

Received November 4, 2021, accepted November 16, 2021, date of publication November 22, 2021, date of current version December 2, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3129788

Modified Newton Integration Algorithm With Noise Tolerance for Discrete Algebraic Matrix Riccati Equation

SIYUAN LIAO^{1,2}, DONGYANG FU^{1,2,3}, HAOEN HUANG^{1,2}, AND CHENGZE JIANG^{1,2}

¹School of Electronics and Information Engineering, Guangdong Ocean University, Zhanjiang 524088, China

²Shenzhen Institute of Guangdong Ocean University, Guangdong Ocean University, Shenzhen 518108, China

³Guangdong Provincial Engineering and Technology Research Center of Marine Remote Sensing and Information Technology, Guangdong Ocean University, Zhanjiang 524088, China

Corresponding authors: Dongyang Fu (fdy163@163.com) and Haoen Huang (hh1848088380@gmail.com)

This work was supported in part by the Key Projects of the Guangdong Education Department under Grant 2019KZDXM019; in part by the Southern Marine Science and Engineering Guangdong Laboratory (Zhanjiang) under Grant ZJW-2019-08; in part by the High-Level Marine Discipline Team Project of Guangdong Ocean University under Grant 00202600-2009; in part by the First Class Discipline Construction Platform Project of Guangdong Ocean University, in 2019, under Grant 231419026; in part by the Guangdong Graduate Academic Forum Project under Grant 230420003; in part by the Postgraduate Education Innovation Project of Guangdong Ocean University under Grant 202159; and in part by the Provincial-Level College Student Innovation and Entrepreneurship Training Project under Grant S202110566052.

ABSTRACT Nowadays, the discrete algebraic matrix Riccati equation (DAMRE) is widely used in control system theory, engineering application, etc. In order to solve the problem with the high accuracy of DAMRE, a large number of researchers have achieved great success in theoretical analysis and actively explored methods. In addition, they have achieved great success in both theoretical analysis and practical investigation, and some actively explored methods or practical investigations have been very effective. However, since previous research has not considered noise tolerance, this may cause unacceptable results and unsatisfactory effectiveness in practical utilization scenarios. To this end, inspired by the traditional Newton-Raphson iterative (NRI) algorithm, a modified Newton integration (MNI) algorithm is proposed with excellent noise tolerance ability. Through theoretical analyses, the proposed MNI algorithm is confirmed to retain the fast convergence property of the NRI algorithm and also possess strong noise tolerance. The numerical experiment results demonstrate that the proposed MNI algorithm has advantages in accuracy and noise tolerance compared with other algorithms.


INDEX TERMS Discrete algebraic matrix Riccati equation (DAMRE), modified newton integration (MNI) algorithm, noise tolerance ability.

I. INTRODUCTION

The discrete algebraic matrix Riccati equation (DAMRE) is widely encountered in a variety of mathematics, physics, control and system theory, engineering calculation [1]–[5]. In recent years, there have been two main types of methods for solving the DAMRE, such as numerical algorithms and their related modifications as well as Hilbert space methods.

In terms of the numerical algorithms, the most common method for solving the DAMRE is the Newton-Raphson iterative (NRI) algorithm [6]. For example, Guo *et al.* early

employed the NRI algorithm to research the DAMRE and obtain some valuable results [7]. In addition to Guo *et al.*'s research, Feitzinger *et al.* provide an inexact Kleinman-Newton method for solving the DAMRE [8], which possesses better global convergence performance and lower computational cost. Furthermore, Benner *et al.* improve the Kleinman-Newton method [9] by modifying the Lyapunov method [10]–[12] with a low-rank structure, which avoids some necessary assumptions via the Lyapunov method. Moreover, Valeria provides a novel method combined with the projection technique [13] for the DAMRE, which has the accurate line search capability, and its steady-state error is monotonically decreasing. However, for these above-mentioned methods, the traditional iterative algorithm does not consider

The associate editor coordinating the review of this manuscript and approving it for publication was Yeliz Karaca .

noise tolerance and its solution accuracy requires further improvement.

For reducing errors and increasing operation speed in solving the DAMRE, Isfahani *et al.* provide the implementation of optimal control theory problems in Hilbert space [14] in an iterative manner with less computational time. Besides, based on fuzzy differential equation on Hilbert space, Arqub *et al.* present a novel solving method to the DAMRE [15], which can attain more accurate results with more minor error and faster convergence. Next, in light of the above research [15], Arqub *et al.* further furnish another computational algorithm for the DAMRE with a reproducing kernel Hilbert space [16], which can improve the accuracy of the DAMRE solution, but the calculation process is complicated. Furthermore, inspired by Arqub *et al.*'s research, Sakar [17] also take into account the difference between constant and variable coefficients of the DAMRE and construct a more extensive and convenient solving method, whose steady-state error becomes significantly smaller, and its numerical solution is more precise. To sum up, the Hilbert space methods can be well employed in the DAMRE, while they have some drawbacks that the Hilbert space methods cannot ignore. Mainly, the Hilbert space methods are more complex and inconvenient for practical engineering calculation and application.

Nevertheless, the aforementioned studies do not consider the effect of noise on the solution procedure. But, in practical applications, noise from external sources or internal computing equipment can cause difficulty in solving the DAMRE [18], [19]. For this reason, it is quite critical to design a computational algorithm with noise tolerance and good convergence. Therefore, in this paper, a modified Newton integration (MNI) algorithm is proposed for solving the DAMRE, which is constructed in a discrete form and combines the advantages of the above methods. The proposed MNI algorithm not only possesses the fast convergence speed like the numerical algorithms, but it also has the same high precision solution as the Hilbert space method. Furthermore, by numerical experiments, the proposed MNI algorithm exhibits fast convergence performance and excellent noise tolerance compared to the zeroing neural network (ZNN) algorithm [20]–[22], the gradient neural network (GNN) algorithm [23], [24], and the NRI algorithm [25].

The rest of this article is composed of four parts. It is expressed as follows: In Section II, the derivation process of the proposed MNI algorithm is presented and compared to other algorithms. Subsequently, in order to verify that the proposed MNI algorithm is superior to different algorithms, Section III lists the corresponding theoretical. Moreover, in Section IV, through comparative numerical experiments, the proposed MNI algorithm is verified to possess the fast convergence and strong robustness performance with different sampling periods, whether in noisy or noise-free environments. Finally, Section V summarizes the full article. The main contributions of this article can be summarized as

- 1) The proposed MNI algorithm is an important improvement on the NRI algorithm [25] for solving the DAMRE with fast convergence and strong robustness.
- 2) The proposed MNI algorithm significantly improves the accuracy of the steady-state error from $O(\tau)$ to $O(\tau^2)$ without the use of any time difference information, where τ denotes the sampling period.
- 3) The proposed MNI algorithm handles the DAMRE in noisy environments with a discrete-time form, which facilitates practical implementation on digital devices.

II. PROGRAMMING AND FORMULA DERIVATION

Without loss of generality to solve the DAMRE [26], the problem can be stated as

$$A_k^T X_k + X_k A_k - X_k U_k X_k + C_k = \mathbf{0}, \quad (1)$$

where superscript T denotes the transpose of a matrix or a vector, factor matrices A_k , U_k , and $C_k \in \mathbb{R}^{n \times n}$. Giving the definition of the updating index $k = t/\tau$, where t denotes the instant time and τ denotes the sampling period.

Then, the error function of equation (1) can be written as

$$E_k = A_k^T X_k + X_k A_k - X_k U_k X_k + C_k, \quad (2)$$

with unknown matrix $X_k \in \mathbb{R}^{n \times n}$. In order to use the NRI algorithm [25] to find X_k , thus, the derivative of E_k with respect to X_k is as follows:

$$\frac{\partial E_k}{\partial X_k} = I \otimes A_k^T + A_k^T \otimes I - (U_k X_k)^T \otimes I - I \otimes (X_k U_k) \in \mathbb{R}^{n^2 \times n^2}, \quad (3)$$

where symbol \otimes denotes Kronecker product and matrix I denotes identity matrix. Let equation (3) equals W_k , that is

$$W_k = I \otimes A_k^T + A_k^T \otimes I - (U_k X_k)^T \otimes I - I \otimes (X_k U_k) \in \mathbb{R}^{n^2 \times n^2}.$$

Then, using the NRI algorithm [25] to solve the DAMRE (1), the iterative formula is

$$\mathbf{x}_{k+1} = \mathbf{x}_k - W_k^{-1} \mathbf{e}_k, \quad (4)$$

where superscript -1 represents the inverse operation of a square matrix. In addition, defined

$$\begin{cases} \mathbf{e}_k = \text{vec}(E_k) \in \mathbb{R}^{n^2}, \\ \mathbf{x}_k = \text{vec}(X_k) \in \mathbb{R}^{n^2}, \end{cases}$$

where $\text{vec}(\cdot)$ denotes the matrix vector operator. Thereafter, dividing both sides of the equation (4) by τ , it can be rewritten into

$$\frac{\mathbf{x}_{k+1} - \mathbf{x}_k}{\tau} = -\frac{W_k^{-1} \mathbf{e}_k}{\tau}.$$

According to the definition of derivative, the above equation can be rewritten as in the limit as $\tau \rightarrow 0$

$$\dot{\mathbf{x}}_k = -\frac{1}{\tau} W_k^{-1} \mathbf{e}_k. \quad (5)$$

In order to facilitate theoretical derivation of the following formula, the Euler forward difference formula [27], [28] can be written as

$$\dot{\mathbf{x}}_k = \frac{\mathbf{x}_{k+1} - \mathbf{x}_k}{\tau} + \mathcal{O}(\tau). \quad (6)$$

Adding an integration feedback term to improve the robustness [29] of formula (5) and combining the Euler forward difference formula (6), the proposed modified Newton integration (MNI) algorithm for solving DAMRE (1) is expressed as

$$\mathbf{x}_{k+1} = \mathbf{x}_k - W_k^{-1} \left(\mathbf{e}_k + \zeta \sum_{i=1}^k \mathbf{e}_i \right), \quad (7)$$

where $\zeta = \beta\tau^2 \in (0, 1)$ denotes the coefficient of integration term.

Remark 1: It is worth noting that from the automatic control theory point of view, the NRI algorithm (4) can be regarded as a proportional control algorithm. It is well known that the proportional control action is proportional to the deviation of the system [30], [31]. Once the system has deviated, the proportional regulation immediately generates a regulating action to reduce the deviation. However, this control action has no noise tolerance, which means that any slight disturbance in the system will have a considerable impact. Therefore, the MNI algorithm (7) proposed in this paper is designed to overcome the shortcomings of the NRI algorithm (4) and eliminate the steady state error by introducing an integral term in the controller. The integral term is integral to the error depending on the time, and the value of the integral term increases as the time increases. Thus, even if the error is small, the integral term increases with time, and it drives the output of the controller to increase so that the steady-state error is further reduced until it equals zero. This is equivalent to converting proportional control to proportional-integral control, making the NRI algorithm (4) noise tolerance.

In order to better compare the superiority of the proposed MNI algorithm (7), the gradient neural network algorithm (GNN) [23] is subsequently introduced as

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha \cdot \nabla \mathbf{e}_k, \quad (8)$$

where $\alpha \in \mathbb{R}^+$ denotes the step size; the gradient of error matrix $\nabla \mathbf{e}_k$ can be constructed as follows:

$$\begin{cases} \nabla E_k = 2A_k H_k + 2H_k A^T - (2H_k X_k^T U_k^T + 2F_k H_k), \\ F_k = X_k U_k, \\ H_k = C_k + A_k^T X_k + X_k A_k - F_k X_k. \end{cases}$$

Besides, the zeroing neural network algorithm (ZNN) [20] to solve the DAMRE (1) is provided as

$$\mathbf{x}_{k+1} = (L_k M_k + I) \mathbf{x}_k - L_k (\mathbf{c}_k - \lambda \mathbf{e}_k), \quad (9)$$

with

$$\begin{cases} L_k = \tau(B_k + S_k)^{-1} \in \mathbb{R}^{n^2 \times n^2}, \\ M_k = D_k - G_k \in \mathbb{R}^{n^2 \times n^2}, \end{cases}$$

where $\lambda \in \mathbb{R}^+$ is a coefficient.

Remark 2: To make the equation (2) more concise, the following matrices are constructed using Kronecker product as

$$\begin{cases} B_k = I \otimes (A_k^T - X_k U_k) \in \mathbb{R}^{n^2 \times n^2}, \\ S_k = (A_k - U_k X_k) \otimes I \in \mathbb{R}^{n^2 \times n^2}, \\ D_k = I \otimes (X_k \dot{U}_k - \dot{A}_k^T) \in \mathbb{R}^{n^2 \times n^2}, \\ G_k = \dot{A}_k^T \otimes I \in \mathbb{R}^{n^2 \times n^2}, \\ J_k = A_k^T \otimes I \in \mathbb{R}^{n^2 \times n^2}. \end{cases}$$

where the vector $\mathbf{c}_k = \text{vec}(C_k) \in \mathbb{R}^{n^2}$.

III. ANALYSES AND PROOFS

In this section, the corresponding theoretical analyses and proofs of the MNI algorithm (7) are provided, which illustrate its high accuracy and strong robustness under different noise environments.

Theorem 1: With the formula (6), the proposed MNI algorithm (7) can be converted into

$$\mathbf{e}_{k+1} + \zeta \sum_{i=1}^k \mathbf{e}_i + \mathcal{O}(\tau^2) = \mathbf{0}, \quad (10)$$

where $\mathcal{O}(\tau^2)$ represents the second-order error.

Proof : The proposed MNI algorithm (7) can be expressed as followed directly by (11).

$$W_k(\mathbf{x}_{k+1} - \mathbf{x}_k) = - \left(\mathbf{e}_k + \zeta \sum_{i=1}^k \mathbf{e}_i \right). \quad (11)$$

Concurrently, on the basis of the derivative definition of the discrete time function, it has

$$\begin{aligned} \dot{\mathbf{e}}_k &= \lim_{\tau \rightarrow 0} \frac{\mathbf{e}_{k+1} - \mathbf{e}_k}{\tau} \\ &= \lim_{\tau \rightarrow 0} \frac{(W_k \mathbf{x}_{k+1} + \mathbf{c}_k) - (W_k \mathbf{x}_k + \mathbf{c}_k)}{\tau} \\ &= W_k \dot{\mathbf{x}}_k, \end{aligned} \quad (12)$$

Thus, based on the above inference, dividing both sides of the equation (11) simultaneously by τ can obtain:

$$\dot{\mathbf{e}}_k = \frac{W_k(\mathbf{x}_{k+1} - \mathbf{x}_k)}{\tau} = -\frac{1}{\tau} \left(\mathbf{e}_k + \zeta \sum_{i=0}^k \mathbf{e}_i \right). \quad (13)$$

whereafter, by virtue of the Euler forward difference formula (6), the above equation (13) can be formulated as

$$\frac{1}{\tau}(\mathbf{e}_{k+1} - \mathbf{e}_k) + \mathcal{O}(\tau) = -\frac{1}{\tau} \left(\mathbf{e}_k + \zeta \sum_{i=0}^k \mathbf{e}_i \right).$$

Evidently, the above formula can be turned into

$$\mathbf{e}_{k+1} + \zeta \sum_{i=1}^k \mathbf{e}_i + \mathcal{O}(\tau^2) = \mathbf{0}. \quad (14)$$

The proof is thus complete. \square

To verify that the MNI algorithm (7) globally converges to $\mathbf{O}(\tau^2)$ under the zero-noise environment, the next theorem will be given as below.

Theorem 2: The steady-state error $\lim_{k \rightarrow \infty} \|e_k\|_2$ of proposed MNI algorithm (7) is equal to $\mathbf{O}(\tau^2)$, where $\|\cdot\|_2$ denotes the L_2 norm operation.

Proof: Based on the Theorem 1, taking any m^{th} element from e_k as a subsystem generates

$$e_{k+1}^m + \zeta \sum_{i=0}^k e_i^m + O(\tau^2) = 0. \quad (15)$$

Letting the subsystem of the $(m)^{\text{th}}$ element of matrix E_k at $(k-1)\tau$ time instant, it has

$$e_k^m + \zeta \sum_{i=0}^{k-1} e_i^m + O(\tau^2) = 0. \quad (16)$$

Next, subtracting formula (16) from formula (15) can generate

$$e_{k+1}^m = \rho e_k^m + O(\tau^2), \quad (17)$$

where $\rho = 1 - \zeta \in (0, 1)$. It is natural to develop the iterative procedure (17) as follows:

$$\begin{aligned} e_{k+1}^m &= \rho e_k^m + O(\tau^2), \\ e_k^m &= \rho e_{k-1}^m + O(\tau^2), \\ &\vdots \\ e_2^m &= \rho e_1^m + O(\tau^2). \end{aligned}$$

Afterward, both sides of the above equation multiplying $-\rho^{k-n}$, where n denotes the n^{th} equation of the iteration formula (17). Last, using the cumulative method to add up above equations can produce

$$e_{k+1}^m = \rho^k e_1^m + O(\tau^2). \quad (18)$$

When k tends to infinity, it has

$$\lim_{k \rightarrow \infty} e_{k+1}^m = \lim_{k \rightarrow \infty} (\rho^k e_1^m + O(\tau^2)) = O(\tau^2). \quad (19)$$

Due to e_k^m and e_k possess the same property, the steady-state error $\lim_{k \rightarrow \infty} \|e_k\|_2$ is equal to $\mathbf{O}(\tau^2)$ under the zero-noise environment. That is to say, the proposed MNI algorithm (7) globally converges to $\mathbf{O}(\tau^2)$ for solving the DAMRE (1) under the zero-noise environment.

The proof is thus complete. \square

To further make a thorough inquiry into the theoretical solution of the proposed MNI algorithm (7) in the case of constant and linear noisy environments, the next theorem is given.

Theorem 3: Assuming the linear noise is $\mathbf{q}_k = \boldsymbol{\varphi}k\tau + \mathbf{v}$, which $\boldsymbol{\varphi} \in \mathbb{R}^{n^2}$ is a coefficient vector, and $\mathbf{v} \in \mathbb{R}^{n^2}$ is a constant vector. The steady-state error $\lim_{k \rightarrow \infty} \|e_k\|_2$ of the proposed MNI algorithm (7) is $\|\boldsymbol{\varphi}/\beta\tau\|_2 + \mathbf{O}(\tau^2)$. When $\boldsymbol{\varphi} = 0$, \mathbf{q}_k is transformed into the constant noise. At this point,

the steady-state error $\lim_{k \rightarrow \infty} \|e_k\|_2$ of the MNI algorithm (7) is $\mathbf{O}(\tau^2)$, no matter $\mathbf{q}_k = \mathbf{v}$ is.

Proof: When the linear or constant noise components exist in the system, the noise can be divided into two parts. One is the main noise $\mathbf{q}_k = \boldsymbol{\varphi}k\tau + \mathbf{v}$ and the other residual noise is $\mathbf{O}(\tau^2)$. According to Theorem 1 and considering the main noise, it has

$$e_{k+1} + \zeta \sum_{i=1}^k e_i + \boldsymbol{\varphi}k\tau + \mathbf{v} = \mathbf{0}. \quad (20)$$

Taking any the m^{th} element of matrix e_k as a subsystem, the above equation (20) is presented as

$$e_{k+1}^m + \zeta \sum_{i=1}^k e_i^m + \varphi^m k\tau + v^m = 0. \quad (21)$$

Performing Z-transformation of formula (21) can generate

$$ze^m(z) + \zeta \frac{ze^m(z)}{z-1} + \frac{z\varphi^m\tau}{(z-1)^2} + \frac{zv^m}{z-1} = 0. \quad (22)$$

After formulating and simplifying, the above formula is re-expressed as

$$e^m(z) = -\frac{z\varphi^m\tau + z(z-1)v^m}{(z-1)(z+\zeta-1)z}. \quad (23)$$

Obviously, poles of $e^m(z)$ are $z_1 = 0$, $z_2 = 1$ and $z_3 = 1 - \zeta$. Utilizing the final value theorem for formula (23), the result is

$$\begin{aligned} \lim_{k \rightarrow \infty} e_k^m &= \lim_{z \rightarrow 1} (z-1)e^m(z) \\ &= \lim_{z \rightarrow 1} \frac{-z\varphi^m\tau - z(z-1)v^m}{z^2 + \zeta z - 1} \\ &= -\frac{\varphi^m}{\beta\tau}. \end{aligned} \quad (24)$$

Afterwards, to prove residual noise in the formula (7) that can be written as $e_{k+1} + \zeta \sum_{i=1}^k e_i + \mathbf{O}(\tau^2) = \mathbf{0}$. It is simple to find that the proof method is the same as Theorem 2. Therefore, the steady-state error of the residual noise is equal to $\mathbf{O}(\tau^2)$. In summary, the steady-state error $\lim_{k \rightarrow \infty} \|e_k\|_2$ of the proposed MNI algorithm (7) is $\|\boldsymbol{\varphi}/\beta\tau\|_2 + \mathbf{O}(\tau^2)$.

The proof is thus complete. \square

On top of that, the robustness with respect to random noise of the proposed MNI algorithm (7) is furnished in next theorem.

Theorem 4: Supposing the random noise $\mathbf{q}_k = \xi_k \in \mathbb{R}^{n^2}$, the steady-state error $\lim_{k \rightarrow \infty} \|e_k\|_2$ of the proposed MNI (7) algorithm is less than the upper bound $2n^2 \sup_{1 \leq i \leq k, 1 \leq m \leq n^2} |\xi_i^m|/\zeta + \mathbf{O}(\tau^2)$.

Proof: According to the Theorem 3 the steady-state error of MNI algorithm (7) has two parts, in this theorem, that is $\mathbf{q}_k = \xi_k$ and $\mathbf{O}(\tau^2)$. With the derivation process of the Theorem 2, using the formula of (17), similarly, the formula can be written as

$$e_{k+1}^m = \rho e_k^m + (\xi_k^m - \xi_{k-1}^m). \quad (25)$$

Let $\xi_k^m - \xi_{k-1}^m = \varepsilon_k^m$. The formula (25) is transformed into

$$e_{k+1}^m = \rho e_k^m + \varepsilon_k^m. \quad (26)$$

Then, using the same method, multiplying $-\rho^{k-n}$ on both sides of equation (26), the formula (26) can be produced

$$e_{k+1}^m = -\rho^k e_1^m - (\rho^{k-1} \varepsilon_1^m + \dots + \rho^1 \varepsilon_{k-1}^m + \rho^0 \varepsilon_k^m). \quad (27)$$

When k tends to infinity, ρ^k tends to zero. Consequently, using scaling method, the formula (27) can be transformed as

$$\begin{aligned} \lim_{k \rightarrow \infty} e_{k+1}^m &= \lim_{k \rightarrow \infty} (-\rho^k e_1^m - (\rho^{k-1} \varepsilon_1^m + \dots + \rho^0 \varepsilon_k^m)) \\ &= \max_{1 \leq i \leq k} \varepsilon_i^m \left(\frac{1 - \lim_{k \rightarrow \infty} \rho^{k-1}}{1 - \rho} \right) \\ &< \max_{1 \leq i \leq k} \frac{\varepsilon_i^m}{\zeta} \\ &< 2 \max_{1 \leq i \leq k} \frac{|\xi_i^m|}{\zeta}. \end{aligned} \quad (28)$$

Adding the remnant noise $O(\tau^2)$, the steady-state error of the MNI algorithm (7) under random noise environment is

$$\lim_{k \rightarrow \infty} \|e_k\|_2 < 2n^2 \sup_{\substack{1 \leq i \leq k \\ 1 \leq m \leq n^2}} \frac{|\xi_i^m|}{\zeta} + O(\tau^2). \quad (29)$$

The proof is thus complete. \square

IV. SIMULATION EXPERIMENTS

In this section, the corresponding numerical simulation experiments are conducted to show the superiority of the proposed MNI algorithm (7) in solving the DAMRE (1).

A. EXAMPLE OF MATRIX

Specifically, the DAMRE (1) is given as $A_k^T X_k + X_k A_k - X_k U_k X_k + C_k = 0$, where X_k is an unknown matrix and the other matrices are given below [32],

$$\left\{ \begin{aligned} A_k &= \begin{bmatrix} 5 + \sin(k\tau) & \cos(k\tau) \\ -\cos(k\tau) & 5 + \sin(k\tau) \end{bmatrix}, \\ U_k &= \begin{bmatrix} \psi_k & 0 \\ 0 & \phi_k \end{bmatrix}, C_k = \begin{bmatrix} \chi_k & 0 \\ 0 & \sigma_k \end{bmatrix}, \\ \psi_k &= (4 + e^{-k\tau} - \cos(k\tau))^2, \\ \phi_k &= \left(2 + \frac{1}{k\tau + 1} + \sin(k\tau)\right)^2, \\ \chi_k &= \left(\frac{1}{k\tau + 1}\right)^2 + \sin(k\tau) + 3, \\ \sigma_k &= (2 + e^{-k\tau})^2 - \cos(k\tau) + 6. \end{aligned} \right.$$

B. ANALYSIS THE EXPERIMENT

To better evaluate the proposed MNI algorithm (7), the maximum steady-state residual error (MSSRE) and the average computing time per iteration (ACTPI) are selected as performance indexes. In addition, the total time studied in this paper is $t = k\tau = 20$ s. These metrics are recorded in Table. 1 and the corresponding visual results on MSSRE are presented in Figs. 1–4.

- 1) *Zero Noise Environment*: The MSSRE of all four algorithms in zero noise environment that is the case free from external interference, can be clearly demonstrated in Fig 1, which is arranged in descending order as GNN algorithm (8), ZNN algorithm (9), NRI algorithm (4) and MNI algorithm (7). In fact, Table 1 illustrates the specific MSSRE of the four algorithms. It can be demonstrated here that when the sampling interval $\tau = 0.01$ s, the MSSRE of the MNI algorithm (7) proposed in this paper is 1.9211×10^{-2} , while the MSSRE of the NRI algorithm (4) is 4.3523×10^{-2} , which reflects that the MNI algorithm (7) has better noise tolerance than the NRI algorithm (4). Of course, since the difference in MSSRE between the two is not obvious at the sampling interval of $\tau = 0.01$ s, another experiment with sampling interval $\tau = 0.001$ s is done, and as shown in Fig. 1(b), the NRI algorithm (4) obviously has a higher MSSRE than the proposed MNI algorithm (7). It can be specifically seen in Table 1 that the NRI algorithm (4) is 10 times higher than the proposed MNI algorithm (7). Moreover, the orders of magnitudes of MSSRE of ZNN algorithm (9) as well as GNN algorithm (8) are 10^0 as well as 10^1 , which are different from the proposed MNI algorithm (7).
- 2) *Constant Noise Environment*: Due to having interference noise added, the MSSRE of each algorithm increases than that in the interference-free condition, except for the MNI algorithm (7) proposed in this paper. It still maintains good noise tolerance, and its MSSRE is 1.9211×10^{-2} at sampling interval $\tau = 0.01$ s, which is comparable to the zero noise environment. But the GNN algorithm (8) is another extreme, with the worst noise tolerance in the case of constant noise $q_k = 10$. The MSSRE grows from 1.3947×10^1 in the zero noise environment to 1.3413×10^6 in the constant noise environment. Further, the noise tolerance of each algorithm is more evident at the sampling interval $\tau = 0.001$ s. Overall, the MNI algorithm (7) proposed in this paper performs the best, and the GNN algorithm (8) performs the worst in the constant noise $q_k = 10$ environment.
- 3) *Linear Noise Environment*: Consider the linear noise $q_k = 10 \times (k\tau + 1)$, which is the noise whose noise interference is enhanced as the number of iterations increases. Therefore this has a more tremendous test for different algorithms. Figure 3 clearly shows that the MSSRE of the NRI (4) and GNN (8) algorithms tends to infinity as the noise intensifies, which is enough to reflect that these two algorithms cannot resist the interference of strong noise. Moreover, it is surprising that the ZNN algorithm (9) also can not resist the property of strong noise, and it exhibits strong oscillation. However, the MNI algorithm (7) proposed in this paper maintains excellent noise tolerance regardless of the sampling interval $\tau = 0.01$ s or $\tau = 0.001$ s, with MSSREs of 8.0229×10^0 and 8.0012×10^{-1} , respectively. From Theorem 3 proposed above, it is known that the

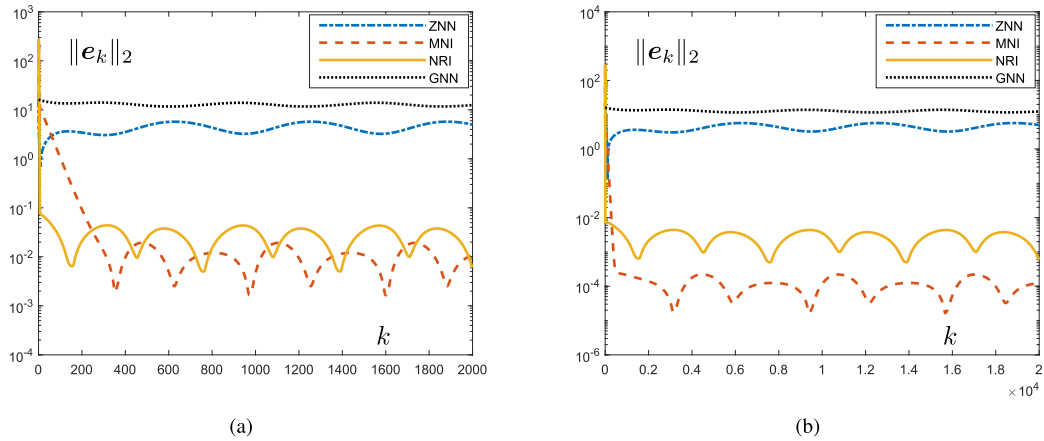


FIGURE 1. Zero noise $q_k = 0$ steady-state error convergence of four different algorithms, the proposed MNI algorithm (7), ZNN algorithm (9), NRI algorithm (4), and GNN algorithm (8). (a) $\tau = 0.01$ s. (b) $\tau = 0.001$ s.

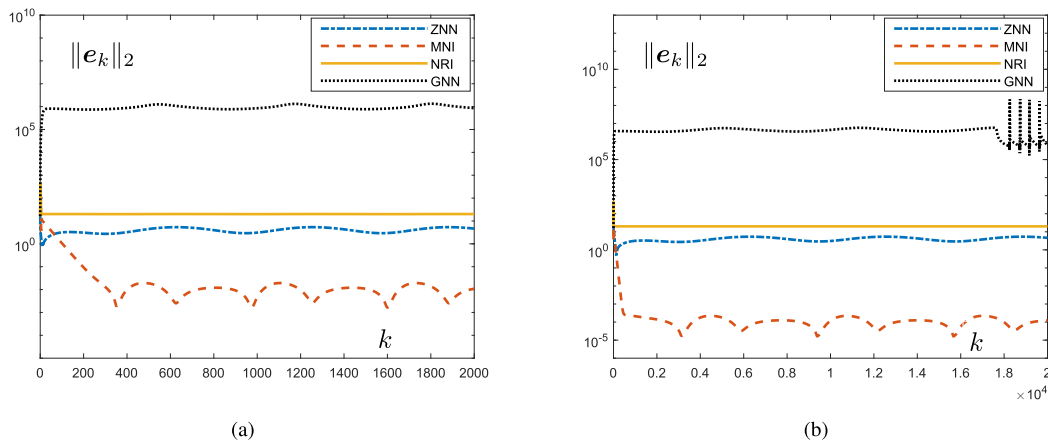


FIGURE 2. Constant noise $q_k = 10$ steady-state error convergence of four different algorithms, the proposed MNI algorithm (7), ZNN algorithm (9), NRI algorithm (4), and GNN algorithm (8). (a) $\tau = 0.01$ s. (b) $\tau = 0.001$ s.

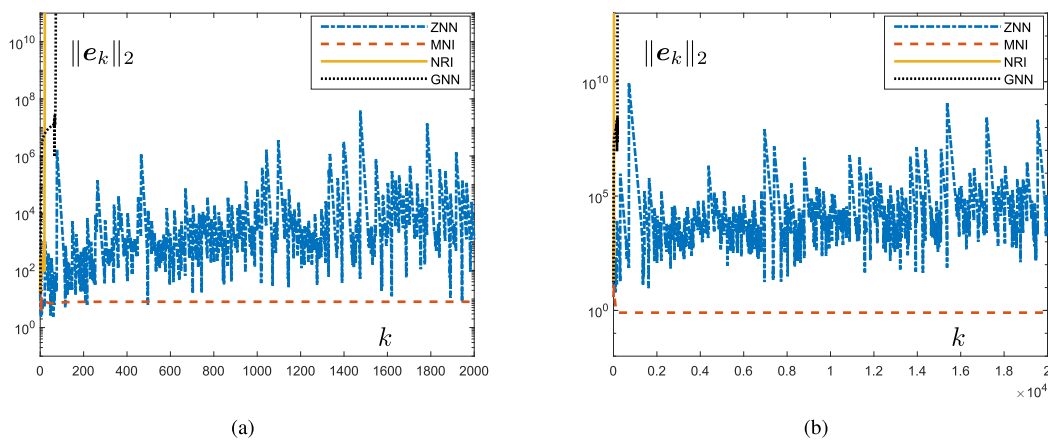


FIGURE 3. Linear noise $q_k = 10 \times (k\tau + 1)$ steady-state error convergence of four different algorithms, the proposed MNI algorithm (7), ZNN algorithm (9), NRI algorithm (4), and GNN algorithm (8). (a) $\tau = 0.01$ s. (b) $\tau = 0.001$ s.

steady-state error of the proposed MNI algorithm (7) is $\|\varphi/\beta\tau\|_2 + \mathcal{O}(\tau^2)$, which is almost a constant. Therefore, the result of the proposed MNI algorithm (7) in Fig. 3 is

a straight line, which also shows that the proposed MNI algorithm (7) is extremely noise tolerance to the linear noise compared to the other algorithms.

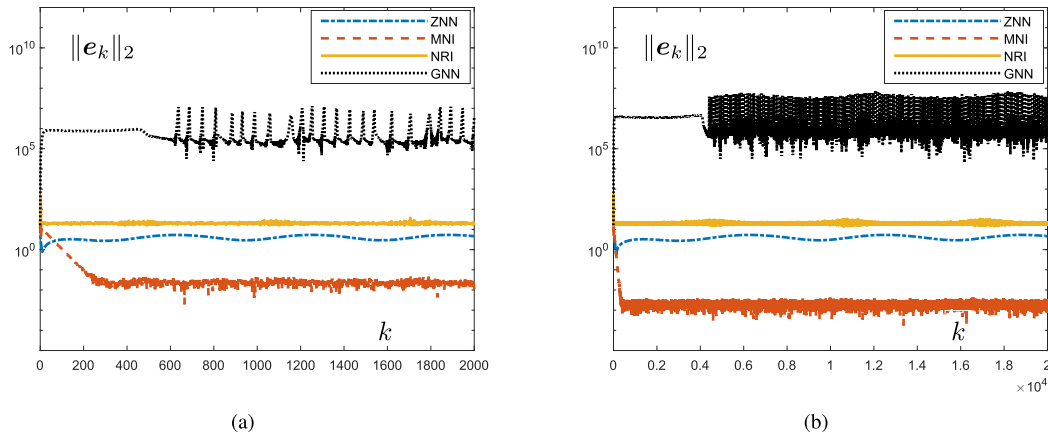


FIGURE 4. Random noise $q_k \in 10 + 2 \times [-1, 1]$ steady-state error convergence of four different algorithms, the proposed MNI algorithm (7), ZNN algorithm (9), NRI algorithm (4), and GNN algorithm (8). (a) $\tau = 0.01$ s. (b) $\tau = 0.001$ s.

TABLE 1. MSSRE and ACTPI (shown in parentheses) for the proposed MNI algorithm (7), ZNN algorithm (9), GNN algorithm (8), and NRI algorithm (4) for different sampling and noise environment.

τ (s)	Algorithm	MSSRE and ACTPI (s) in parentheses with different kinds of noises							
		Zero noise $q_k = 0$		Constant noise $q_k = 10$		Linear noise $q_k = 10 \times (k\tau + 1)$		Random noise $q_k \in 10 + 2 \times [-1, 1]$	
		MSSRE	ACTPI	MSSRE	ACTPI	MSSRE	ACTPI	MSSRE	ACTPI
0.01	ZNN (9)	5.7323×10^0	4.0114×10^{-6}	5.3600×10^0	1.5255×10^{-6}	1.3736×10^7	1.6458×10^{-6}	5.3965×10^0	1.5280×10^{-6}
	GNN (8)	1.3947×10^1	1.6570×10^{-6}	1.3413×10^6	9.8906×10^{-7}	NA ¹	8.8233×10^{-7}	1.5528×10^7	9.7581×10^{-7}
	MNI (7)	1.9211×10^{-2}	2.7447×10^{-6}	1.9211×10^{-2}	1.9696×10^{-6}	8.0229×10^0	1.9178×10^{-6}	4.8502×10^{-2}	1.8332×10^{-6}
	NRI (4)	4.3523×10^{-2}	1.8573×10^{-6}	2.0084×10^1	1.2704×10^{-6}	NA ¹	1.6850×10^{-6}	3.7662×10^1	1.2158×10^{-6}
0.001	ZNN (9)	5.7329×10^0	1.6216×10^{-5}	5.3606×10^0	1.5018×10^{-5}	1.2748×10^9	1.5314×10^{-5}	5.3687×10^0	1.5030×10^{-5}
	GNN (8)	1.3947×10^1	9.4135×10^{-6}	2.2402×10^8	9.1687×10^{-6}	NA ¹	8.6334×10^{-6}	6.4647×10^7	9.1958×10^{-6}
	MNI (7)	2.1951×10^{-4}	1.9635×10^{-5}	2.1952×10^{-4}	1.8748×10^{-5}	8.0012×10^{-1}	1.8665×10^{-5}	4.3088×10^{-3}	1.8685×10^{-5}
	NRI (4)	4.3655×10^{-3}	1.2195×10^{-5}	2.0008×10^1	1.1836×10^{-5}	NA ¹	1.4453×10^{-5}	3.6887×10^1	1.1689×10^{-5}

¹ "NA" denotes that this value tends to infinite.

4) *Random Noise Environment*: In real life, random noise is frequently encountered. Therefore, it is important to investigate the influence of four algorithms under the random noise environment. Because of the properties of random noise, it is possible for the MSSRE to take the form of an oscillation. Theorem 4 proves that its steady-state error $\|e\|_2$ tends to infinity when the number of iterations increases to infinity, which is confirmed in Fig. 4. In addition, Fig. 4 demonstrates the excellent noise tolerance of the proposed MNI algorithm. In terms of specific values, in Table 1 the MSSRE of the proposed MNI algorithm (7) in this paper is 4.8502×10^{-2} and 4.3088×10^{-3} for $\tau = 0.01$ s and $\tau = 0.001$ s, respectively, which is considerably higher than the MSSRE of the remaining algorithms. Although there are some oscillations of the MSSRE generated by the MNI algorithm (7) under random noise $q_k \in 10 + 2 \times [-1, 1]$ environment, it still conforms to the conclusion drawn in Theorem 4.

Based on the above experiments, the following conclusions can be derived. First of all, the proposed MNI algorithm (7) performs better under three different noise environments compared to the NRI algorithm (4), GNN algorithm (8), and ZNN algorithm (9). Second, MSSRE is the measure that reflects the accuracy of each algorithm in a zero noise environment. And the MNI algorithm (7) proposed in this paper possesses a low MSSRE. As a result, the proposed MNI

algorithm (7) not only possesses a high accuracy for solving the DAMRE (1), but also has the strong robustness against various noise perturbations.

V. CONCLUSION

In this paper, in order to solve the discrete algebraic matrix Riccati equation (DAMRE), a modified Newton integration algorithm (7) has been proposed from the control theory perspective, which differs from the conventional continuous-time algorithm to solve the DAMRE. Compared with the other three algorithms, the NRI algorithm (4), GNN algorithm (8), and ZNN algorithm (9), the proposed MNI algorithm possesses excellent noise tolerance ability under various noise environments. In general, the proposed MNI algorithm is not only more accurate than other algorithms, but also with a faster convergence speed. In the future, different matrices of the DAMRE will be tried to verify the validity of the proposed MNI algorithm. In the future, researchers will investigate some faster and higher precision methods to solve the DAMRE.

REFERENCES

- [1] V. Kučera, "The discrete Riccati equation of optimal control," *Kybernetika*, vol. 8, no. 5, pp. 430–447, 1972.
- [2] S. Lal and P. Kumari, "Approximation of functions with bounded derivative and solution of Riccati differential equations by Jacobi wavelet operational matrix," *Appl. Math. Comput.*, vol. 394, Apr. 2021, Art. no. 125834.

- [3] D. Vaughan, "A nonrecursive algebraic solution for the discrete Riccati equation," *IEEE Trans. Autom. Control*, vol. AC-15, no. 5, pp. 597–599, Oct. 1970.
- [4] M. G. Sakar, A. Akgül, and D. Baleanu, "On solutions of fractional Riccati differential equations," *Adv. Difference Equ.*, vol. 2017, no. 1, p. 39, Feb. 2017.
- [5] I. Ahmad, O. Bazighifan, A. Abouelregal, and H. Ahmad, "Multistage optimal homotopy asymptotic method for the nonlinear Riccati ordinary differential equation in nonlinear physics," *Appl. Math. Inf. Sci.*, vol. 14, no. 6, pp. 1–7, Jan. 2020.
- [6] H. Huang, D. Fu, X. Xiao, Y. Ning, H. Wang, L. Jin, and S. Liao, "Modified Newton integration neural algorithm for dynamic complex-valued matrix pseudoinversion applied to mobile object localization," *IEEE Trans. Ind. Inform.*, vol. 17, no. 4, pp. 2432–2442, Apr. 2021.
- [7] C.-H. Guo and A. J. Laub, "On a Newton-like method for solving algebraic Riccati equations," *SIAM J. Matrix Anal. Appl.*, vol. 21, no. 2, pp. 694–698, Jan. 2000.
- [8] F. Feitzinger, T. Hylla, and E. W. Sachs, "Inexact Kleinman–Newton method for Riccati equations," *SIAM J. Matrix Anal. Appl.*, vol. 31, no. 2, pp. 272–288, Jan. 2009.
- [9] P. Benner, M. Heinkenschloss, J. Saak, and H. K. Weichelt, "An inexact low-rank Newton–ADI method for large-scale algebraic Riccati equations," *Appl. Numer. Math.*, vol. 108, pp. 125–142, Oct. 2016.
- [10] A.-G. Wu, H.-J. Sun, and Y. Zhang, "Two iterative algorithms for stochastic algebraic Riccati matrix equations," *Appl. Math. Comput.*, vol. 339, pp. 410–421, Dec. 2018.
- [11] T. Nguyen and Z. Gajic, "Solving the matrix differential Riccati equation: A Lyapunov equation approach," *IEEE Trans. Autom. Control*, vol. 55, no. 1, pp. 191–194, Jan. 2010.
- [12] A.-G. Wu, H.-J. Sun, and Y. Zhang, "A novel iterative algorithm for solving coupled Riccati equations," *Appl. Math. Comput.*, vol. 364, Jan. 2020, Art. no. 124645.
- [13] V. Simoncini, "Analysis of the rational Krylov subspace projection method for large-scale algebraic Riccati equations," *SIAM J. Matrix Anal. Appl.*, vol. 37, no. 4, pp. 1655–1674, Jan. 2016.
- [14] F. T. Isfahani and R. Mokhtari, "A numerical approach based on the reproducing kernel Hilbert space for solving a class of boundary value optimal control problems," *Iranian J. Sci. Technol., Trans. A, Sci.*, vol. 42, no. 4, pp. 2309–2318, Dec. 2018.
- [15] O. A. Arqub, M. AL-Smadi, S. Momani, and T. Hayat, "Numerical solutions of fuzzy differential equations using reproducing kernel Hilbert space method," *Soft Comput.*, vol. 20, no. 8, pp. 3283–3302, Aug. 2016.
- [16] O. A. Arqub and B. Maayah, "Modulation of reproducing kernel Hilbert space method for numerical solutions of Riccati and Bernoulli equations in the Atangana–Baleanu fractional sense," *Chaos, Solitons Fractals*, vol. 125, pp. 163–170, Aug. 2019.
- [17] M. G. Sakar, "Iterative reproducing kernel Hilbert spaces method for Riccati differential equations," *J. Comput. Appl. Math.*, vol. 309, pp. 163–174, Jan. 2017.
- [18] D. Angluin and P. Laird, "Learning from noisy examples," *Mach. Learn.*, vol. 2, no. 4, pp. 343–370, 1988.
- [19] H. Huang, D. Fu, J. Zhang, X. Xiao, G. Wang, and S. Liao, "Modified Newton integration neural algorithm for solving the multi-linear M-tensor equation," *Appl. Soft Comput.*, vol. 96, Nov. 2020, Art. no. 106674.
- [20] L. Jin, S. Li, B. Liao, and Z. Zhang, "Zeroing neural networks: A survey," *Neurocomputing*, vol. 267, pp. 597–604, Dec. 2017.
- [21] X. Xiao, C. Jiang, H. Lu, L. Jin, D. Liu, H. Huang, and Y. Pan, "A parallel computing method based on zeroing neural networks for time-varying complex-valued matrix Moore–Penrose inversion," *Inf. Sci.*, vol. 524, pp. 216–228, Jul. 2020.
- [22] C. Jiang, X. Xiao, D. Liu, H. Huang, H. Xiao, and H. Lu, "Nonconvex and bound constraint zeroing neural network for solving time-varying complex-valued quadratic programming problem," *IEEE Trans. Ind. Inform.*, vol. 17, no. 10, pp. 6864–6874, Oct. 2021.
- [23] Y. Zhang, K. Chen, and H.-Z. Tan, "Performance analysis of gradient neural network exploited for online time-varying matrix inversion," *IEEE Trans. Autom. Control*, vol. 54, no. 8, pp. 1940–1945, Aug. 2009.
- [24] P. S. Stanimirović, V. N. Katsikis, and S. Li, "Hybrid GNN-ZNN models for solving linear matrix equations," *Neurocomputing*, vol. 316, pp. 124–134, Nov. 2018.
- [25] G. Wang, H. Huang, L. Shi, C. Wang, D. Fu, L. Jin, and X. Xiuchun, "A noise-suppressing Newton-raphson iteration algorithm for solving the time-varying Lyapunov equation and robotic tracking problems," *Inf. Sci.*, vol. 550, pp. 239–251, Mar. 2021.
- [26] M. A. Lohe, "Systems of matrix Riccati equations, linear fractional transformations, partial integrability and synchronization," *J. Math. Phys.*, vol. 60, no. 7, Jul. 2019, Art. no. 072701.
- [27] Y. Zhang, L. Jin, D. Guo, Y. Yin, and Y. Chou, "Taylor-type 1-step-ahead numerical differentiation rule for first-order derivative approximation and ZNN discretization," *J. Comput. Appl. Math.*, vol. 273, pp. 29–40, Jan. 2015.
- [28] M. Liu, H. Li, Y. Li, L. Jin, and Z. Huang, "From WASD to BLS with application to pattern classification," *Appl. Soft Comput.*, vol. 108, Sep. 2021, Art. no. 107455.
- [29] D. Fu, H. Huang, L. Wei, X. Xiao, L. Jin, S. Liao, J. Fan, and Z. Xie, "Modified Newton integration algorithm with noise tolerance applied to robotics," *IEEE Trans. Syst., Man, Cybern. Syst.*, early access, Jan. 21, 2021, doi: 10.1109/TSMC.2021.3049386.
- [30] M. Liu, L. Chen, X. Du, L. Jin, and M. Shang, "Activated gradients for deep neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Sep. 1, 2021, doi: 10.1109/TNNLS.2021.3106044.
- [31] M. Liu, B. Peng, and M. Shang, "Lower limb movement intention recognition for rehabilitation robot aided with projected recurrent neural network," *Complex Intell. Syst.*, Mar. 2021, doi: 10.1007/s40747-021-00341-w.
- [32] A. Aleksandrov and O. Mason, "Diagonal Riccati stability and applications," *Linear Algebra Appl.*, vol. 492, pp. 38–51, Mar. 2016.



SIYUAN LIAO is currently pursuing the B.E. degree in marine technology with the School of Electronics and Information Engineering, Guangdong Ocean University, Zhanjiang, China. His current research interests include deep learning and remote sensing image classification.



DONGYANG FU received the Ph.D. degree in physical oceanography from the South China Sea Institute of Oceanology, Chinese Academy of Sciences, Beijing, China, in 2010. He is currently a Professor with the School of Electronics and Information Engineering, Guangdong Ocean University, Zhanjiang, China. His current research interests include ocean color remote sensing and its application, remote sensing in off-shore water quality, response of upper ocean to typhoon, and neural networks.



HAOEN HUANG received the B.E. degree in electrical engineering and automation from Guangdong Ocean University, Zhanjiang, China, in 2019, where he is currently pursuing the M.Agr. degree in agricultural engineering and information technology with the School of Electronics and Information Engineering. His current research interests include Newton algorithm, neural networks, and robotics.



CHENGZE JIANG received the B.E. degree in software engineering from Guangdong Ocean University, Zhanjiang, China, in 2019, where he is currently pursuing the M.Agr. degree in agricultural engineering and information technology with the School of Electronics and Information Engineering. His current research interests include neural networks and computer vision.