mathematics

MDPI

Communication

Efficient Algorithm for the Computation of the Solution to a Sparse Matrix Equation in Distributed Control Theory

Leonardo Pedroso * and Pedro Batista

Institute for Systems and Robotics, Instituto Superior Técnico, Universidade de Lisboa, 1049-001 Lisboa, Portugal; pbatista@isr.tecnico.ulisboa.pt

* Correspondence: leonardo.pedroso@tecnico.ulisboa.pt

Abstract: In this short communication, an algorithm for efficiently solving a sparse matrix equation, which arises frequently in the field of distributed control and estimation theory, is proposed. The efficient algorithm stems from the fact that the sparse equation at hand can be reduced to a system of linear equations. The proposed algorithm is shown to require significantly fewer floating point operations than the state-of-the-art solution. The proposed solution is applied to a real-life example, which models a wide range of industrial processes. The experimental results show that the solution put forward allows for a significant increase in efficiency in relation to the state-of-the-art solution. The significant increase in efficiency of the presented algorithm allows for a valuable widening of the applications of distributed estimation and control.

Keywords: sparsity constraint; sparse matrix; sparse matrix equation; distributed control; distributed estimation

Citation: Pedroso, L.; Batista, P. Efficient Algorithm for the Computation of the Solution to a Sparse Matrix Equation in Distributed Control Theory. *Mathematics* **2021**, *9*, 1497. https://doi.org/10.3390/math9131497

Academic Editors: Theodore E. Simos and Charampos Tsitouras

Received: 29 May 2021
Accepted: 23 June 2021
Published: 25 June 2021

1. Notation

The $n \times o$ null matrix and the null vector of dimension n are denoted by $\mathbf{0}_{n\times o}$ and $\mathbf{0}_n$, respectively. The i-th component of a vector $\mathbf{v}$ is denoted by $[\mathbf{v}]_i$, and the entry (i,j) of a matrix $\mathbf{A}$ is denoted by $[\mathbf{A}]_{i,j}$. The vectorization of a matrix $\mathbf{A}$, denoted herein by $\mathrm{vec}(\mathbf{A})$, returns a vector composed of the concatenated columns of $\mathbf{A}$. The Kronecker product of two matrices $\mathbf{A}$ and $\mathbf{B}$ is denoted by $\mathbf{A} \otimes \mathbf{B}$. The cardinality of a set χ is denoted by $|\chi|$.

2. Introduction

Sparse matrix equations arise in various real-life problems in a broad range of engineering fields. Thus, given their impact, they have been extensively studied for a long time [1]. In recent developments in the field of distributed estimation and control theory, a sparse matrix equation has arisen, for which, to the best of the knowledge of the authors, an efficient algorithm for its solution has not yet been proposed.

Consider the matrix equation

$$\begin{cases} [\mathbf{AXB} - \mathbf{C}]_{i,j} = 0 & , (i,j) \in \chi \\ [\mathbf{X}]_{i,j} = 0 & , (i,j) \notin \chi \end{cases}, \tag{1}$$

where $\mathbf{A} \in \mathbb{R}^{n\times n}$, $\mathbf{B} \in \mathbb{R}^{o\times o}$, and $\mathbf{C} \in \mathbb{R}^{n\times o}$ are known and $\mathbf{X} \in \mathbb{R}^{n\times o}$ is the unknown. Let $\mathbf{E} \in \mathbb{R}^{n\times o}$ represent a known sparsity pattern, and define the set χ of integer pairs of the form (i,j) to index the nonzero entries of $\mathbf{E}$ as

$$\begin{cases} (i,j) \in \chi & , [\mathbf{E}]_{i,j} \neq 0 \\ (i,j) \notin \chi & , \text{otherwise} \end{cases}, i = 1,\ldots,n,\ j = 1,\ldots,o\,. \tag{2}$$

Equation (1) arises frequently in the field of distributed control and estimation theory. In fact, it is the backbone of the state-of-the-art methods for distributed filter design [2] and

controller synthesis [3] for large-scale dynamical systems. However, the solution to (1), presented in these papers, relies on an inefficient closed-form matrix computation. Define a matrix $\mathbf{Z}$, such that the vector $\mathbf{Z}\text{vec}(\mathbf{E})$ contains all the nonzero elements of $\mathbf{E}$. The solution to (1) presented in the literature is given by

$$\text{vec}(\mathbf{X}) = \mathbf{Z}^T \left(\mathbf{Z}(\mathbf{B} \otimes \mathbf{A})\mathbf{Z}^\mathbf{T} \right)^{-1} \mathbf{Z}\text{vec}(\mathbf{C}) \,, \tag{3}$$

as proven in ([2], Appendix B). Note that (3) requires the multiplication of a matrix of dimension $|\chi| \times no$ by other dimension $no \times no$, which requires $\mathcal{O}(|\chi|(no)^2)$ floating point operations. The number of practical applications, where it is convenient to model the interconnection of complex systems as a whole network, has been steadily increasing [4]. They have emerged in a broad range of engineering fields such as irrigation networks [5,6], power systems networks [7,8], traffic networks [9–12], and industrial processes [13]. Nevertheless, given the substantial increase in memory and processing power needed to control such systems, the classical control solutions cannot be implemented in practice. That is why, over the past decades, an effort has been made towards the development of distributed and computationally efficient algorithms to handle such systems. In this short communication, an algorithm to obtain an efficient and exact solution to (1) is proposed, which significantly increases the applicability of distributed methods that address the control and estimation problems for large-scale dynamical systems.

3. Materials and Methods

The following result is the basis for the algorithm proposed in this short communication.

Theorem 1. *The system of Equation* (1) *can be reduced to a system of linear equations of dimension* $|\chi|$*,*

$$\mathbf{S}\mathbf{x} = \mathbf{r} \,, \tag{4}$$

where $\mathbf{S} \in \mathbb{R}^{|\chi| \times |\chi|}$, $\mathbf{r} \in \mathbb{R}^{|\chi|}$, *and* $\mathbf{x} \in \mathbb{R}^{|\chi|}$ *is the vector of the ordered nonzero entries of* $\text{vec}(\mathbf{X})$.

Proof. For every integer pair $(i, j) \in \chi$, the first equation of (1) is given by

$$\sum_{k=1}^{n} \sum_{l=1}^{o} [\mathbf{A}]_{i,k} [\mathbf{X}]_{k,l} [\mathbf{B}]_{l,j} - [\mathbf{C}]_{i,j} = 0 \,. \tag{5}$$

Given the second equation of (1), one can rewrite (5) as

$$\sum_{(k,l) \in \chi} [\mathbf{A}]_{i,k} [\mathbf{X}]_{k,l} [\mathbf{B}]_{l,j} - [\mathbf{C}]_{i,j} = 0 \,. \tag{6}$$

Denote by $\mathbf{x} \in \mathbb{R}^{|\chi|}$ the vector of the ordered nonzero entries of $\text{vec}(\mathbf{X})$, and let $\boldsymbol{\phi} = [\phi_1 \; \phi_2]^T : \mathbb{N}^1 \to \mathbb{N}^2$ denote the resulting one-to-one map between the entries of $\mathbf{x}$ and the nonzero entries of $\mathbf{X}$. Defining $p := \boldsymbol{\phi}^{-1}(i, j)$ and $q := \boldsymbol{\phi}^{-1}(k, l)$, it follows that satisfying (6) for every integer pair $(i, j) \in \chi$ is equivalent to satisfying $\mathbf{S}\mathbf{x} = \mathbf{r}$, where $\mathbf{S} \in \mathbb{R}^{|\chi| \times |\chi|}$ and $\mathbf{r} \in \mathbb{R}^{|\chi|}$ are defined such that

$$[\mathbf{S}]_{p,q} = [\mathbf{A}]_{\phi_1(p), \phi_1(q)} [\mathbf{B}]_{\phi_2(q), \phi_2(p)} \,, \tag{7}$$

$p = 1, \ldots, |\chi| \,, q = 1, \ldots, |\chi|$; and

$$[\mathbf{r}]_p = [\mathbf{C}]_{\boldsymbol{\phi}(p)} \,, \; p = 1, \ldots, |\chi| \,; \tag{8}$$

respectively. It then follows that the solution to (1) is

$$[\mathbf{X}]_{i,j} = \begin{cases} [\mathbf{x}]_{\boldsymbol{\phi}^{-1}(i,j)} & , (i,j) \in \chi \\ 0 & , (i,j) \notin \chi \end{cases}, \tag{9}$$

where $\mathbf{x}$ is the solution of (4). □

Corollary 1. *It follows immediately from Theorem 1 that* (1) *admits: (i) one and only one solution if* $\mathbf{S}$ *is full rank; (ii) infinitely many solutions if* (4) *is consistent and* $\mathbf{S}$ *is not full rank; and (iii) no solution if* (4) *is inconsistent.*

Proposition 1. *Algorithm 1 can be used to obtain the solution to* (1)*, with* $\mathcal{O}(|\chi|^3)$ *floating point operations.*

Proof. Algorithm 1 computes $\mathbf{S} \in \mathbb{R}^{|\chi| \times |\chi|}$ and $\mathbf{r} \in \mathbb{R}^{|\chi|}$ according to (7) and (8), respectively. It, then, follows from Theorem 1 that (1) can be reduced to the solution to $\mathbf{S}\mathbf{x} = \mathbf{r}$. The LU decomposition of $\mathbf{S}$ can be carried out with $\mathcal{O}(|\chi|^3)$ floating point operations, using Gaussian elimination. The type of solution is then assessed according to Corollary 1. If the solution is unique, it follows from Theorem 1 that it is given by (9), which is implemented in Algorithm 1. □

Algorithm 1 Algorithm to efficiently compute the exact solution to (1)

Input: $\mathbf{A} \in \mathbb{R}^{n \times n}$, $\mathbf{B} \in \mathbb{R}^{o \times o}$, $\mathbf{C} \in \mathbb{R}^{n \times o}$, and $\mathbf{E} \in \mathbb{R}^{n \times o}$
Output: Solution to (1), $\mathbf{X} \in \mathbb{R}^{n \times o}$, if one unique solutions exists.
Compute set χ according to (2);
$p \leftarrow 0$; $\mathbf{S} \leftarrow \mathbf{0}_{|\chi| \times |\chi|}$; $\mathbf{r} \leftarrow \mathbf{0}_{|\chi|}$; $\mathbf{X} \leftarrow \mathbf{0}_{n \times o}$;
for $(i,j) \in \chi$ **do** `// Possibly parallelly`
  $p \leftarrow p + 1$; $[\mathbf{r}]_p \leftarrow [\mathbf{C}]_{i,j}$; $q \leftarrow 0$;
  for $(k,l) \in \chi$ **do**
    $q \leftarrow q + 1$; $[\mathbf{S}]_{p,q} \leftarrow [\mathbf{A}]_{i,k}[\mathbf{B}]_{l,j}$;
  end
end
Solve $\mathbf{S}\mathbf{x} = \mathbf{r}$ for $\mathbf{x}$; `// LU decomposition`
Assess type of solution according to Corollary 1;
if $\mathbf{S}\mathbf{x} = \mathbf{r}$ *has a unique solution* **then**
  $p \leftarrow 0$;
  for $(i,j) \in \chi$ **do** `// Possibly parallelly`
    $p \leftarrow p + 1$; $[\mathbf{X}]_{i,j} \leftarrow [\mathbf{x}]_p$;
  end
  return $\mathbf{X}$;
else
  Flag type of solution;
end

Remark 1. *It is important to point out that each of the three main steps of Algorithm 1 can be carried out in parallel. The first and last steps for loops of Algorithm 1 (which are commented on) can be carried out in parallel, for each* j*. In this case, the variable* p *has to be initialized, for each* j*, with the number of nonzero entries of* $\mathbf{E}$ *in the first* $j-1$ *columns. Moreover, the LU decomposition can also be carried out in parallel [14,15].*

Remark 2. *There are asymptotically faster algorithms for computing the LU decomposition of a square matrix. For instance, the Strassen algorithm [16] can perform the LU decomposition of* $\mathbf{S}$ *in* $\mathcal{O}(|\chi|^{\log_2 7})$ *floating point operations. Moreover, the Coppersmith–Winograd algorithm [17] requires only* $\mathcal{O}(|\chi|^{2.38})$ *floating point operations. However, an improvement in relation to*

the Strassen algorithm can only be noticed for large values of $|\chi|$. For more details, see ([18], Section 1.4).

Remark 3. *In the field of distributed estimation and control theory, $|\chi|$ is the sum of the number of sensor output signals available to each node of the network, and n is the order of the state–space dynamics of the whole system network. Given that, for most applications, each node has a similar order and a similar number of sensor output signals available, then $|\chi| \approx cn$, where $c \in \mathbb{N}$ is a constant. The system detailed in Section 4 illustrates this claim. It, thus, follows that the ratio of the number of floating points operations required by the proposed algorithm and the number required by the state-of-the-art methods is $\mathcal{O}(o^{-2})$.*

4. Results

In this section, the solution to the sparse matrix equation (1) is used to design a distributed filter for a large-scale network of N tanks. The computational efficiency of the synthesis with the proposed solution is compared to the state-of-the-art synthesis procedure. This network is widely studied [3,19,20] because it is analogous to a broad range of industrial processes. Consider N interconnected tanks, as shown in Figure 1, where N is an even integer. The water level of tank i is denoted by h_i. The network is actuated by $N/2$ pumps, which are controlled by the lower tanks, whose inputs are denoted by u_i for $i = 1, \ldots, N/2$, in accordance with the schematic. Each pump is connected to a three-way valve that regulates the fraction of the flow, held constant, that goes to each of the tanks supplied by the pump. Each tank has a sensor, which measures its water level. Making use of mass balances and Bernoulli's law, the system dynamics, in the absence of noise, are given by

$$\begin{cases} A_i \dot{h}_i(t) = -a_i\sqrt{2gh_i(t)} + a_{\frac{N}{2}+i}\sqrt{2gh_{\frac{N}{2}+i}(t)} + \gamma_i k_i u_i(t), \ i = 1, \ldots, N/2 \\ A_i \dot{h}_i(t) = -a_i\sqrt{2gh_i(t)} + (1-\gamma_{i-\frac{N}{2}-1})k_{i-\frac{N}{2}-1}u_{i-\frac{N}{2}-1}(t), \ i = \frac{N}{2}+2, \ldots, N \\ A_{\frac{N}{2}+1}\dot{h}_{\frac{N}{2}+1}(t) = -a_{\frac{N}{2}+1}\sqrt{2gh_{\frac{N}{2}+1}(t)} + (1-\gamma_{\frac{N}{2}})k_{\frac{N}{2}}u_{\frac{N}{2}}(t), \end{cases} \quad , \quad (10)$$

where A_i and a_i are the cross sections of tank i and of its outlet hole, respectively; the constant γ_i represents the fraction of the flow that passes through the valve i to the lower tanks; k_i is the constant of proportionality between the mass flow and the input of pump i; and g denotes the acceleration of gravity.

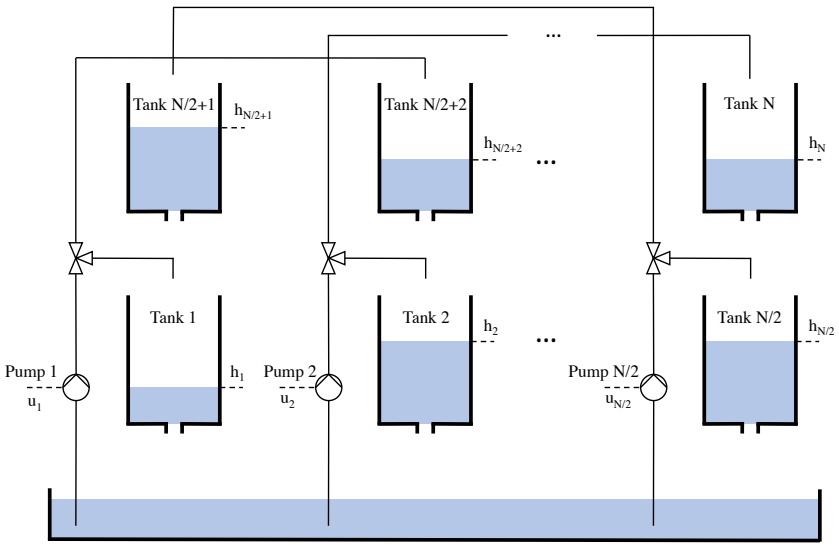

Figure 1. Schematic of the N tanks network.

The values of the constant parameters of the network are presented in Table 1. The dynamics of the network are nonlinear, thus the system is linearized about its operational point, which is selected to be $h(i) = 15\,\text{cm}, i = 1, \ldots, N$. The distributed filter is synthesized with the finite-horizon method ([2], Section 5), which is shown to have better state estimation performance in relation to other distributed filter synthesis methods. Each iteration k of this method computes a new gain, $\mathbf{K}(k) \in \mathbb{R}^{n \times o}$, solving

$$\begin{cases} \left[\mathbf{\Lambda}(k+1)\mathbf{K}(k)\mathbf{S}(k) - \mathbf{\Lambda}(k+1)\mathbf{P}(k|k-1)\mathbf{C}^T\right]_{i,j} = 0 & , (i,j) \in \chi \\ [\mathbf{K}(k)]_{i,j} = 0 & , (i,j) \notin \chi \end{cases}, \tag{11}$$

where $\mathbf{\Lambda}(k+1) \in \mathbb{R}^{n \times n}$, $\mathbf{S}(k) \in \mathbb{R}^{o \times o}$, and $\mathbf{P}(k|k-1) \in \mathbb{R}^{n \times n}$ are defined in [2], with $n = o = N$. Moreover, $\mathbf{C} = \mathbf{I}_N$ is the output matrix of the network and, given the configuration of the network, the sparsity pattern that defines the set χ is given by $\mathbf{E} = \mathbf{I}_N$. Note that (11) has the same form as (1), for which a novel efficient solution is proposed in this short communication.

Table 1. Values of the physical constants of the N tanks network.

Constant	Value
A_i, i odd	28 cm^2
A_i, i even	32 cm^2
a_i, i odd $\leq N/2$	0.071 cm^2
a_i, i even $\leq N/2$	0.057 cm^2
a_i, $i > N/2$	0.040 cm^2
g	981 $\text{cm}\,\text{s}^{-2}$
k_i	3.33 $\text{cm}^3\,\text{s}^{-1}\,\text{V}^{-1}$
γ_i, i odd	0.7
γ_i, i even	0.6

The goal is to compare the computational efficiency of the gain synthesis using the solution to (11) proposed in this short communication and compare it to the state-of-the-art solution (3). In that sense, the gain synthesis is performed for both alternatives, for a range of N. Figure 2 depicts the elapsed wall-clock time for the distributed filter synthesis, resulting from the average of three simulations of a MATLAB implementation on a single thread of a server with 24GB RAM and 24 Intel Xeon hexa-core CPUs at 2.40GHz. A MATLAB implementation of the distributed filter synthesis, of the application to the N tanks network, and of Algorithm 1, can be found in the *DECENTER* toolbox available at https://decenter2021.github.io (accessed on 23 June 2021). Figure 2 also depicts the linear regression in the logarithmic scale of the asymptotic evolution, which is considered to be achieved for $N > 40$. First, if the solution of (11) is the bottleneck of the gain synthesis procedure, one would expect that the synthesis wall-clock time grows asymptotically with $\mathcal{O}(N^5)$ for the state-of-the-art solution, and $\mathcal{O}(N^3)$ for the novel solution proposed in this short communication. It is possible to conclude from Figure 2 that this is, in fact, the case. The experimental complexity is slightly lower than what was projected, since MATLAB makes use of efficient LU decomposition algorithms like the ones pointed out in Remark 2. Second, it is also possible to confirm that the use of the solution proposed in this short communication leads to a ratio of wall-clock synthesis time between the proposed solution and the state-of-the-art solution that evolves with $\mathcal{O}(N^{-2.035})$. This improvement significantly widens the applicability of various distributed methods to large-scale systems.

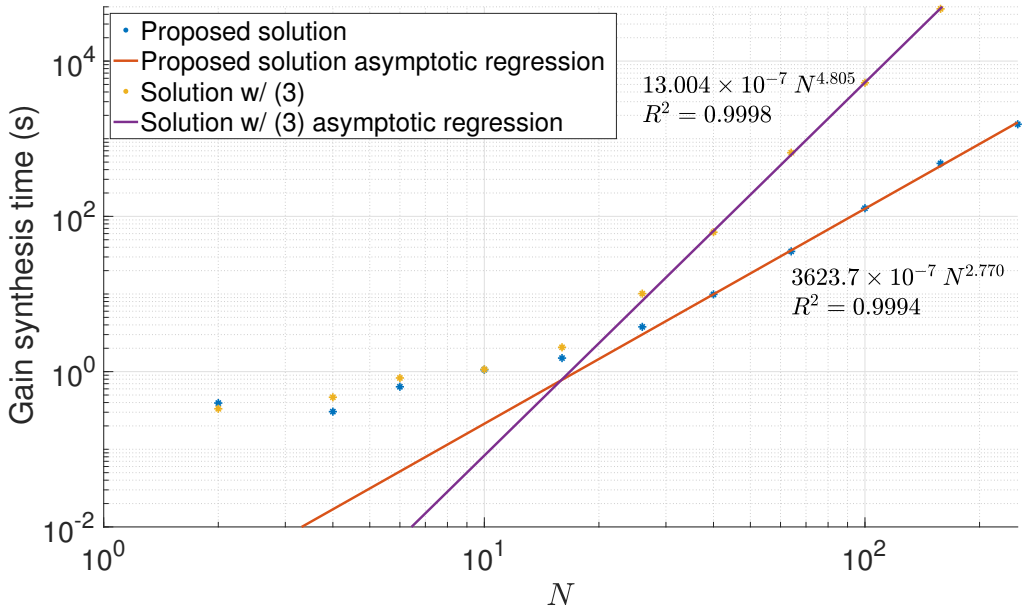

Figure 2. Synthesis wall-clock time of a distributed filter for the N tanks network, as a function of the dimension of the network.

5. Conclusions

In this short communication, an algorithm for efficiently solving the sparse matrix equation (1), which arises frequently in the field of distributed control and estimation theory, is proposed. The state-of-the-art methods for distributed filter design and controller synthesis rely on a very inefficient computation of the solution to (1), requiring $\mathcal{O}(|\chi|(no)^2)$ floating point operations. In this letter, an algorithm is proposed to compute the solution to this equation with $\mathcal{O}(|\chi|^3)$ floating point operations. For distributed estimation and control theory applications, the ratio of the number of floating points operations required by the proposed algorithm and the number required by the state-of-the-art methods is $\mathcal{O}(o^{-2})$. Furthermore, the algorithm proposed in this short communication was applied to an example representative of real-life large-scale industrial processes, and was compared to the state-of-the-art solution. It is shown that the solution put forward allows for a decrease of the computational complexity of the gain synthesis from $\mathcal{O}(N^{4.805})$, which is obtained for the state-of-the-art solution, to $\mathcal{O}(N^{2.770})$, where N is the dimension of the network. The significantly greater efficiency achieved with the proposed algorithm allows for the state-of-the-art distributed control and estimation algorithms to be implemented in an online framework and for large-scale dynamical systems, which would otherwise be unfeasible.

Author Contributions: Conceptualization, L.P.; methodology, L.P. and P.B.; software, L.P.; validation, L.P. and P.B.; formal analysis, L.P.; writing—original draft preparation, L.P.; writing—review and editing, L.P. and P.B; visualization, L.P.; supervision, P.B.; project administration, P.B.; funding acquisition, P.B. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Fundação para a Ciência e a Tecnologia (FCT) through LARSyS-FCT Project UIDB/50009/2020 and through the FCT project DECENTER [LISBOA-01-0145-FEDER-029605], funded by the Programa Operacional Regional de Lisboa 2020 and PIDDAC programs.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Duff, I.S. A survey of sparse matrix research. *Proc. IEEE* **1977**, *65*, 500–535. [CrossRef]
2. Viegas, D.; Batista, P.; Oliveira, P.; Silvestre, C. Discrete-time distributed Kalman filter design for formations of autonomous vehicles. *Control Eng. Pract.* **2018**, *75*, 55–68. [CrossRef]
3. Viegas, D.; Batista, P.; Oliveira, P.; Silvestre, C. Distributed controller design and performance optimization for discrete-time linear systems. *Optim. Control. Appl. Methods* **2020**, 1–18. [CrossRef]
4. Šiljak, D.D.; Zečević, A. Control of large-scale systems: Beyond decentralized feedback. *Annu. Rev. Control* **2005**, *29*, 169–179. [CrossRef]
5. Conde, G.; Quijano, N.; Ocampo-Martinez, C. Modeling and control in open-channel irrigation systems: A review. *Annu. Rev. Control* **2021**, *51*, 153–171. [CrossRef]
6. Cantoni, M.; Weyer, E.; Li, Y.; Ooi, S.K.; Mareels, I.; Ryan, M. Control of large-scale irrigation networks. *Proc. IEEE* **2007**, *95*, 75–91. [CrossRef]
7. Chen, C.; Wang, J.; Kishore, S. A distributed direct load control approach for large-scale residential demand response. *IEEE Trans. Power Syst.* **2014**, *29*, 2219–2228. [CrossRef]
8. Bumiller, G.; Lampe, L.; Hrasnica, H. Power line communication networks for large-scale control and automation systems. *IEEE Commun. Mag.* **2010**, *48*, 106–113. [CrossRef]
9. Tan, T.; Bao, F.; Deng, Y.; Jin, A.; Dai, Q.; Wang, J. Cooperative deep reinforcement learning for large-scale traffic grid signal control. *IEEE Trans. Cybern.* **2019**, *50*, 2687–2700. [CrossRef]
10. Keyvan-Ekbatani, M.; Yildirimoglu, M.; Geroliminis, N.; Papageorgiou, M. Multiple concentric gating traffic control in large-scale urban networks. *IEEE Trans. Intell. Transp. Syst.* **2015**, *16*, 2141–2154. [CrossRef]
11. Carlson, R.C.; Papamichail, I.; Papageorgiou, M.; Messmer, A. Optimal mainstream traffic flow control of large-scale motorway networks. *Transp. Res. Part C Emerg. Technol.* **2010**, *18*, 193–212. [CrossRef]
12. Li, Y.; Yang, L.; Yang, G. Network-based coordinated motion control of large-scale transportation vehicles. *IEEE/ASME Trans. Mechatron.* **2007**, *12*, 208–215. [CrossRef]
13. Vadigepalli, R.; Doyle Iii, F.J. Structural analysis of large-scale systems for distributed state estimation and control applications. *Control Eng. Pract.* **2003**, *11*, 895–905. [CrossRef]
14. Buoni, J.J.; Farrell, P.A.; Ruttan, A. Algorithms for LU decomposition on a shared memory multiprocessor. *Parallel Comput.* **1993**, *19*, 925–937. [CrossRef]
15. Liu, Z.; Cheung, D. Efficient parallel algorithm for dense matrix LU decomposition with pivoting on hypercubes. *Comput. Math. Appl.* **1997**, *33*, 39–50. [CrossRef]
16. Strassen, V. Gaussian elimination is not optimal. *Numer. Math.* **1969**, *13*, 354–356. [CrossRef]
17. Coppersmith, D.; Winograd, S. Matrix multiplication via arithmetic progressions. In Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing, New York, NY, USA, 25–27 May 1987 ; pp. 1–6.
18. Pan, V. Complexity of algorithms for linear systems of equations. In *Computer Algorithms for Solving Linear Algebraic Equations*; Springer: Berlin/Heidelberg, Germany, 1991; pp. 27–56.
19. Johansson, K.H. The quadruple-tank process: A multivariable laboratory process with an adjustable zero. *IEEE Trans. Control Syst. Technol.* **2000**. [CrossRef]
20. Casavola, A.; Garone, E.; Tedesco, F. A distributed multi-agent command governor strategy for the coordination of networked interconnected systems. *IEEE Trans. Autom. Control* **2014**, *59*, 2099–2112. [CrossRef]