$\underline{f}_1 = (k_{p1} \ 0 \ \dots \ 0)^\top, \quad (13)$$

$$\underline{f}_2 = (0 \ \xi_2 \ \dots \ (n-1)\xi_n)^\top. \quad (14)$$

The homogeneous part of the relation in Eq. 10 is of the same form as the relation in Eq. 7, and the term $\underline{f}_1 r - \underline{f}_2 d$ represents the reference as well as inter-vehicle distance induced inflows.

4. Existing methods for the approximation of time-delay systems

In this section, the various current approximation methods for time-delay systems are reviewed. The form of the studied systems is shown by the following relation:

$$\dot{\underline{x}}(t) = -\Phi\underline{x}(t) + \Gamma\underline{x}(t - \tau), \quad \underline{x}(h) = x_0, \quad (15)$$

which is the homogeneous part of Eq. 10, where $\theta(h)$ denotes the initial condition with $h \in [-\tau, 0]$, and has a quasi-polynomial characteristic equation:

$$\lambda I_n + \Phi - \Gamma e^{-\tau\lambda} = 0, \quad (16)$$

where the delay component induces an exponential term.

The roots of Eq. 16 determine the transient behaviour of the system in Eq. 15. As Eq. 15 possesses an infinite number of solutions, it is important to develop such an equivalent system which has a finite number of eigenvalues and is a good approximation of the original system in Eq. 15. If a good delay-free approximation is available, it

could be applied to the transient behaviour analysis and control design for delayed systems.

Let a special solution be denoted by \tilde{x} , which is uniquely determined by the value $\tilde{x}(0)$, and independent of $\theta(h)$, $h \in [-\tau, 0)$, thus forming an n parameter family. In a linear autonomous system, this corresponds to the eigensolution generated by exactly n characteristic roots (multiplicities included) which lies in the half-plane $\text{Re } \lambda > -1/\tau$, see Ref. [11].

Theorem 1. [11] Consider a Delay Differential Equation (DDE) system of the form shown in relation Eq. 15 with a Lipschitz criterion of:

$$(\|\Phi\| + \|\Gamma\|)\tau e < 1, \quad (17)$$

further noted as **smallness condition**, for every solution \underline{x} of Eq. 15 a globally defined solution $\tilde{x} : \mathbb{R} \rightarrow \mathbb{R}^n$ exists that satisfies the growth condition $\sup_{t \leq 0} \|\tilde{x}(t)\| e^{t/\tau} < \infty$ such that

$$\|x(t) - \tilde{x}(t)\| \rightarrow 0 \quad \text{exponentially as } t \rightarrow \infty.$$

For further related discussions, see Refs. [12] and [13]. The aforementioned theorem yields n dominant eigenvalues which can accurately represent a DDE system. The system shown in the relation in Eq. 15 uses the consensus protocol, which ensures that the eigenvalues are located in the left half-plane in the complex region. The final location of the dominant eigenvalues is created as a semi-circle with origin 0, radius $1/\tau$, and a negative real part.

4.1 The modified chain approximation

The modified chain approximation method creates an approximating system directly from the state-space representation of the delayed system [14]. For a DDE in the form of Eq. 15, the modified chain approximation method yields an approximating linear system in the form of

$$\begin{aligned} \dot{\underline{y}}_0(t) &= -\Phi \underline{y}_0(t) + \frac{m}{\tau} I_n \underline{y}_m(t) \\ \dot{\underline{y}}_1(t) &= \Gamma \underline{y}_0(t) - \frac{m}{\tau} I_n \underline{y}_1(t) \\ &\vdots \\ \dot{\underline{y}}_k(t) &= \frac{m}{\tau} I_n \underline{y}_{k-1}(t) - \frac{m}{\tau} I_n \underline{y}_k(t), \quad 2 \leq k \leq m \end{aligned} \quad (18)$$

The output vector $\underline{z} = \underline{y}_0$ represents the approximation of the solution \underline{x} of the relation in Eq. 15. The approximating system in the form of a matrix is shown in

$$\dot{\underline{y}}(t) = G \underline{y}(t) \quad (19)$$

with

$$G = \begin{pmatrix} -\Phi & 0_n & 0_n & \cdots & 0_n & \frac{m}{\tau} I_n \\ \Gamma & -\frac{m}{\tau} I_n & 0_n & \cdots & 0_n & 0_n \\ 0_n & \frac{m}{\tau} I_n & -\frac{m}{\tau} I_n & \cdots & 0_n & 0_n \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0_n & 0_n & 0_n & \cdots & \frac{m}{\tau} I_n & -\frac{m}{\tau} I_n \end{pmatrix} \quad (20)$$

where $\underline{y} \in \mathbb{R}^{mn}$ denotes the state vector, $G \in \mathbb{R}^{(m+1)n \times (m+1)n}$ represents the matrices of the system, $0_n \in \mathbb{R}^{n \times n}$ stands for the zero matrix, n is the number of agents and m denotes the number of approximating equations.

According to Ref. [14] the output of the system (Eq. 19) defined as $\underline{z}(t) = \underline{y}_0(t)$ linearly converges into the solution of the original DDE system such that $c > 0$ and $\sup_{t \geq 0} \|\underline{x}(t) - \underline{z}(t)\| \leq \frac{c}{m}$.

4.2 The Lambert W function

The Lambert W function can be used to find the dominant eigenvalues of a quasi-polynomial equation (Eq. 16). Every $W(s)$ function that satisfies

$$W(s)e^{W(s)} = s \quad (21)$$

by definition is referred to as a Lambert W function [15], where s is either a scalar or matrix complex number function. The Lambert W function has multiple branches denoted as $W_k(s)$ with $k = 0, \pm 1, \pm 2, \dots, \pm \infty$.

Example 4.1. If a scalar DDE is present in the form of

$$\dot{x}(t) = ax(t) + bx(t - \tau) + cu(t), \quad (22)$$

with $a, b, c, \theta \in \mathbb{R}$, $x(h) = \theta(h)$ for $h \in [-\tau, 0]$, the quasi-polynomial of the homogeneous part can be written as

$$(\lambda - a)e^{\tau\lambda} = b. \quad (23)$$

If both sides are multiplied by $\tau e^{-a\tau}$, then

$$(\lambda - a)\tau e^{\tau(\lambda - a)} = b\tau e^{-a\tau}, \quad (24)$$

which satisfies Eq. 21 with $W(b\tau e^{-a\tau}) = (\lambda - a)\tau$, and the eigenvalues can be calculated using the branches of the Lambert W function as

$$\lambda_k = \frac{1}{\tau} W_k(b\tau e^{-a\tau}) + a. \quad (25)$$

In our case, in terms of the relation in Eq. 15, the aforementioned solution is generalized as

$$\underline{\lambda}_k = \frac{1}{\tau} W_k(\Gamma \tau Q_k) - \Phi, \quad (26)$$

where Q_k can be calculated by solving the equation numerically

$$W_k(\Gamma\tau Q_k)e^{W_k(\Gamma\tau Q_k)-\Phi\tau} = \Gamma\tau \quad (27)$$

for Q_k [16].

Although many numerical solvers provide native support for the solution of the scalar Lambert W function, the general case requires additional solver tools. Therefore, the LambertWDDE Toolbox [17] was created. The function `find_Sk` assumes τ , Γ and $-\Phi$. The returned values are the eigenvalues λ_k and the Q_k parameters for a given k branch. The toolbox can create the approximating solution for the given system as

$$\tilde{x}(t) = \sum_{k=m_1}^{m_2} e^{\lambda_k t} C_k^I, \quad (28)$$

where \tilde{x} is the approximating solution of the original delay system and the parameter C_k^I can be computed with the help of `find_CI` for a given $m_1 < k < m_2$ branch. In the scalar case, C_k^I takes the form of

$$C_k^I = \frac{x_0 + be^{-\lambda_k \tau} \int_0^\tau \theta(t - \tau) dt}{1 + b\tau e^{-\lambda_k \tau}}. \quad (29)$$

4.3 The Quasi-polynomial root-finder algorithm

The quasi-polynomial root-finder algorithm calculates the dominant eigenvalues of a system based directly on the quasi-polynomial equation in Eq. 16.

The quasi-polynomial equation of the system in Eq. 16 can be written as:

$$P(\lambda) = \sum_{k=0}^N Q_k(\lambda) e^{-\alpha_k \tau \lambda}, \quad (30)$$

where Q_k is a polynomial with real coefficients and $\alpha_k \in \mathbb{R}$. The objective is to compute the spectrum in the region of the complex plane $\mathbb{D} \subset \mathbb{C}$ with boundaries $\beta_{\min} < \text{Re}(\mathbb{D}) < \beta_{\max}$ and $\omega_{\min} < \text{Im}(\mathbb{D}) < \omega_{\max}$.

Let the surfaces defined by the real and imaginary parts of $P(\lambda)$ be:

$$\text{Re}(P(\beta, \omega)) = 0 \quad (31)$$

$$\text{Im}(P(\beta, \omega)) = 0 \quad (32)$$

The eigenvalues can be located at the points of intersection of the zero-level curves of the surfaces $\text{Re}(P(\beta, \omega)) = 0$ and $\text{Im}(P(\beta, \omega)) = 0$ as shown in Ref. [18]. The accuracy of the algorithm is increased by Newton's method and by adapting the grid density of \mathbb{D} as shown in Ref. [16].

The Quasi-polynomial root-finder algorithm (QPmR) [19] is implemented in MATLAB. The function expects the region of interest $[\beta_{\min}, \beta_{\max}, \omega_{\min}, \omega_{\max}]$ to be in the complex plane of the polynomial coefficient matrix of the quasi-polynomial where one row corresponds to

one polynomial multiplied by the same exponential term. The delay vector, computational accuracy and grid step are also required.

In our case this translates into a region of interest $[-\frac{1}{\tau}, 0, -\frac{1}{\tau}, \frac{1}{\tau}]$ if the smallness condition (Eq. 17) is satisfied. The first row of the matrix of polynomial coefficients contains the coefficients of the delay-free part so that the delay vector is of the form $[0, \tau, 2\tau, \dots]$.

The algorithm covers the given region with a mesh grid, then evaluates the quasi-polynomial at each point of the grid by splitting it into a real and an imaginary part. The zero-level curves are then mapped with the help of the `contour` plotting algorithm. The computational error is checked and if it is too large, the algorithm is restarted using a modified grid density as described in Ref. [16]. If the computational error is smaller than the given level of tolerance, the computed dominant eigenvalues are returned.

5. Explicit matrix approximation method

An approximation method was devised where the convergence rate is exponential, the degree of the resulting system in the form of Eq. 4 matches exactly the degree of the delayed MAS given in Eq. 15, and the same properties are exhibited in specific cases.

The Banach fixed-point theorem was used as discussed in Ref. [20] to explicitly find a linear system of the form of Eq. 4 which approximates the homogeneous part of the system in Eq. 10.

If an (X, f) metric space is present and $T : B \rightarrow B$ is a contraction with a bounded set $B \subset X$, and $q < 1$ such that

$$f(T(x), T(y)) \leq qf(x, y) \quad (33)$$

by definition T admits a unique fixed point \tilde{x} such as $T(\tilde{x}) = \tilde{x}$, and this fixed point can be found by starting from an arbitrary element $x_0 \in B$ with the sequence

$$x_n = T(x_{n-1}), \quad (34)$$

where $x_n \rightarrow \tilde{x}$.

A DDE system is defined in Eq. 15 by the corresponding smallness condition of Eq. 16. Since all normed spaces are metric spaces, the metric space $X = \mathbb{R}^{n \times n}$ is set with f as the induced matrix norm. The contraction $T : B \rightarrow B$ is present such that

$$T(\Lambda) = -\Phi + \Gamma e^{-\Lambda\tau}, \quad (35)$$

and $B = \{\Lambda \in \mathbb{R}^{n \times n} \mid \|\Lambda\| \leq (\|\Phi\| + \|\Gamma\|)e\}$. The relation in Eq. 33 holds true for $(\|\Phi\| + \|\Gamma\|)\tau e < 1$.

Let $\Lambda_1, \Lambda_2 \in B$ such that $\|\Lambda_1\| > \|\Lambda_2\|$. The left side of the inequality in Eq. 33 can be written as

$$\|T(\Lambda_1) - T(\Lambda_2)\| = \|\Gamma\| \|e^{-\Lambda_1\tau} - e^{-\Lambda_2\tau}\|.$$

It is evident that $\|\Gamma\| \leq \|\Gamma\| + \|\Phi\|$ and the maximum norm can be used as

$$\|e^{-\Lambda_1\tau} - e^{-\Lambda_2\tau}\| \leq \tau \|\Lambda_1 - \Lambda_2\| e^{\tau \max\{\|\Lambda_1\|, \|\Lambda_2\|\}}.$$

It can be seen that

$$\|T(\Lambda_1) - T(\Lambda_2)\| \leq \tau \|\Lambda_1 - \Lambda_2\| (\|\Gamma\| + \|\Phi\|) e^{\tau(\|\Gamma\| + \|\Phi\|)}$$

and by applying the smallness condition $\|\Gamma\| + \|\Phi\| \leq 1/(\tau e)$ (Eq. 17),

$$\|T(\Lambda_1) - T(\Lambda_2)\| \leq \|\Lambda_1 - \Lambda_2\|$$

is obtained, which proves that $T : B \rightarrow B$ is a contraction. This shows that by solving

$$\Lambda = -\Phi + \Gamma e^{-\Lambda\tau} \quad (36)$$

for the Λ matrix, a system is created

$$\frac{d\tilde{x}}{dt} = \Lambda\tilde{x}, \quad (37)$$

which approximates the DDE system of Eq. 15. As a result of the proposed iterative method, the eigenvalues and eigenvectors of Eq. 37 approximate, with a given degree of precision, the dominant eigenvalues of Eq. 14.

5.1 Comparison with the existing methods

- Chain approximation:
 - + The result is an approximating system with known system matrices (both eigenvalues and eigenvectors are known).
 - The resulting system is of a higher degree than the delay system.
 - The convergence rate of the algorithm is linear.
- Lambert W function:
 - + The result is a trajectory approximation.
 - + The convergence rate of the algorithm is exponential.
 - The algorithm requires numerical solvers for an exponential matrix equation.
 - Multiple branches of the Lambert W function must be used to create an accurate approximation, and the number of eigenvalues in a branch cannot be predetermined generally.
- QPmR algorithm:
 - + The algorithm determines the exact number of eigenvalues in the given complex domain, with the given computational error.
 - + The convergence rate of the algorithm is exponential.
 - The algorithm uses the quasi-polynomial equation, thus, will not contain any information on the eigenvectors.
 - The algorithm uses numerical solvers to compute the zeros of the zero-level curves created from the quasi-polynomial equation.

- Approximation of an explicit matrix:
 - + The result is an approximating system.
 - + The resulting system is of the same degree as the approximating system.
 - The algorithm calculates a matrix exponential numerically, which is a compute-intensive task.

6. Simulations and results

Let us consider two cases: a MAS consisting of five and twenty-five agents, respectively, with first-order dynamics in a platoon configuration as shown in the relation of Eq. 10, with the constant initial condition $\underline{\theta}(h) = \underline{x}_0$.

For the comparisons, a 6th order chain approximation was used. In the case of the Lambert W function, the initial matrix $Q_0 = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$ and the branches $k = -2, -1, 0, 1$ were used. For the quasi-polynomial root-finder algorithm (QPmR), a symbolic calculation to find the characteristic quasi-polynomial equation of the system was used, and the plane of the search was set to $[-\tau, 0] \times [-\tau j, \tau j]$. The algorithm for explicit matrix approximation was used with the initial matrix $\Lambda_0 = 0_{n \times n}$. The error threshold $1e-7$ was used in every iterative algorithm.

The dominant eigenvalues of the system consisting of five agents, with minor differences, was identified by every approximation method. In the case of the larger system, the chain and explicit matrix approximations could generate a result, while the quasi-polynomial root-finder algorithm and the Lambert W function were determined by numerical calculations. As such, the smaller system was chosen as a point of comparison for the algorithms.

Tables 1 and 2 contain a comparative summary of the four aforementioned algorithms: the number of iterations, the overall computation time and the dominant eigenvalues identified for a platoon consisting of five agents as shown in Fig. 2.

Fig. 3 shows that the linear system is generated by the explicit matrix approximation in just twelve steps for the DMAS that consists of twenty-five agents.

Fig. 4 shows that the resultant approximating system exhibits the same steady-state and transient behavior as the original delayed system using the same initial conditions. Since the initial position falls within the range of

Table 1: The number of iterations and the overall computation time for the algorithms.

Name of algorithm	No. of cycles	Computation time
Chain approximation	1	0.02 s
Lambert W function	35	7.5 min
QPmR algorithm	12	17.18 s
Explicit algorithm	4	0.03 s

Table 2: The eigenvalues returned by the studied algorithms.

Name of algorithm	Eigenvalues
Chain approximation	-0.1080, -0.9471, -1, -2.4736, -4.2511
Lambert W function	-0.1081, -0.9482, -1, -2.4658, -4.1603
QPmR algorithm	-0.1081, -0.9481, -1, -2.4661, -4.1603
Explicit algorithm	-0.1080, -0.9481, -1, -2.4661, -4.1603

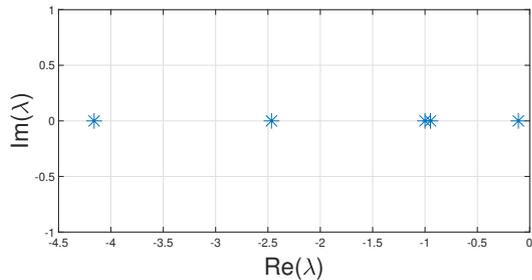


Figure 2: Dominant poles of the DMAS consisting of five agents.

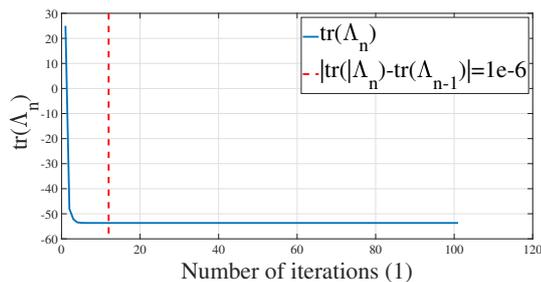


Figure 3: The explicit matrix approximation returns a valid system after 12 iterations for 25 agents.

0 to 100 m, the error of the approximating system is 8 cm in the transient domain and 5 mm under steady-state conditions, as shown in Fig. 5.

7. Conclusions

An iterative algorithm was proposed and tested based on which a degree-preserving approximation model can be created for a class of MAS in a platoon formation consisting of agents that exhibit first-order dynamics in the presence of a small communication delay. The algorithm uses methods of numerical computation. The obtained MAS is of the same degree and steady state as the delayed MAS, moreover, it correctly approximates the transient behavior. The algorithm was compared with existing methods for approximating eigenvalues and systems. Simulations show that the presented algorithm is suitable for the analysis of complex MAS.

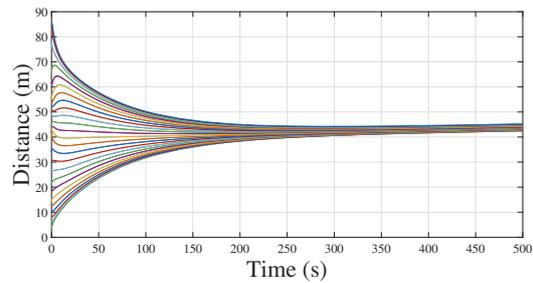


Figure 4: The trajectories of the platoon of the delayed MAS compared to the platoon of the approximated MAS created by the explicit matrix approximation.

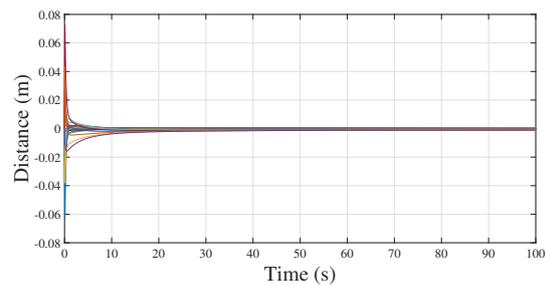


Figure 5: The trajectory of the approximation error.

Acknowledgement

The authors would like to express their gratitude to Dr. Mihály Pítuk (University of Pannonia, Hungary) for his useful comments.

REFERENCES

- [1] Kubera, Y.; Mathieu, P.; Picault, S.: Everything can be Agent!, in Proceedings of the ninth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS'2010) (W. van der Hoek, G.A. Kaminka, Y. Lespérance, M. Luck, S. Sen, eds.) (International Foundation for Autonomous Agents and Multiagent Systems, Toronto, Ontario, Canada), 1547–1548
- [2] Panait, L.; Luke, S.: Cooperative multi-agent learning: The state of the art, *Auton. Agents Multi-Agent Syst.*, 2005 **11**(3), 387–434, DOI: 10.1007/s10458-005-2631-2, ISBN: 978-981-10-2491-7
- [3] Zabat, M.; Stabile, N.; Farascarioli, S.; Browand, F.: The aerodynamic performance of platoons: A final report, Institute of Transportation Studies, Research Reports, Working Papers, Proceedings, Institute of Transportation Studies, UC Berkeley, 1995
- [4] Cheng-Lin, L.; Fei, L.: Consensus Problem of Delayed Linear Multi-agent Systems, Springer-Briefs in Electrical and Computer Engineering (Springer Singapore), 2017, DOI: 10.1007/978-981-10-2492-4, ISBN:978-981-10-2491-7

- [5] Wim, M.; Silviu-Iulian, N.: Stability and stabilization of time-delay systems: An eigenvalue-based approach (SIAM), 2007, DOI: [10.1137/1.9780898718645](https://doi.org/10.1137/1.9780898718645)
- [6] Lewis, F.L.; Zhang, H.; Hengster-Movric, K.; Das, A.: Cooperative control of multi-agent systems: Optimal and adaptive design approaches (Springer), 2014, DOI: [10.1007/978-1-4471-5574-4](https://doi.org/10.1007/978-1-4471-5574-4)
- [7] Trudeau, R.J.: Introduction to Graph Theory, Dover Books on Mathematics (Dover Publications), 2nd edn., 1994, ISBN: 978-0486678702
- [8] Chaiken, S.; Kleitman, D.J.: Matrix tree theorems, *J. Comb. Theory A*, 1978 **24**(3), 377 – 381, DOI: [10.1016/0097-3165\(78\)90067-5](https://doi.org/10.1016/0097-3165(78)90067-5)
- [9] Mesbahi, M.; Egerstedt, M.: Graph theoretic methods in multiagent networks, Princeton Series in Applied Mathematics (Princeton University Press), 2010, DOI: [10.1515/9781400835355](https://doi.org/10.1515/9781400835355)
- [10] Cepeda-Gomez, R.; Olgac, N.: An exact method for the stability analysis of linear consensus protocols with time delay, *IEEE T. Autom. Control*, 2011 **56**(7), 1734–1740, DOI: [10.1109/TAC.2011.2152510](https://doi.org/10.1109/TAC.2011.2152510)
- [11] Arino, O.; Pituk, M.: More on linear differential systems with small delays, *J. Differential Equations*, 2001 **170**(2), 381 – 407, DOI: [10.1006/jdeq.2000.3824](https://doi.org/10.1006/jdeq.2000.3824)
- [12] Györi, I.; Pituk, M.: Asymptotically ordinary delay differential equations, *Functional Differential Equations*, 2005 **12**, 187–208
- [13] Györi, I.; Pituk, M.: Asymptotic formulas for a scalar linear delay differential equation, *Electron. J. Qual. Theory Differ. Equ.*, 2016 **1**(72), 1–14, DOI: [10.14232/ejqtde.2016.1.72](https://doi.org/10.14232/ejqtde.2016.1.72)
- [14] Krasznai, B.; Györi, I.; Pituk, M.: The modified chain method for a class of delay differential equations arising in neural networks, *Math. Comput. Model.*, 2010 **51**(5), 452 – 460, DOI: [10.1016/j.mcm.2009.12.001](https://doi.org/10.1016/j.mcm.2009.12.001)
- [15] Corless, R.M.; Gonnet, G.H.; Hare, D.E.G.; Jeffrey, D.J.; Knuth, D.E.: On the Lambert W function, *Adv. Comput. Math.*, 1996 **5**(1), 329–359, DOI: [10.1007/BF02124750](https://doi.org/10.1007/BF02124750)
- [16] Vyhlídal, T.; Lafay, J.F.; Sipahi, R.: Delay systems: From theory to numerics and applications, *Advances in Delays and Dynamics* (Springer International Publishing), 2013, DOI: [10.1007/978-3-319-01695-5](https://doi.org/10.1007/978-3-319-01695-5)
- [17] Yi, S.; Duan, S.; Nelson, P.W.; Ulsoy, A.G.: The Lambert W Function Approach to Time Delay Systems and the LambertW_DDE Toolbox, *IFAC Proceedings Volumes*, 2012 **45**(14), 114–119, DOI: [10.3182/20120622-3-US-4021.00008](https://doi.org/10.3182/20120622-3-US-4021.00008)
- [18] Vyhlídal, T.; Zítek, P.: Quasipolynomial mapping based rootfinder for analysis of Time delay systems, in Proc. of IFAC Workshop on Time-Delay Systems, TDS 2003, DOI: [10.1016/S1474-6670\(17\)33330-X](https://doi.org/10.1016/S1474-6670(17)33330-X)
- [19] Vyhlídal, T.; Zítek, P.: QPmR – Quasipolynomial rootfinder: Algorithm update and examples, *Advances in Delays and Dynamics*, vol. 1 (Springer International Publishing, Cham), 2013 299–312, DOI: [10.1007/978-3-319-01695-5_22](https://doi.org/10.1007/978-3-319-01695-5_22), <http://www.cak.fs.cvut.cz/algorithms/qpmr>
- [20] Latif, A.: Banach contraction principle and its generalizations, *Topics in Fixed Point Theory* (Springer International Publishing, Cham), 2014 33–64, DOI: [10.1007/978-3-319-01586-6_2](https://doi.org/10.1007/978-3-319-01586-6_2)