Payame Noor University

Control and Optimization in Applied Mathematics (COAM)
Vol. 7, No. 2, Summer-Autumn 2022 (1-34), ©2016 Payame Noor University, Iran
DOI. 10.30473/COAM.2022.62848.1192 (Cited this article)

Research Article

Integrated Fault Detection and Robust Control for Linear Uncertain Switched Systems with Mode-Dependent Time-Varying State Delay

Sayyed Hossein Ejtahed[1], **Naser Pariz**[*1], **Ali Karimpour**[1]

[1]Department of Electrical Engineering, Faculty of Engineering, Ferdowsi University of Mashhad,
P.O. Box. 91779-48974, Mashhad, Iran.

Received: February 3, 2022; **Accepted:** July 17, 2022.

Abstract. Switched linear systems are noted as a major category of control systems. Fault detection of these systems is affected by switching phenomena and therefore their integrated fault detection and robust control (IFDRC) are the central issues of recent studies. Existing studies on IFDRC do not consider the effects of all of the parameter uncertainties, input disturbance, and mode-dependent time-varying state delay in the presence of mode-dependent average dwell time (MDADT) switching together in these systems. To address the issue based on output feedback, in this paper, the IFDRC design problem is formulated as a multi-objective or mixed H_∞/H_- optimization problem. H_∞ performance indicator guarantees the robustness of residual to disturbance, and H_- performance represents the sensitivity index of residual to the fault. A piecewise Lyapunov-Krasovskii function is employed together with the MDADT scheme and therefore, sufficient conditions are derived in terms of linear matrix inequalities (LMIs) to deal with the problem. Then to clarify the design procedure, we also present an algorithm in light of the proposed approach. Eventually, to illustrate the efficiency of the suggested approach, the designed IFDRC framework is simulated for a case study of an Electrical Circuit system.

Keywords. Integrated fault detection and control, Switched systems, Uncertainty, Variable state delay.

MSC. 34H05; 94C12; 94C10; 62F35.

* Corresponding author
ejtahed@stu.um.ac.ir, n-pariz@um.ac.ir, karimpor@um.ac.ir

1 Introduction

As an important class of hybrid systems, switched systems are a combination of multiple subsystems and a switching law. The switching law determines an active subsystem at the particular switching time instant. Many practical applications are modeled as switched systems, such as power electronics [36], Buck-Boost converter [10], Ball-and-Beam systems [12], flight control systems, etc. [28].

For real-world processes, fault detection (FD) has become more significant due to the increasing demand for the efficiency of supervision, safety, and reliability. Model-based FD methods have been widely used and developed over the past decades. This technique is to construct residuals based on some measured output signals of the system. The occurrence of faults is determined by comparing residuals in fault-free and faulty situations [39]. Many results have emerged from this topic for switched systems [23, 29, 37]. On the other hand, it is possible, and more importantly, desirable to consider a framework for integrating the design of fault diagnosis filters and feedback controllers. This simultaneous design unifies both control and diagnosis modules into an integrated unit. Therefore, it is unavoidable and certain that an integrated fault detection and control (IFDC) design technique should result in a far less general difficulty as compared to an approach where the two modules are designed separately [7]. Some techniques in the IFDC area are as follows: a method subject to a dwell time constraint [38], an approach based on Dynamic Observer [5], A Linear Matrix Inequality Approach [2], Average Dwell Time constraint [2, 10], and IFDC schemes under mode-dependent average dwell time constraint [33].

As a common phenomenon in many dynamic physical processes, the delay and parameter uncertainties may weaken the fault detection sensitivity and disturbance attenuation capability. Therefore, it is important to take into consideration the effect of state delay and parameter uncertainties for designing fault detection and control units under the presence of unknown inputs. Meanwhile, only a few papers have taken the state delay into account. Some of them have supposed it constant [27, 35, 39] and others have assumed it time-varying [19, 24]. Due to the complexity caused by the presence of parameter uncertainties, a few results on FD of switched delay systems with parameter uncertainties have been reported [21, 24]. As far as we know, there is a very limited number of research considering both the variable state delay and parameter uncertainties [24].

Innovation and the main contribution

In this paper, we investigate the problems of fault detection and robust control for switched linear systems in a general framework. Some documents in this field employ one of the below-mentioned five concepts separately, or at most a combination of two or three cases of them. To the best of our knowledge, the IFDRC design with a variety of these five items is not tackled yet for the switched systems. The main contribution of our work is to propose a general framework for designing IFDRC for the switched systems considering these concepts:

- MDADT: mode-dependent average dwell time,
- MDTVD: mode-dependent time-varying state delay,
- Parameter Uncertainty,
- Input disturbance,
- Mixed H_∞/H_-.

In this paper, the mode-dependent average dwell time (MDADT), which will release the restrictions of ADT, is used with mode-dependent time-varying (MDTV) state delay, and norm-bounded parameter uncertainties, and unknown input disturbances. Further, in an output feedback framework, sufficient conditions are derived and formulated for weighted H_∞ performance in terms of a set of linear matrix inequalities with MDADT switching to attenuate the disturbance of the corresponding switched linear systems. Sufficient conditions for weighted H_- performance to amplify fault sensitivity are also derived and developed in terms of a set of matrix inequalities. Based on the proposed scheme, the IFDRC problem is solved by the convex optimization technique, and the dynamic controller/detectors associated with the designed switching law are obtained such that the system with the mentioned constraint satisfies the indices.

The remainder of this paper is organized as follows: Section 2 presents the problem statement, necessary definitions, and preliminaries. It recalls the corresponding criterion and lemmas for the switched systems' fault detection and control with MDADT switching. In Section 3, the main results for mixed weighted H_∞/H_- integrated fault detection and robust control unit (IFDRCU) design for linear uncertain continuous-time switched systems with MDTV state delay and input disturbance under MDADT constraint design approaches are illustrated in detail by two theorems. The residual evaluation function and the threshold are provided. To demonstrate the effectiveness of the proposed method, an Electrical Circuit system is given as a numerical example in Section 4, followed by a conclusion in the last section.

Notations

In this paper, some standard notations are used. For a matrix A, A^T denotes its transpose. Here, $A > 0 (A >= 0)$ and $A < 0 (A <= 0)$ mean that the matrix is positive and negative (semi-)definite, respectively. The symbol $*$ used in a matrix denotes the terms which are readily inferred from symmetry. The Hermitian part of a square matrix A is denoted by $He(A) := A + A^T$. The values $\lambda_{\max}(A)$ and $\lambda_{\min}(A)$ are the maximum and minimum eigenvalues of A, respectively. $\mathbb{R}^n$ stands for the n-dimensional real vector space; where $\mathbb{R}^{n\times m}$ indicates the space of $n \times m$ matrices with real entries; $\|x\|^2 = x^T x = x_1^2 + \cdots + x_n^2$, where x_i is the i-th element of the vector, $x \in \mathbb{R}^n$; let $\underline{l} = \{1, \ldots, l\}$, where l is an arbitrary positive integer; $\mathbb{Z}^+$ implies the set of positive integers. l_2 stands for the 2-norm; 0 and I represent the zero and identity matrices with appropriate dimensions, respectively. $A^\perp$ is defined as an orthogonal basis for the null space of A while satisfying $A^\perp A = 0$.

2 Problem Statement and Preliminaries

In this section, problem formulation, necessary assumptions, definitions, lemmas, and IFDRC concepts are presented.

2.1 The main system model

Consider the following switched linear system with mode-dependent time-varying state delays and parameter uncertainty.

$$\begin{cases} \dot{x}(t) = A_{\sigma(t)}(t)x(t) + A_{d\,\sigma(t)}(t)x(t - d_{\sigma(t)}(t)) + B_{\sigma(t)}u(t) + B_{\omega\,\sigma(t)}(t)\omega(t) + B_{f\,\sigma(t)}(t)f(t), \\ y(t) = C_{\sigma(t)}(t)x(t) + D_{\omega\sigma(t)}(t)\omega(t) + D_{f\,\sigma(t)}(t)f(t), \\ x(\theta) = \phi(\theta), \quad \theta \in [-d, 0]. \end{cases} \tag{1}$$

Here, $x(t) \in \mathbb{R}^n$ is the state vector, $u(t) \in \mathbb{R}^m$ denotes the control input vector, $\omega(t) \in \mathbb{R}^r$ represents the bounded disturbance input, $f(t) \in \mathbb{R}^s$ is the fault signal, and $y(t) \in \mathbb{R}^q$ signifies the measured output vector. It is assumed that $\omega(t)$ and $f(t)$ belong to $L_2[0, \infty)$ and $\|\omega(t)\|_2 \leq \delta_\omega$, $\|f(t)\|_2 \leq \delta_f$, where δ_ω , δ_f are represented as known constants. $\phi(\theta)$ is the continuous vector-valued initial function on $[-d, 0]$. $\sigma(t) : [0, \infty) \to \underline{l}$ is a right continuous piecewise constant function that denotes the switching law and $l > 1$ is the number of subsystems. $\sigma(t) = i$ means that the i-th subsystem is activated at time t. If

$t \in [t_k, t_{k+1})$, then $\sigma(t) = \sigma(t_k)$. The duration time $[t_k, t_{k+1})$ is called the dwell time of the currently enabled subsystem. The value t_k represent the switching time instants and $t_0 < t_1 < \cdots < t_k$, $(k \in \mathbb{Z}^+)$ represents the switching time sequence of the switching signal. $A_i, A_{di}, B_i, B_{\omega i}, B_{fi}, C_i, D_{\omega i}$, and D_{fi} represent known real constant system matrices with appropriate dimensions. $d_i(t)$ stands for the mode-dependent time-varying delay in state variables, which is a continuous function satisfying $0 < d_i(t) < d_i < d$ and $\dot{d}_i(t) < \rho_i$, and d_i, d, ρ_i are known positive scalars.

Assumption 1. ([25]): For input matrices $B_i \in \mathbb{R}^{n\times m}$ with $(B_i) = m$, there exist nonsingular matrices T_i such that

$$T_i B_i = \begin{bmatrix} I \\ 0 \end{bmatrix}. \tag{2}$$

In general, for a specified B_i, the corresponding T_i is not unique. One of the matrices T_i is

$$T_i = \begin{bmatrix} (B_i^T B_i)^{-1} B_i^T \\ B_i^{\perp} \end{bmatrix}. \tag{3}$$

Also, the model uncertainties are as in (4) and $\Delta A_i, \Delta A_{di}, \Delta B_{\omega i}, \Delta B_{fi}, \Delta C_i, \Delta D_{\omega i}$, and ΔD_{fi} are norm-bounded matrices, and therefore, we obtain

$$\begin{cases} A_{\sigma(t)}(t) = A_{\sigma(t)} + \Delta A_{\sigma(t)}(t), \\ A_{d\,\sigma(t)}(t) = A_{d\,\sigma(t)} + \Delta A_{d\,\sigma(t)}(t), \\ B_{\sigma(t)} = B_{\sigma(t)}, \\ B_{\omega\,\sigma(t)}(t) = B_{\omega\,\sigma(t)} + \Delta B_{\omega\,\sigma(t)}(t), \\ B_{f\,\sigma(t)}(t) = B_{f\,\sigma(t)} + \Delta B_{f\,\sigma(t)}(t), \\ C_{\sigma(t)}(t) = C_{\sigma(t)} + \Delta C_{\sigma(t)}(t), \\ D_{\omega\,\sigma(t)}(t) = D_{\omega\,\sigma(t)} + \Delta D_{\omega\,\sigma(t)}(t), \\ D_{f\,\sigma(t)}(t) = D_{f\,\sigma(t)} + \Delta D_{f\,\sigma(t)}(t). \end{cases} \tag{4}$$

Assumption 2. ([11]): The parameter uncertainties are assumed to satisfy the following norm-bounded conditions:

$$\begin{bmatrix} \Delta A_i(t) & \Delta A_{di}(t) & \Delta B_{\omega i}(t) & \Delta B_{fi}(t) \\ \Delta C_i(t) & \Delta C_{di}(t) & \Delta D_{\omega i}(t) & \Delta D_{fi}(t) \end{bmatrix} = \begin{bmatrix} M_{i1} \\ M_{i2} \end{bmatrix} Q(t) \begin{bmatrix} N_{i1} & N_{i2} & N_{i3} & N_{i4} \end{bmatrix}, \tag{5}$$

where $M_{ij} (j = 1, 2)$ and $N_{ik} (k = 1, 2, 3, 4)$ are known real constant matrices and $Q(t) \in \mathbb{R}^{k\times k}$ is an unknown Lebesque-measurable real time-varying matrix subject to the following condition.

$$Q^T(t) Q(t) \le I, \tag{6}$$

for each t.

Remark 1. It is worth to be mentioned that a regulated output can also be considered for the main system in which both effects of fault and disturbance should be minimized on it to achieve a robust control objective. But for the sake of simplicity, it is ignored in this work [33, 40].

2.2 Integrated fault detection and robust control unit

To generate the control and the residual signal simultaneously, the integrated fault detection and robust control unit (IFDRCU) is employed, which integrates a fault detector and an output feedback controller within a switched linear system, as follows:

$$\begin{cases} \dot{x}_m(t) = A_{m\sigma(t)}x_m(t) + B_{m\sigma(t)}y(t), \\ r(t) = C_{m\sigma(t)}x_m(t) + D_{m\sigma(t)}y(t), \\ u(t) = K_{m\sigma(t)}x_m(t) + L_{m\sigma(t)}y(t), \end{cases} \tag{7}$$

where $x_m(t) \in \mathbb{R}^n$ represents the controller state vector and $r(t) \in \mathbb{R}^q$ is the residual signal. The matrices $A_{mi}, B_{mi}, C_{mi}, D_{mi}, K_{mi}$, and L_{mi} are the IFDRCU gains with appropriate dimensions, which should be determined.

Assumption 3. ([28]): The switching signal is not known beforehand, but it is assumed that it is determined instantaneously and IFDRCU switches synchronously with the main system. This is a common assumption in the literature. It is also considered that faults will not occur in the switching signal.

2.3 Closed-loop system description

Combining the aforementioned structures of the main system and the IFDRCU and defining the augmented state vector as $\varsigma^T(t) = [x^T(t)\ x_m^T(t)]$ to include filters state, the following augmented switched system is obtained:

$$\begin{cases} \dot{\varsigma}(t) = \bar{A}_{\sigma(t)}(t)\varsigma(t) + \bar{A}_{d\sigma(t)}(t)\varsigma(t - d_{\sigma(t)}(t)) + \bar{B}_{\omega\sigma(t)}(t)\omega(t) + \bar{B}_{f\sigma(t)}(t)f(t), \\ r(t) = \bar{C}_{\sigma(t)}(t)\varsigma(t) + \bar{D}_{\omega\sigma(t)}(t)\omega(t) + \bar{D}_{f\sigma(t)}(t)f(t), \end{cases} \tag{8}$$

where

$$\bar{A}_{\sigma(t)}(t) = \begin{bmatrix} A_{\sigma(t)}(t) + B_{\sigma(t)}L_{m\sigma(t)}C_{\sigma(t)}(t) & B_{\sigma(t)}K_{m\sigma(t)} \\ B_{m\sigma(t)}C_{\sigma(t)}(t) & A_{m\sigma(t)} \end{bmatrix},$$

$$\bar{A}_{d\sigma(t)}(t) = \begin{bmatrix} A_{d\sigma(t)}(t) & 0 \\ 0 & 0 \end{bmatrix},$$

$$
\begin{aligned}
\bar{B}_{\omega\sigma(t)}(t) &= \begin{bmatrix} B_{\omega\sigma(t)}(t) + B_{\sigma(t)}L_{m\sigma(t)}D_{\omega\sigma(t)}(t) \\ B_{m\sigma(t)}D_{\omega\sigma(t)}(t) \end{bmatrix}, \\
\bar{B}_{f\sigma(t)}(t) &= \begin{bmatrix} B_{f\sigma(t)}(t) + B_{\sigma(t)}L_{m\sigma(t)}D_{f\sigma(t)}(t) \\ B_{m\sigma(t)}D_{f\sigma(t)}(t) \end{bmatrix}, \\
\bar{C}_{\sigma(t)}(t) &= \begin{bmatrix} D_{m\sigma(t)}C_{\sigma(t)}(t) & C_{m\sigma(t)} \end{bmatrix}, \\
\bar{D}_{\omega\sigma(t)}(t) &= D_{m\sigma(t)}D_{\omega\sigma(t)}(t), \\
\bar{D}_{f\sigma(t)}(t) &= D_{m\sigma(t)}D_{f\sigma(t)}(t).
\end{aligned}
$$

2.4 The IFDRC design problem

In this section, the main problem is formulated as a multi-objective or mixed H_∞/H_- optimization problem. Therefore, our objective here is to design a switching law, a control signal, and a fault detection filter (see Figure 1) such that the exponential stability of the augmented switched system (8) is guaranteed with the specified mode-dependent average dwell time (MDADT). By setting the zero initial conditions, the effect of fault on the residual signal is maximized while the impact of disturbance is minimized on it considering the parameter uncertainties of the main system.

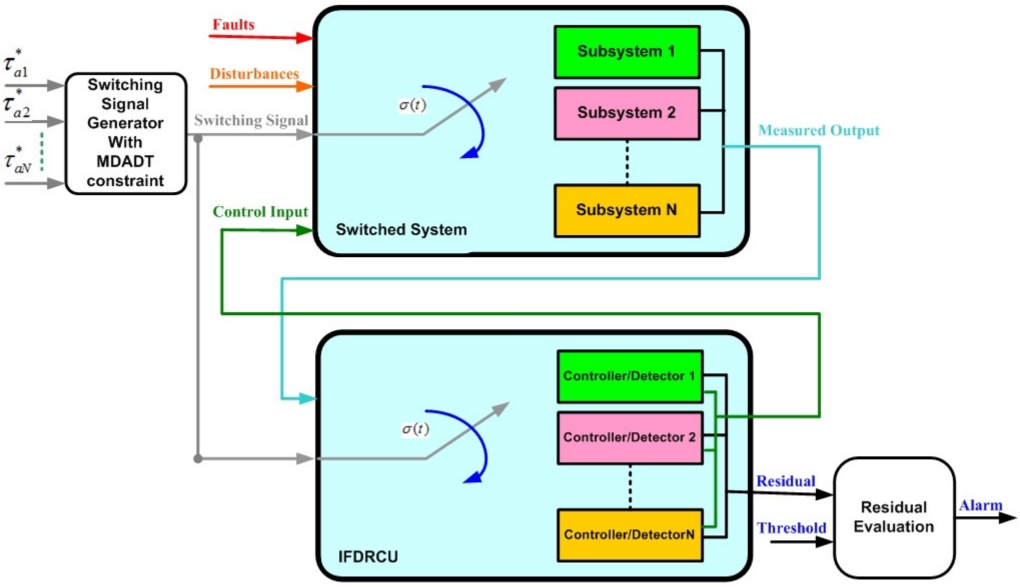


Figure 1: Switched system and Integrated Fault Detection & Robust Control Unit (IFDRCU).

2.4.1 Performance indices

For a given $\alpha_M > 0$, disturbance attenuation is characterized by the following weighted l_2 gain, which is called the weighted H_∞ performance index (α_M, γ_1) problem. It also ensures that the undetected faults are not disastrous.

$$\int_0^\infty e^{-\alpha_M t} r^T(t) r(t) dt \leq \gamma_1^2 \int_0^\infty \omega^T(t)\omega(t) dt. \tag{9}$$

Here, γ_1 is a prescribed level of disturbance attenuation. The smaller γ_1, the less affected the residual signal by disturbance.

Given $\alpha_m > 0$, fault sensitivity amplification is characterized by the following weighted l_2 gain, which is called the weighted H_- performance index (α_m, γ_2) problem.

$$\int_0^\infty r^T(t) r(t) dt \geq \gamma_2^2 \int_0^\infty e^{-\alpha_m t} f^T(t) f(t) dt. \tag{10}$$

Here, γ_2 is a prescribed level of fault sensitivity. The greater γ_2, the more sensitive to fault the residual signal.

Note that in general, α_M can be different from α_m.

Remark 2. The parameters α_M and α_m present the weighted l_2 gain index owing to the MDADT switching strategy. If $\alpha_M = \alpha_m$ is small enough which means that τ_a is selected sufficiently large, then the weighted l_2 gain approaches obviously the normal H_∞ problem. In fact, H_∞ performance is an unsolved problem for switched systems with the constraint of ADT, and therefore, a weighted H_∞ performance index should be utilized [4, 13]. Some claims in this area, such as those in [25], are not meaningful. In this work, we used a weighted H_∞/H_- performance index.

Remark 3. Some authors use a standard H_∞ model matching problem to change the H_- optimization problem into an H_∞ optimization problem by defining $r_e(t) = r(t) - f_w(t)$. This means that the residual signal, $r(t)$, robustly tracks a filtered version of the fault signal, i.e., $f_w(t)$. The filter $W(s)$ should be chosen appropriately as a stable transfer function. Since there is no straightforward method to determine this transfer function [8], the complexity is increased, and compared to those methods, our approach is more direct [33]. In some works, such as [24], $W(s)$ is defined as the filter, but the augmented system is not affected by the filter dynamics. In some other studies, like [2, 10], the use H_∞ problem is used instead of H_-, without defining $r_e(t) = r(t) - f_w(t)$.

2.4.2 Problem formulation

In this paper, the whole problem of IFDRC is transformed into the following mixed H_∞/H_- optimization problem. It is called a multi-objective or mixed optimization problem in the literature because it has two different objects and involves different norms [16, 20, 30].

$$\min_{\substack{\text{s.t.}\\(9),(10)}} c_1\gamma_1 - c_2\gamma_2. \tag{11}$$

In practice, the two scalars $c_1, c_2 \geq 0$ are used for a trade-off between the fault detection and control requirements. For example, if the H_∞ performance index is given, the relevant scalar $c_1 = 0$ [33].

2.5 Mathematical preliminaries

This section provides definitions and lemmas corresponding to the switched systems' fault detection and control with MDADT switching.

Definition 1. ([34]): For a switching signal $\sigma(t)$ and $\forall\, T \geq t \geq 0$, let $N_{\sigma i}(t,T)$ be the number of times that the i-th subsystem is activated on the interval $[t,T)$, and $T_i(t,T)$ present the total running time of the ith subsystem on the interval $[t,T),\, i \in \underline{l}$. If there exist positive numbers $N_{0i} \geq 0$ and $\tau_{ai} > 0$ such that

$$N_{\sigma i}(t,T) \leq N_{0i} + \frac{T_i(t,T)}{\tau_{ai}}, \tag{12}$$

for each $T \geq t \geq 0$, then we say that $\sigma(t)$ has a mode-dependent average dwell time, (MDADT),τ_{ai}, and the constant N_{0i} is called the mode-dependent chatter bound.

Remark 4. Although the constant N_{0i} should not be less than 2 in the case of average dwell time switching, it is usual in the literature to be assumed as zero for the sake of mathematical simplification [18]. In the sequel, we considered it not necessarily zero.

Definition 2. ([31]): Given scalars $\alpha > 0$ and $\gamma > 0$ the augmented system in (8) is said to be exponentially stable with weighted H_∞ performance (α, γ), if under $\sigma(t)$, it is exponentially stable with $\omega(t) = 0$, and under zero initial condition, that is, $\phi(\theta) = 0,\, \theta \in [-d, 0]$, for any non-zero $\omega(t) \in L_2[0,\infty)$, it holds that.

$$\int_0^\infty e^{-\alpha s} r^T(s) r(s) ds \leq \gamma^2 \int_0^\infty \omega^T(s)\omega(s) ds. \tag{13}$$

Lemma 1. ([3]): (Schur complement lemma) Let Y be a symmetric matrix of real numbers partitioned as follows and D be invertible. Then Y is positive definite if and only if D and its Schur complement, (Y/D), are both positive definite.

$$Y = \begin{bmatrix} A & B \\ B^T & D \end{bmatrix} \succ 0 \Leftrightarrow D \succ 0 \quad \text{and} \quad Y/D = A - BD^{-1}B^T \succ 0. \tag{14}$$

Lemma 2. For two given symmetric matrices, $\Phi, \tilde{\Phi} \in \mathbb{R}^{n\times n}$, where for each $i \neq j$

$$\Phi_{ii} \leq \tilde{\Phi}_{ii} \quad \text{and} \quad \Phi_{ij} = \tilde{\Phi}_{ij},$$

we have

$$\tilde{\Phi} \prec 0 \Rightarrow \Phi \prec 0. \tag{15}$$

Proof. Defining $\Lambda = (\lambda_1, \cdots, \lambda_n)$ *s.t.* $\lambda_i = \Phi_{ii} - \tilde{\Phi}_{ii} \leq 0$, from the assumption we have $\Phi - \tilde{\Phi} = \Lambda I_{n\times n} \prec 0$, therefore,

$$x^T \tilde{\Phi} x \prec 0 \Rightarrow x^T(\Phi - \Lambda I_{n\times n})x \prec 0 \Rightarrow x^T \Phi x - x^T \Lambda I_{n\times n} x \prec 0 \Rightarrow x^T \Phi x \prec 0.$$

□

Lemma 3. For a positive definite matrix $\Gamma \in \mathbb{R}^{n\times n}$, and any arbitrary symmetric matrix $\Lambda \in \mathbb{R}^{n\times n}$, we have

$$\Lambda \Gamma^{-1} \Lambda \geq 2\Lambda - \Gamma. \tag{16}$$

Proof. From the positive definiteness of Γ, it is clear that $x^T \Gamma^{-1} x \succ 0$. One can choose $x = (\Gamma - \Lambda)y$, therefore, $y^T(\Gamma - \Lambda)\Gamma^{-1}(\Gamma - \Lambda)y \succ 0$ which will result in

$$\begin{aligned} y^T(I_{n\times n} - \Lambda\Gamma^{-1})(\Gamma - \Lambda)y \succ 0 &\Rightarrow y^T(\Gamma - \Lambda - \Lambda\Gamma^{-1}\Gamma + \Lambda\Gamma^{-1}\Lambda)y \succ 0 \\ &\Rightarrow \Gamma - 2\Lambda + \Lambda\Gamma^{-1}\Lambda \succ 0. \end{aligned}$$

□

Lemma 4. ([41]): (Generalized square inequality lemma) If $X \in \mathbb{R}^{m\times n}, Y \in \mathbb{R}^{n\times m}, F \in \mathbb{R}^{n\times n}$, and F can be time-varying, then for arbitrary $\delta > 0$,

$$FF^T \leq I \Rightarrow He(XFY) \leq \delta XX^T + \delta^{-1} Y^T Y. \tag{17}$$

Lemma 5. ([2]): For two arbitrary scalars λ, κ, and two functions $\phi(t)$, and $\vartheta(t)$ satisfying

$$\dot{\phi}(t) \leq -\lambda\phi(t) + \kappa\vartheta(t), \tag{18}$$

we have

$$\phi(t) \leq e^{-\lambda(t-t_0)}\phi(t_0) + \kappa \int_{t_0}^{t} e^{-\lambda(t-\nu)}\vartheta(\nu)d\nu. \tag{19}$$

This inequality is a special case of the comparison lemma for integrals.

Lemma 6. ([15]): (Finsler's lemma) If $\Psi \in \mathbb{R}^{n\times n}, Z \in \mathbb{R}^{p\times n}$, where $rank(Z) < n$, then the inequality

$$Z^{\perp T} \Psi Z^{\perp} \prec 0, \tag{20}$$

is satisfied, if and only if there exists $X \in \mathbb{R}^{n\times p}$ such that

$$\Psi + He(XZ) \prec 0. \tag{21}$$

3 The Main Results

As stated in the previous section, the problem of IFDRC design for switched linear systems with mode-dependent time-varying state delay and parameter uncertainty can be formulated as a multi-objective or mixed H_∞/H_- optimization problem. In this section, we will drive sufficient conditions for analyzing the stability of the augmented system as well as obtaining fault detection and robust control objectives. These conditions will be addressed in LMIs forms.

3.1 The weighted H_∞ performance problem

In the following theorem, based on Definition 2 and the weighted H_∞ performance index (α_M, γ_1) defined in (9), sufficient conditions for the exponential stability of the augmented system in the presence of parameter uncertainties and input disturbances are derived. These conditions are in the form of LMIs. Then, an estimate of the state decay ratio is calculated. In addition, the minimum allowable average time for each subsystem to be active is calculated to satisfy the weighted H_∞ performance index (α_M, γ_1). Finally, the IFDRCU gains are determined.

Theorem 1. For given scalars $\alpha_M > 0$, $\mu_i^M \geq 1$, assume that there exist positive definite matrices $P_i \succ 0$, $R_i \succ 0, S_i \succ 0$ and appropriately-dimensioned real matrices $\widehat{A}_{mi}$, $\widehat{B}_{mi}$, C_{mi}, D_{mi}, $\widehat{K}_{mi}$, $\widehat{L}_{mi}, G_i, H_i = H_i^T$, as well as constant scalars $\gamma_{10} > 0$ and $\delta_{1i} > 0$ such that the following inequalities hold:

$$P_i \leq \mu_i^M P_j, R_i \leq \mu_i^M R_j, S_i \leq \mu_i^M S_j, \qquad i, j \in \underline{l}, \tag{22}$$

$$\Omega_{Mi} = \begin{bmatrix} \Phi_{Mi} & \Lambda_{Mi} \\ * & -\delta_{1i} I \end{bmatrix} \prec 0, \tag{23}$$

$$\Sigma_{Mi} = \begin{bmatrix} H_i & G_i \\ * & e^{-\alpha_M d_i} S_i \end{bmatrix} \succ 0, \tag{24}$$

where

$$\Phi_{Mi} = \begin{bmatrix} \Phi_{Mi11} & \Phi_{Mi12} & \Phi_{Mi13} & \Phi_{Mi14} & \Phi_{Mi15} \\ * & \Phi_{Mi22} & 0 & 0 & \Phi_{Mi25} \\ * & 0 & \Phi_{Mi33} & \Phi_{Mi34} & \Phi_{Mi35} \\ * & 0 & * & -I & 0 \\ * & * & * & 0 & S_i - 2P_i \end{bmatrix}, \tag{25}$$

$$\Lambda_{Mi}=\begin{bmatrix} P_{i1}M_{i1}+T_i^T\begin{bmatrix}\widehat{L}_{mi}\\0\end{bmatrix}M_{i2} & P_{i1}M_{i1} & 0\\ \widehat{B}_{mi}\,M_{i2} & 0 & 0\\ 0 & 0 & 0\\ 0 & 0 & 0\\ 0 & 0 & 0\\ D_{mi}M_{i2} & 0 & 0\\ \sqrt{d_i}P_{i1}M_{i1} & \sqrt{d_i}P_{i1}M_{i1} & \sqrt{d_i}T_i^T\begin{bmatrix}\widehat{L}_{mi}\\0\end{bmatrix}M_{i2}\\ 0 & 0 & \sqrt{d_i}\,\widehat{B}_{mi}\,M_{i2}\end{bmatrix} \tag{26}$$

$$\begin{aligned}\Phi_{Mi11}={}&He\left(\begin{bmatrix} P_{i1}A_i+T_i^T\begin{bmatrix}\widehat{L}_{mi}\\0\end{bmatrix}C_i+\delta_{1i}N_{i1}^TN_{i1} & T_i^T\begin{bmatrix}\widehat{K}_{mi}\\0\end{bmatrix}\\ \widehat{B}_{mi}\,C_i & \widehat{A}_{mi}\end{bmatrix}\right)\\ &+\alpha_MP_i+R_i+d_iH_{i1}+G_{i1}+G_{i1}^T, \end{aligned}\tag{27}$$

$$\Phi_{Mi12}=\begin{bmatrix}P_{i1}A_{di} & 0\\ 0 & 0\end{bmatrix}+d_iH_{i2}-G_{i1}+G_{i2}^T,$$

$$\Phi_{Mi13}=\begin{bmatrix}P_{i1}B_{\omega i}+T_i^T\begin{bmatrix}\widehat{L}_{mi}\\0\end{bmatrix}D_{\omega i}+2\delta_{1i}N_{i1}^TN_{i3}\\ \widehat{B}_{mi}\,D_{\omega i}\end{bmatrix},$$

$$\Phi_{Mi14}=\begin{bmatrix}C_i^TD_{mi}^T\\ C_{mi}^T\end{bmatrix},$$

$$\Phi_{Mi15}=\sqrt{d_i}\begin{bmatrix}A_i^TP_{i1}+C_i^T\begin{bmatrix}\widehat{L}_{mi}^T & 0\end{bmatrix}T_i & C_i^T\,\widehat{B}_{mi}^T\\ \begin{bmatrix}\widehat{K}_{mi}^T & 0\end{bmatrix}T_i & \widehat{A}_{mi}^T\end{bmatrix},$$

$$\Phi_{Mi22}=-(1-\rho_i)e^{-\alpha_Md_i}R_i+d_iH_{i3}-G_{i2}-G_{i2}^T+\delta_{1i}\begin{bmatrix}N_{i2}^TN_{i2} & 0\\ 0 & 0\end{bmatrix},$$

$$\Phi_{Mi25}=\sqrt{d_i}\begin{bmatrix}A_{di}^TP_{i1} & 0\\ 0 & 0\end{bmatrix},$$

$$\Phi_{Mi33}=-\gamma_{10}^2I+2\delta_{1i}N_{i3}^TN_{i3},$$

$$\Phi_{Mi34}=D_{\omega i}^TD_{mi}^T,$$

$$\Phi_{Mi35}=\sqrt{d_i}\begin{bmatrix}B_{\omega i}^TP_{i1}+D_{\omega i}^T\begin{bmatrix}\widehat{L}_{mi}^T & 0\end{bmatrix}T_i & D_{\omega i}^T\,\widehat{B}_{mi}^T\end{bmatrix}.$$

In these relations, M_{ij}, N_{ij} are defined in (5) and T_i's are defined in (3) and

$$G_i\triangleq\begin{bmatrix}G_{i1}^T & G_{i2}^T\end{bmatrix}^T, \tag{28}$$

$$H_i \triangleq \begin{bmatrix} H_{i1} & H_{i2} \\ * & H_{i3} \end{bmatrix}, \tag{29}$$

$$P_i = \begin{bmatrix} P_{i1} & 0 \\ 0 & P_{i2} \end{bmatrix}, \quad P_{i1} = T_i^T \begin{bmatrix} \overparen{P}_{i1} & 0 \\ 0 & \overparen{P}_{i2} \end{bmatrix} T_i. \tag{30}$$

Then the augmented system (8) is exponentially stable and satisfies the weighted H_∞ performance index (α_M, γ_1) in (9) for any switching signal with MDADT met by (31)

$$\tau_{ai}^M > \tau_{ai}^{M*} = \frac{\ln \mu_i^M}{\alpha_M}. \tag{31}$$

Finally, the IFDRCU matrices will be calculated as

$$A_{mi} = P_{i2}^{-1}\,\overparen{A}_{mi}, \qquad B_{mi} = P_{i2}^{-1}\,\overparen{B}_{mi}, \qquad K_{mi} = \overparen{P}_{i1}^{-1}\,\overparen{K}_{mi}, \qquad L_{mi} = \overparen{P}_{i1}^{-1}\,\overparen{L}_{mi}\,. \tag{32}$$

Proof. Construct the following Lyapunov-Krasovskii functional (LKF) candidate:

$$\begin{aligned} V(\varsigma_t, \sigma) &\triangleq V_1(\varsigma_t, \sigma) + V_2(\varsigma_t, \sigma) + V_3(\varsigma_t, \sigma), \\ V_1(\varsigma_t, \sigma) &\triangleq \varsigma^T(t) P_\sigma \varsigma(t), \\ V_2(\varsigma_t, \sigma) &\triangleq \int_{t-d_\sigma(t)}^{t} e^{\alpha_M(s-t)} \varsigma^T(s) R_\sigma \varsigma(s)\, ds, \\ V_3(\varsigma_t, \sigma) &\triangleq \int_{-d_\sigma}^{0} \int_{t+\theta}^{t} e^{\alpha_M(s-t)} \dot{\varsigma}^T(s) S_\sigma \dot{\varsigma}(s)\, ds\, d\theta, \end{aligned} \tag{33}$$

where real matrices $P_\sigma \succ 0, R_\sigma \succ 0$, and $S_\sigma \succ 0$ should be determined.

By calculating the derivative of LKF along with the solution of the augmented system and using the Leibniz integral rule for differentiation under the integral sign, we have:

$$\begin{aligned} \dot{V}(\varsigma_t, \sigma) + \alpha_M V(\varsigma_t, \sigma) = {} & 2\varsigma^T(t) P_\sigma \dot{\varsigma}(t) \\ & - (1 - \dot{d}_\sigma(t)) e^{-\alpha_M d_\sigma(t)} \varsigma^T(t - d_\sigma(t)) R_\sigma \varsigma(t - d_\sigma(t)) \\ & + \varsigma^T(t)(\alpha_M P_\sigma + R_\sigma)\varsigma(t) + d_\sigma \dot{\varsigma}^T(t) S_\sigma \dot{\varsigma}(t) \\ & - \int_{t-d_\sigma}^{t} e^{\alpha_M(s-t)} \dot{\varsigma}^T(s) S_\sigma \dot{\varsigma}(s)\, ds, \end{aligned} \tag{34}$$

and note that

$$-\int_{t-d_\sigma}^{t} e^{\alpha_M(s-t)} \dot{\varsigma}^T(s) S_\sigma \dot{\varsigma}(s)\, ds \le -\int_{t-d_\sigma(t)}^{t} e^{-\alpha_M d_\sigma} \dot{\varsigma}^T(s) S_\sigma \dot{\varsigma}(s)\, ds, \tag{35}$$

$$-(1 - \dot{d}_\sigma(t)) e^{-\alpha_M d_\sigma(t)} \le -(1 - \rho_\sigma) e^{-\alpha_M d_\sigma} \le -(1 - \rho) e^{-\alpha_M d}. \tag{36}$$

It is obvious that

$$\begin{aligned}\dot{V}(\varsigma_t,\sigma)+\alpha_M V(\varsigma_t,\sigma) \le{}& 2\varsigma^T(t)P_\sigma\,\dot{\varsigma}(t)\\ &-(1-\rho_\sigma)e^{-\alpha_M d_\sigma}\,\varsigma^T(t-d_\sigma(t))R_\sigma\,\varsigma(t-d_\sigma(t))\\ &+\varsigma^T(t)(\alpha_M P_\sigma+R_\sigma)\varsigma(t)+d_\sigma\,\dot{\varsigma}^T(t)S_\sigma\,\dot{\varsigma}(t)\\ &-\int_{t-d_\sigma(t)}^{t} e^{-\alpha_M d_\sigma}\dot{\varsigma}^T(s)S_\sigma\,\dot{\varsigma}(s)\,ds. \end{aligned}\tag{37}$$

Regarding (9), we define $I_\infty(t)\triangleq r^T(t)r(t)-\gamma_{10}^2\omega^T(t)\omega(t)$, and

$$\begin{aligned}\dot{V}(\varsigma_t,\sigma)+\alpha_M V(\varsigma_t,\sigma)+\mathrm{I}_\infty(t) \le{}& 2\varsigma^T(t)P_\sigma\,\dot{\varsigma}(t)\\ &-(1-\rho_\sigma)e^{-\alpha_M d_\sigma}\,\varsigma^T(t-d_\sigma(t))R_\sigma\,\varsigma(t-d_\sigma(t))\\ &+\varsigma^T(t)(\alpha_M P_\sigma+R_\sigma)\varsigma(t)+d_\sigma\dot{\varsigma}^T(t)S_\sigma\,\dot{\varsigma}(t)\\ &-\int_{t-d_\sigma(t)}^{t} e^{-\alpha_M d_\sigma}\dot{\varsigma}^T(s)S_\sigma\,\dot{\varsigma}(s)\,ds\\ &+r^T(t)r(t)-\gamma_{10}^2\omega^T(t)\omega(t). \end{aligned}\tag{38}$$

Substituting the derivative of the state vector from equation (8) for $f(t)=0$ we find that

$$\begin{aligned}\dot{V}(\varsigma_t,\sigma)+\alpha_M V(\varsigma_t,\sigma)+\mathrm{I}_\infty(t) \le{}& \varsigma_1^T(t,\sigma)\Theta_\sigma(t)\,\varsigma_1(t,\sigma)\\ &-\int_{t-d_\sigma(t)}^{t} e^{-\alpha_M d_\sigma}\dot{\varsigma}^T(s)S_\sigma\,\dot{\varsigma}(s)\,ds, \end{aligned}\tag{39}$$

where

$$\begin{aligned}\varsigma_1(t,\sigma)&\triangleq\begin{bmatrix}\varsigma^T(t) & \varsigma^T(t-d_\sigma(t)) & \omega^T(t)\end{bmatrix}^T,\\ \Theta_\sigma(t)&=\begin{bmatrix}\Theta_{\sigma11}(t) & \Theta_{\sigma12}(t) & \Theta_{\sigma13}(t)\\ * & \Theta_{\sigma22}(t) & \Theta_{\sigma23}(t)\\ * & * & \Theta_{\sigma33}(t)\end{bmatrix},\\ \Theta_{\sigma11}(t)&=He(P_\sigma\bar{A}_\sigma(t))+\alpha_M P_\sigma+R_\sigma+d_\sigma\bar{A}_\sigma^T(t)S_\sigma\bar{A}_\sigma(t)+\bar{C}_\sigma^T(t)\bar{C}_\sigma(t),\\ \Theta_{\sigma12}(t)&=P_\sigma\bar{A}_{d\sigma}(t)+d_\sigma\bar{A}_\sigma^T(t)S_\sigma\bar{A}_{d\sigma}(t),\\ \Theta_{\sigma13}(t)&=P_\sigma\bar{B}_{\omega\sigma}(t)+d_\sigma\bar{A}_\sigma^T(t)S_\sigma\bar{B}_{\omega\sigma}(t)+\bar{C}_\sigma^T(t)\bar{D}_{\omega\sigma}(t),\\ \Theta_{\sigma22}(t)&=-(1-\rho_\sigma)e^{-\alpha_M d_\sigma}R_\sigma+d_\sigma\bar{A}_{d\sigma}^T(t)S_\sigma\bar{A}_{d\sigma}(t),\\ \Theta_{\sigma23}(t)&=d_\sigma\bar{A}_{d\sigma}^T(t)S_\sigma\bar{B}_{\omega\sigma}(t),\\ \Theta_{\sigma33}(t)&=d_\sigma\bar{B}_{\omega\sigma}^T(t)S_\sigma\bar{B}_{\omega\sigma}(t)+\bar{D}_{\omega\sigma}^T(t)\bar{D}_{\omega\sigma}(t)-\gamma_{10}^2 I. \end{aligned}\tag{40}$$

Defining $\varsigma_2(t,\sigma)\triangleq\begin{bmatrix}\varsigma^T(t) & \varsigma^T(t-d_\sigma(t))\end{bmatrix}^T$ and $H_\sigma\triangleq\begin{bmatrix}H_{\sigma1} & H_{\sigma2}\\ * & H_{\sigma3}\end{bmatrix}$, we obtain

$$\int_{t-d_\sigma(t)}^{t}\varsigma_2^T(t,\sigma)H_\sigma\,\varsigma_2(t,\sigma)\,ds\le d_\sigma\varsigma_2^T(t,\sigma)H_\sigma\,\varsigma_2(t,\sigma).\tag{41}$$

By the Newton-Leibniz formula, for any arbitrary matrices $G_\sigma \triangleq \begin{bmatrix} G_{\sigma 1}^T & G_{\sigma 2}^T \end{bmatrix}^T$, we have

$$\varsigma_2^T(t,\sigma)G_\sigma \left[\varsigma(t) - \varsigma(t-d_\sigma(t)) - \int_{t-d_\sigma(t)}^{t} \dot{\varsigma}(s)\,ds\right] = 0. \tag{42}$$

Suppose that the Lyapunov matrix P_σ can be considered as a block-diagonal matrix such that in (30), by T_σ as defined in (3) we obtain

$$\begin{aligned}
&P_{\sigma 1}B_\sigma = T_\sigma^T \begin{bmatrix} \overset{\frown}{P}_{\sigma 1} & 0 \\ 0 & \overset{\frown}{P}_{\sigma 2} \end{bmatrix} T_\sigma B_\sigma = T_\sigma^T \begin{bmatrix} \overset{\frown}{P}_{\sigma 1} & 0 \\ 0 & \overset{\frown}{P}_{\sigma 2} \end{bmatrix}\begin{bmatrix} I \\ 0 \end{bmatrix} = T_\sigma^T \begin{bmatrix} \overset{\frown}{P}_{\sigma 1} \\ 0 \end{bmatrix}, \\
&P_{\sigma 1}B_\sigma L_{m\sigma} = T_\sigma^T \begin{bmatrix} \overset{\frown}{P}_{\sigma 1} \\ 0 \end{bmatrix} L_{m\sigma} = T_\sigma^T \begin{bmatrix} \overset{\frown}{P}_{\sigma 1}\, L_{m\sigma} \\ 0 \end{bmatrix} \triangleq T_\sigma^T \begin{bmatrix} \overset{\frown}{L}_{m\sigma} \\ 0 \end{bmatrix}, \\
&P_{\sigma 1}B_\sigma K_{m\sigma} = T_\sigma^T \begin{bmatrix} \overset{\frown}{P}_{\sigma 1} \\ 0 \end{bmatrix} K_{m\sigma} = T_\sigma^T \begin{bmatrix} \overset{\frown}{P}_{\sigma 1}\, K_{m\sigma} \\ 0 \end{bmatrix} \triangleq T_\sigma^T \begin{bmatrix} \overset{\frown}{K}_{m\sigma} \\ 0 \end{bmatrix}, \\
&\overset{\frown}{A}_{m\sigma} \triangleq P_{\sigma 2}A_{m\sigma}, \qquad \overset{\frown}{B}_{m\sigma} \triangleq P_{\sigma 2}B_{m\sigma}.
\end{aligned} \tag{43}$$

By combining (39), (41) and (42), we can write

$$\begin{aligned}
\dot{V}(\varsigma_t,\sigma) + \alpha_M V(\varsigma_t,\sigma) + I_\infty(t) \le\ & \varsigma_1^T(t,\sigma)\Pi_\sigma(t)\varsigma_1(t,\sigma) \\
& - \int_{t-d_\sigma(t)}^{t} \varsigma_3^T(t,s,\sigma)\Sigma_{M\sigma}\,\varsigma_3(t,s,\sigma)\,ds,
\end{aligned} \tag{44}$$

where $\Sigma_{M\sigma}$ is defined in (24) and

$$\varsigma_3(t,s,\sigma) \triangleq \begin{bmatrix} \varsigma^T(t) & \varsigma^T(t-d_\sigma(t)) & \dot{\varsigma}^T(s) \end{bmatrix}^T, \tag{45}$$

$$\Pi_\sigma(t) \triangleq \begin{bmatrix} \Pi_{\sigma 11}(t) & \Pi_{\sigma 12}(t) & \Pi_{\sigma 13}(t) \\ * & \Pi_{\sigma 22}(t) & \Pi_{\sigma 23}(t) \\ * & * & \Pi_{\sigma 33}(t) \end{bmatrix}, \tag{46}$$

$$\begin{aligned}
\Pi_{\sigma 11}(t) =\ & He\left(\begin{bmatrix} P_{\sigma 1}A_\sigma(t) + T_\sigma^T \begin{bmatrix} \overset{\frown}{L}_{m\sigma} \\ 0 \end{bmatrix} C_\sigma(t) & T_\sigma^T \begin{bmatrix} \overset{\frown}{K}_{m\sigma} \\ 0 \end{bmatrix} \\ \overset{\frown}{B}_{m\sigma}\, C_\sigma(t) & \overset{\frown}{A}_{m\sigma} \end{bmatrix}\right) \\
& + \alpha_M P_\sigma + R_\sigma + d_\sigma H_{\sigma 1} + G_{\sigma 1} + G_{\sigma 1}^T + d_\sigma \bar{A}_\sigma^T(t)S_\sigma\bar{A}_\sigma(t) + \bar{C}_\sigma^T(t)\bar{C}_\sigma(t), \\
\Pi_{\sigma 12}(t) =\ & \begin{bmatrix} P_{\sigma 1}A_{d\sigma}(t) & 0 \\ 0 & 0 \end{bmatrix} + d_\sigma \bar{A}_\sigma^T(t)S_\sigma \bar{A}_{d\sigma}(t) + d_\sigma H_{\sigma 2} - G_{\sigma 1} + G_{\sigma 2}^T, \\
\Pi_{\sigma 13}(t) =\ & \begin{bmatrix} P_{\sigma 1}B_{\omega\sigma}(t) + T_\sigma^T \begin{bmatrix} \overset{\frown}{L}_{m\sigma} \\ 0 \end{bmatrix} D_{\omega\sigma}(t) \\ \overset{\frown}{B}_{m\sigma}\, D_{\omega\sigma}(t) \end{bmatrix} + d_\sigma \bar{A}_\sigma^T(t)S_\sigma \bar{B}_{\omega\sigma}(t) + \bar{C}_\sigma^T(t)\bar{D}_{\omega\sigma}(t),
\end{aligned}$$

$$\begin{aligned}
&\Pi_{\sigma 22}(t) = -(1-\rho_\sigma)e^{-\alpha_M d_\sigma} R_\sigma + d_\sigma \bar{A}^T_{d\sigma}(t) S_\sigma \bar{A}_{d\sigma}(t) + d_\sigma H_{\sigma 3} - G_{\sigma 2} - G^T_{\sigma 2},\\
&\Pi_{\sigma 23}(t) = d_\sigma \bar{A}^T_{d\sigma}(t) S_\sigma \bar{B}_{\omega\sigma}(t),\\
&\Pi_{\sigma 33}(t) = d_\sigma \bar{B}^T_{\omega\sigma}(t) S_\sigma \bar{B}_{\omega\sigma}(t) + \bar{D}^T_{\omega\sigma}(t)\bar{D}_{\omega\sigma}(t) - \gamma_{10}^2 I. \qquad (47)
\end{aligned}$$

From (44), it is clear that $\dot{V}(\varsigma_t,\sigma)+\alpha_M V(\varsigma_t,\sigma)+\mathrm{I}_\infty(t) \leq 0$ if $\Pi_\sigma(t) \prec 0$ and $\Sigma_{M\sigma} \succ 0$. By applying the Schur complement lemma 1, i.e., (14) to the inequality $\Pi_\sigma(t) \prec 0$, and using Lemmas 2, 28 with $\Lambda = (0,0,0,2P_\sigma - S_\sigma - P_\sigma S_\sigma^{-1} P_\sigma)$, this inequality can be substituted by $\Xi_\sigma(t) \prec 0$. Then, by considering uncertainties in system parameters defined in (4), which cause system matrices to be time-dependent, we can separate $\Xi_\sigma(t)$ to

$$\Xi_\sigma(t) = \Xi_\sigma + \Delta\Xi_\sigma(t) \prec 0, \qquad (48)$$

where

$$\Xi_\sigma \triangleq \begin{bmatrix} \Xi_{\sigma 11} & \Xi_{\sigma 12} & \Xi_{\sigma 13} & \Xi_{\sigma 14} & \Xi_{\sigma 15}\\ * & \Xi_{\sigma 22} & 0 & 0 & \Xi_{\sigma 25}\\ * & 0 & -\gamma_{10}^2 I & \Xi_{\sigma 34} & \Xi_{\sigma 35}\\ * & 0 & * & -I & 0\\ * & * & * & 0 & S_\sigma - 2P_\sigma \end{bmatrix},$$

$$\begin{aligned}
\Xi_{\sigma 11} &= He\left(\begin{bmatrix} P_{\sigma 1} A_\sigma + T_\sigma^T \begin{bmatrix} \overset{\frown}{L}_{m\sigma} \\ 0 \end{bmatrix} C_\sigma & T_\sigma^T \begin{bmatrix} \overset{\frown}{K}_{m\sigma} \\ 0 \end{bmatrix} \\ \overset{\frown}{B}_{m\sigma}\, C_\sigma & \overset{\frown}{A}_{m\sigma} \end{bmatrix}\right)\\
&\quad + \alpha_M P_\sigma + R_\sigma + d_\sigma H_{\sigma 1} + G_{\sigma 1} + G^T_{\sigma 1},\\
\Xi_{\sigma 12} &= \begin{bmatrix} P_{\sigma 1} A_{d\sigma} & 0 \\ 0 & 0 \end{bmatrix} + d_\sigma H_{\sigma 2} - G_{\sigma 1} + G^T_{\sigma 2},\\
\Xi_{\sigma 13} &= \begin{bmatrix} P_{\sigma 1} B_{\omega\sigma} + T_\sigma^T \begin{bmatrix} \overset{\frown}{L}_{m\sigma} \\ 0 \end{bmatrix} D_{\omega\sigma} \\ \overset{\frown}{B}_{m\sigma}\, D_{\omega\sigma} \end{bmatrix},\\
\Xi_{\sigma 14} &= \begin{bmatrix} C_\sigma^T D^T_{m\sigma} \\ C^T_{m\sigma} \end{bmatrix},\\
\Xi_{\sigma 15} &= \sqrt{d_\sigma}\begin{bmatrix} A_\sigma^T P_{\sigma 1} + C_\sigma^T \begin{bmatrix} \overset{\frown}{L}^T_{m\sigma} & 0 \end{bmatrix} T_\sigma & C_\sigma^T\, \overset{\frown}{B}^T_{m\sigma} \\ \begin{bmatrix} \overset{\frown}{K}^T_{m\sigma} & 0 \end{bmatrix} T_\sigma & \overset{\frown}{A}^T_{m\sigma} \end{bmatrix},\\
\Xi_{\sigma 22} &= -(1-\rho_\sigma)e^{-\alpha_M d_\sigma} R_\sigma + dH_{\sigma 3} - G_{\sigma 2} - G^T_{\sigma 2},\\
\Xi_{\sigma 25} &= \sqrt{d_\sigma}\begin{bmatrix} A^T_{d\sigma} P_{\sigma 1} & 0 \\ 0 & 0 \end{bmatrix},\\
\Xi_{\sigma 34} &= D^T_{\omega\sigma} D^T_{m\sigma},
\end{aligned}$$

$$
\begin{aligned}
\Xi_{\sigma 35} &= \sqrt{d_\sigma}\begin{bmatrix} B^T_{\omega\sigma} P_{\sigma 1} + D^T_{\omega\sigma}\begin{bmatrix} \overset{\frown}{L}^T_{m\sigma} & 0 \end{bmatrix} T_\sigma & D^T_{\omega\sigma}\,\overset{\frown}{B}^T_{m\sigma} \end{bmatrix},\\
\Delta\Xi_{\sigma 11}(t) &= He(\begin{bmatrix} P_{\sigma 1}\Delta A_\sigma(t) + T^T_\sigma \begin{bmatrix} \overset{\frown}{L}_{m\sigma} \\ 0 \end{bmatrix} \Delta C_\sigma(t) & 0 \\ \overset{\frown}{B}_{m\sigma}\, \Delta C_\sigma(t) & 0 \end{bmatrix}),\\
\Delta\Xi_{\sigma 12}(t) &= \begin{bmatrix} P_{\sigma 1}\Delta A_{d\sigma}(t) & 0 \\ 0 & 0 \end{bmatrix},\\
\Delta\Xi_{\sigma 13}(t) &= \begin{bmatrix} P_{\sigma 1}\Delta B_{\omega\sigma}(t) + T^T_\sigma \begin{bmatrix} \overset{\frown}{L}_{m\sigma} \\ 0 \end{bmatrix} \Delta D_{\omega\sigma}(t) \\ \overset{\frown}{B}_{m\sigma}\, \Delta D_{\omega\sigma}(t) \end{bmatrix},\\
\Delta\Xi_{\sigma 14}(t) &= \begin{bmatrix} \Delta C^T_\sigma(t) D^T_{m\sigma} \\ 0 \end{bmatrix},\\
\Delta\Xi_{\sigma 15}(t) &= \sqrt{d_\sigma}\begin{bmatrix} \Delta A^T_\sigma(t) P_{\sigma 1} + \Delta C^T_\sigma(t)\begin{bmatrix} \overset{\frown}{L}^T_{m\sigma} & 0 \end{bmatrix} T_\sigma & \Delta C^T_\sigma(t)\,\overset{\frown}{B}^T_{m\sigma} \\ 0 & 0 \end{bmatrix},\\
\Delta\Xi_{\sigma 25}(t) &= \sqrt{d_\sigma}\begin{bmatrix} \Delta A^T_{d\sigma}(t) P_{\sigma 1} & 0 \\ 0 & 0 \end{bmatrix},\\
\Delta\Xi_{\sigma 34}(t) &= \Delta D^T_{\omega\sigma}(t) D^T_{m\sigma},\\
\Delta\Xi_{\sigma 35}(t) &= \sqrt{d_\sigma}\begin{bmatrix} \Delta B^T_{\omega\sigma}(t) P_{\sigma 1} + \Delta D^T_{\omega\sigma}(t)\begin{bmatrix} \overset{\frown}{L}^T_{m\sigma} & 0 \end{bmatrix} T_\sigma & \Delta D^T_{\omega\sigma}(t)\,\overset{\frown}{B}^T_{m\sigma} \end{bmatrix}.
\end{aligned}
$$

Referring to Assumption 2, we get

$$\Delta\Xi_\sigma(t) = He(\Lambda_{M\sigma}(Q(t), Q(t), Q(t))\Gamma_{M\sigma}), \tag{49}$$

where $\Lambda_{M\sigma}$ is defined in (26), and

$$\Gamma_{M\sigma} \triangleq \begin{bmatrix} N_{\sigma 1} & 0 & 0 & 0 & N_{\sigma 3} & 0 & 0 & 0 \\ 0 & 0 & N_{\sigma 2} & 0 & 0 & 0 & 0 & 0 \\ N_{\sigma 1} & 0 & 0 & 0 & N_{\sigma 3} & 0 & 0 & 0 \end{bmatrix}, \tag{50}$$

and by using the generalized square inequality in Lemma 4, that is (17), we get

$$\Delta\Xi_\sigma(t) \leq \delta^{-1}_{1\sigma}\Lambda_{M\sigma}\Lambda^T_{M\sigma} + \delta_{1\sigma}\Gamma^T_{M\sigma}\Gamma_{M\sigma}. \tag{51}$$

Now according to (51), inequality (48) can be rearranged to

$$(\Xi_\sigma + \delta_{1\sigma}\Gamma^T_{M\sigma}\Gamma_{M\sigma}) + \delta^{-1}_{1\sigma}\Lambda_{M\sigma}\Lambda^T_{M\sigma} \leq 0, \tag{52}$$

and by the new variable $\Phi_{M\sigma} \triangleq \Xi_\sigma + \delta_{1\sigma}\Gamma^T_{M\sigma}\Gamma_{M\sigma}$, we have

$$\Phi_{M\sigma} + \delta^{-1}_{1\sigma}\Lambda_{M\sigma}\Lambda^T_{M\sigma} \leq 0, \tag{53}$$

where $\Phi_{M\sigma}$ is defined in (25). Finally, by using the Schur complement in Lemma 1, that is (14), inequality (53) turns to (23).

Also, from (43), it is apparent that we can calculate IFDRCU parameters and get (32).

At this point, we will prove the exponential stability of the augmented system (8) with $\omega(t)=0, f(t)=0$ and without parameter uncertainties. If (23) and (24) are held, then from (44), we have

$$\dot{V}(\varsigma_t,\sigma) < -\alpha_M V(\varsigma_t,\sigma) - r^T(t)r(t) < -\alpha_M V(\varsigma_t,\sigma). \tag{54}$$

Using Lemma 5, and by integrating (54) from t_k to t we get:

$$V(\varsigma_{t_k},\sigma) \leq e^{-\alpha_M(t-t_k)} V(\varsigma_{t_k},\sigma(t_k)), \tag{55}$$

where t_k is the switching time instant. Using (22) at instant t_k, we have

$$V(\varsigma_{t_k},\sigma(t_k)) \leq \mu_{t_k}^M V(\varsigma_{t_k^-},\sigma(t_k^-)). \tag{56}$$

It follows from (55), (56), and (12) that

$$\begin{aligned} V(\varsigma_t,\sigma) &\leq \mu_{t_k}^M\, e^{-\alpha_M(t-t_k)}\, V(\varsigma_{t_k^-},\sigma(t_k^-)) \leq \cdots \\ &\leq \prod_{j=1}^{N_\sigma(t_0,t)} \mu_{\sigma(t_j)}^M e^{-\alpha_M(t-t_0)}\, V(\varsigma_{t_0},\sigma(t_0)) \\ &\leq e^{\sum_{p=1}^{l} N_{0p}\ln\mu_p^M}\, e^{\max\limits_{p\in \underline{l}}(\frac{\ln\mu_p^M}{\tau_{ap}^M}-\alpha_M)(t-t_0)}\, V(\varsigma_{t_0},\sigma(t_0)). \end{aligned} \tag{57}$$

On the other hand, using Rayleigh's inequality [22], one can easily find from (33) that

$$a\|\varsigma(t)\|^2 \leq V(\varsigma_t,\sigma) \leq b\|\varsigma(t)\|^2, \tag{58}$$

where

$$\begin{aligned} a &= \min\{\lambda_{\min}(P_\sigma)\ \ |\sigma\in\underline{l}\}, \\ b &= \max\{\lambda_{\max}(P_\sigma)\ \ |\sigma\in\underline{l}\} + d.\max\{\lambda_{\max}(R_\sigma)\,|\sigma\in\underline{l}\}, \\ &\quad + \frac{d^2}{2}\max\{\lambda_{\max}(S_\sigma)\,|\sigma\in\underline{l}\}. \end{aligned}$$

Notice from (58) that

$$V(\varsigma_t,\sigma) \geq a\|\varsigma(t)\|^2,$$

$$V(\varsigma_{t_0}, \sigma(t_0)) \leq b \|\varsigma(t_0)\|^2. \tag{59}$$

Combining (57) and (59) results in

$$\|\varsigma(t)\|^2 \leq \frac{b}{a} e^{\sum_{p=1}^{l} N_{0p} \ln \mu_p^M} e^{\max\limits_{p \in \underline{l}} (\frac{\ln \mu_p^M}{\tau_{ap}^M} - \alpha_M)(t - t_0)} \|\varsigma(t_0)\|^2, \tag{60}$$

$$\|\varsigma(t)\| \leq \sqrt{\frac{b}{a} e^{\sum_{p=1}^{l} N_{0p} \ln \mu_p^M}} e^{-\frac{1}{2} \max\limits_{p \in \underline{l}} (\alpha_M - \frac{\ln \mu_p^M}{\tau_{ap}^M})(t - t_0)} \|\varsigma(t_0)\|. \tag{61}$$

This means that the switched system (8) is exponentially stable with the estimated state decay ratio given by (61).

Remark 5. For $\mu_i^M = 1$ in $\tau_{ai}^M > \tau_{ai}^{M*} = \frac{\ln \mu_i^M}{\alpha_M}$ we have $\tau_a > \tau_a^* = 0$ which means that the switching signal is arbitrary, and the only possible case for (22) is the equality instead of inequality which imposes a common Lyapunov function for all subsystems.

Now, we will establish the weighted H_∞ performance (α_M, γ_1) for the augmented system without fault and parameter uncertainties. If (23) and (24) are held, from (44), we have

$$\dot{V}(\varsigma_t, \sigma) < -\alpha_M V(\varsigma_t, \sigma) - I_\infty(t). \tag{62}$$

For any $t > 0$ and for any arbitrary piecewise constant switching signal $\sigma(t)$, we let $t_0 = 0 < t_1 < t_2 < \cdots < t_k < \cdots < t_{N_\sigma(0,t)}$ denote the switching points of the $\sigma(t)$ over the interval $[0, t]$, where $N_\sigma(0, t) = \sum_{k=1}^{l} N_k(0, t)$. For any $t \in [t_k, t_{k+1})$, the $\sigma(t_k)$th subsystem is active. Using Lemma 5, by integrating (62) from t_k to t, it follows from (55), (56) and (12) that

$$\begin{aligned} V(\varsigma_t, \sigma) &\leq e^{-\alpha_M(t - t_k)} V(\varsigma_{t_k}, \sigma(t_k)) - \int_{t_k}^{t} e^{-\alpha_M(t - \nu)} \mathrm{I}_\infty(\nu) d\nu \\ &= \prod_{p=1}^{l} \mu_p^{N_{\sigma p}(t_0, t)} e^{-\alpha_M(t - t_0)} V(\varsigma_{t_0}, \sigma(t_0)) \\ &\quad - \int_{t_0}^{t} \prod_{p=1}^{l} \mu_p^{N_{\sigma p}(\nu, t)} e^{-\alpha_M(t - \nu)} I_\infty(\nu) d\nu. \end{aligned} \tag{63}$$

Notice that for the time between two consequence switching instants, we have from (12):

$$\forall\, t_{j-1} < \nu < t_j \Rightarrow N_{\sigma p}(\nu, t) \leq N_{0p} + \frac{T_p(\nu, t)}{\tau_{ap}} = N_{0p} + \frac{T_p(t_{j-1}, t)}{\tau_{ap}} = N_{\sigma p}(t_{j-1}, t). \tag{64}$$

Since $V(\varsigma_t, \sigma)$ is positive, for zero initial condition, (63) results in

$$\int_{t_0}^{t} e^{-\alpha_M(t - \nu) + \sum_{p=1}^{l} N_{\sigma p}(\nu, t) \ln \mu_p^M} r^T(\nu) r(\nu) d\nu$$

$$\leq \gamma_{10}^2 \int_{t_0}^{t} e^{-\alpha_M (t-\nu)+\sum_{p=1}^{l} N_{\sigma p}(\nu,t)\ln \mu_p^M} \omega^T(\nu)\omega(\nu)\,d\nu. \tag{65}$$

Multiplying the both sides of (65) by $e^{-\sum_{p=1}^{l} N_{\sigma p}(0,t)\ln \mu_p^M}$ yields:

$$\int_{t_0}^{t} e^{-\alpha_M (t-\nu)+(\sum_{p=1}^{l} (N_{\sigma p}(\nu,t)-N_{\sigma p}(0,t))\ln \mu_p^M)} r^T(\nu) r(\nu)\,d\nu$$

$$\leq \gamma_{10}^2 \int_{t_0}^{t} e^{-\alpha_M (t-\nu)-\sum_{p=1}^{l} N_{\sigma p}(0,\nu)\ln \mu_p^M} \omega^T(\nu)\omega(\nu)\,d\nu. \tag{66}$$

From (12) and (31), we know that

$$-\sum_{p=1}^{l} N_{\sigma p}(0,\nu)\ln \mu_p^M \geq -\alpha_M \nu - \alpha_M \sum_{p=1}^{l} \tau_{ap}^M N_{0p}. \tag{67}$$

Therefore

$$\int_{t_0}^{t} e^{-\alpha_M t} r^T(\nu) r(\nu)\,d\nu \leq e^{\alpha_M \sum_{p=1}^{l} \tau_{ap}^M N_{0p}} \gamma_{10}^2 \int_{t_0}^{t} e^{-\alpha_M (t-\nu)} \omega^T(\nu)\omega(\nu)\,d\nu. \tag{68}$$

And we get

$$\int_{t_0}^{t} e^{-\alpha_M t} r^T(\nu) r(\nu)\,d\nu \leq \gamma_1^2 \int_{t_0}^{t} e^{-\alpha_M (t-\nu)} \omega^T(\nu)\omega(\nu)\,d\nu. \tag{69}$$

Integrating the both sides of (69) from t_0 to ∞ will result in

$$\int_{t_0}^{\infty} e^{-\alpha_M \nu} r^T(\nu) r(\nu)\,d\nu \leq \gamma_1^2 \int_{t_0}^{\infty} \omega^T(\nu)\omega(\nu)\,d\nu. \tag{70}$$

This means that the switched system (8) satisfies the weighted H_∞ performance (α_M, γ_1) with $\gamma_1 = \gamma_{10} \exp\left(0.5\,\alpha_M \sum_{p=1}^{l} \tau_{ap}^M N_{0p}\right)$ in (9). This completes the proof. □

3.2 The weighted H_- performance problem

In the following theorem, given the IFDRCU gain matrices and based on the weighted H_- performance index (α_m, γ_2) defined in (10), sufficient conditions in the form of matrix inequalities are derived for the exponential stability of the augmented system in the presence of parameter uncertainties and input disturbances. In addition, the minimum allowable average time per each subsystem activity is calculated to satisfy the weighted H_- performance index (α_m, γ_2).

Theorem 2. For given scalars $\alpha_m > 0, \mu_i^m \geq 1$ and appropriately-dimensioned real matrices $\overset{\frown}{A}_{mi}, \overset{\frown}{B}_{mi}, C_{mi}, D_{mi}, \overset{\frown}{K}_{mi}, \overset{\frown}{L}_{mi}$, if there exist positive definite matrices $P_i \succ 0$, $R_i \succ 0, S_i \succ 0$, appropriately-dimensioned real matrices $G_i, H_i = H_i^T, Y_{ki}$ $(i = 1, 2, 3)$, and constant scalars $\gamma_{20} > 0, \delta_{2i} > 0$ such that the following inequalities hold, then we have

$$P_i \leq \mu_i^m P_j, R_i \leq \mu_i^m R_j, S_i \leq \mu_i^m S_j \qquad i, j \in \underline{l}, \tag{71}$$

$$\Omega_{mi} = \begin{bmatrix} \Phi_{mi} & \Lambda_{mi} \\ * & -\delta_{2i} I \end{bmatrix} \prec 0, \tag{72}$$

$$\Sigma_{mi} = \begin{bmatrix} H_i & G_i \\ * & e^{-\alpha_m d_i} S_i \end{bmatrix} \succ 0, \tag{73}$$

where

$$\Phi_{mi} = \begin{bmatrix} \Phi_{mi11} & \Phi_{mi12} & \Phi_{mi13} & \Phi_{mi14} & \Phi_{mi15} \\ * & \Phi_{mi22} & 0 & 0 & \Phi_{mi25} \\ * & 0 & \Phi_{mi33} & \Phi_{mi34} & \Phi_{mi35} \\ * & 0 & * & -3I & 0 \\ * & * & * & 0 & S_i - 2P_i \end{bmatrix}, \tag{74}$$

$$\Lambda_{mi} = \begin{bmatrix} P_{i1} M_{i1} + (T_i^T \begin{bmatrix} \overset{\frown}{L}_{mi} \\ 0 \end{bmatrix} - Y_{1i}^T D_{mi}) M_{i2} & P_{i1} M_{i1} & 0 \\ (\overset{\frown}{B}_{mi} - Y_{2i}^T D_{mi}) M_{i2} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ -Y_{3i}^T D_{mi} M_{i2} & 0 & 0 \\ D_{mi} M_{i2} & 0 & 0 \\ \sqrt{d_i} P_{i1} M_{i1} & \sqrt{d_i} P_{i1} M_{i1} & \sqrt{d_i} T_i^T \begin{bmatrix} \overset{\frown}{L}_{mi} \\ 0 \end{bmatrix} M_{i2} \\ 0 & 0 & \sqrt{d_i}\, \overset{\frown}{B}_{mi}\, M_{i2} \end{bmatrix}, \tag{75}$$

$$\Phi_{mi11} = He\left(\begin{bmatrix} P_{i1} A_i + (T_i^T \begin{bmatrix} \overset{\frown}{L}_{mi} \\ 0 \end{bmatrix} - Y_{1i}^T D_{mi}) C_i + \delta_{2i} N_{i1}^T N_{i1} & T_i^T \begin{bmatrix} \overset{\frown}{K}_{mi} \\ 0 \end{bmatrix} - Y_{1i}^T C_{mi} \\ (\overset{\frown}{B}_{mi} - Y_{2i}^T D_{mi}) C_i & \overset{\frown}{A}_{mi} - Y_{2i}^T C_{mi} \end{bmatrix}\right),$$

$$+ \alpha_m P_i + R_i + d_i H_{i1} + G_{i1} + G_{i1}^T,$$

$$\Phi_{mi12} = \begin{bmatrix} P_{i1} A_{di} & 0 \\ 0 & 0 \end{bmatrix} + d_i H_{i2} - G_{i1} + G_{i2}^T,$$

$$\begin{aligned}
\Phi_{mi13} &= \begin{bmatrix} P_{i1}B_{fi} + (T_i^T \begin{bmatrix} \overset{\frown}{L}_{mi} \\ 0 \end{bmatrix} - Y_{1i}^T D_{mi})D_{fi} - C_i^T D_{mi}^T Y_{3i} + 2\delta_{2i} N_{i1}^T N_{i4} \\ (\overset{\frown}{B}_{mi} - Y_{2i}^T D_{mi})D_{fi} - C_{mi}^T Y_{3i} \end{bmatrix}, \\
\Phi_{mi14} &= \begin{bmatrix} Y_{1i}^T + C_i^T D_{mi}^T \\ Y_{2i}^T + C_{mi}^T \end{bmatrix}, \\
\Phi_{mi15} &= \sqrt{d_i} \begin{bmatrix} A_i^T P_{i1} + C_i^T \begin{bmatrix} \overset{\frown}{L}_{mi}^T & 0 \end{bmatrix} T_i & C_i^T \overset{\frown}{B}_{mi}^T \\ \begin{bmatrix} \overset{\frown}{K}_{mi}^T & 0 \end{bmatrix} T_i & \overset{\frown}{A}_{mi}^T \end{bmatrix}, \\
\Phi_{mi22} &= -(1-\rho_i)e^{-\alpha_m d_i} R_i + d_i H_{i3} - G_{i2} - G_{i2}^T + \delta_{2i} \begin{bmatrix} N_{i2}^T N_{i2} & 0 \\ 0 & 0 \end{bmatrix}, \\
\Phi_{mi25} &= \sqrt{d_i} \begin{bmatrix} A_{di}^T P_{i1} & 0 \\ 0 & 0 \end{bmatrix}, \\
\Phi_{mi33} &= \gamma_{20}^2 I - He(Y_{3i}^T D_{mi} D_{fi}) + 2\delta_{2i} N_{i4}^T N_{i4}, \\
\Phi_{mi34} &= D_{fi}^T D_{mi}^T + Y_{3i}^T, \\
\Phi_{mi35} &= \sqrt{d_i} \begin{bmatrix} B_{fi}^T P_{i1} + D_{fi}^T \begin{bmatrix} \overset{\frown}{L}_{mi}^T & 0 \end{bmatrix} T_i & D_{fi}^T \overset{\frown}{B}_{mi}^T \end{bmatrix},
\end{aligned} \tag{76}$$

where M_{ij}, N_{ij} are defined in (5), T_i is defined in (3), and

$$G_i \triangleq \begin{bmatrix} G_{i1}^T & G_{i2}^T \end{bmatrix}^T, \tag{77}$$

$$H_i \triangleq \begin{bmatrix} H_{i1} & H_{i2} \\ * & H_{i3} \end{bmatrix}, \tag{78}$$

$$P_i = \begin{bmatrix} P_{i1} & 0 \\ 0 & P_{i2} \end{bmatrix}, \qquad P_{i1} = T_i^T \begin{bmatrix} \overset{\frown}{P}_{i1} & 0 \\ 0 & \overset{\frown}{P}_{i2} \end{bmatrix} T_i, \tag{79}$$

$$\overset{\frown}{A}_{mi} = P_{i2} A_{mi}, \qquad \overset{\frown}{B}_{mi} = P_{i2} B_{mi}, \qquad \overset{\frown}{K}_{mi} = \overset{\frown}{P}_{i1} K_{mi}, \qquad \overset{\frown}{L}_{mi} = \overset{\frown}{P}_{i1} L_{mi}. \tag{80}$$

Then the augmented system (8) is exponentially stable and satisfies the weighted H_- performance index (α_m, γ_2) in (10) for any switching signal with MDADT met by (81)

$$\tau_{ai}^m > \tau_{ai}^{m*} = \frac{\ln \mu_i^m}{\alpha_m}. \tag{81}$$

Proof. By defining $I_-(t) \triangleq \gamma_{20}^2 f^T(t) f(t) - r^T(t) r(t)$, this theorem can be proved by employing similar techniques as in the proof of Theorem 1.

For $\omega(t) = 0$, the inequality $\dot{V}(\varsigma_t, \sigma) + \alpha_m V(\varsigma_t, \sigma) + I_-(t) \leq 0$ holds if both (73) and the inequality (82) hold.

$$\begin{bmatrix} \Psi_{\sigma 11}(t) & \Psi_{\sigma 12}(t) & \Psi_{\sigma 13}(t) & \Psi_{\sigma 14}(t) \\ * & \Psi_{\sigma 22}(t) & 0 & \Psi_{\sigma 24}(t) \\ * & 0 & \Psi_{\sigma 33}(t) & \Psi_{\sigma 34}(t) \\ * & * & * & -P_\sigma S_\sigma^{-1} P_\sigma \end{bmatrix} - \begin{bmatrix} \bar{C}_\sigma^T(t)\bar{C}_\sigma(t) & 0 & \bar{C}_\sigma^T(t)\bar{D}_{f\sigma}(t) & 0 \\ 0 & 0 & 0 & 0 \\ * & 0 & \bar{D}_{f\sigma}^T(t)\bar{D}_{f\sigma}(t) & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \prec 0. \quad (82)$$

Applying Lemmas 2 and 28, (82) can be rewritten as

$$\begin{bmatrix} \Psi_{\sigma 11}(t) & \Psi_{\sigma 12}(t) & \Psi_{\sigma 13}(t) & \Psi_{\sigma 14}(t) \\ * & \Psi_{\sigma 22}(t) & 0 & \Psi_{\sigma 24}(t) \\ * & 0 & \Psi_{\sigma 33}(t) & \Psi_{\sigma 34}(t) \\ * & * & * & S_\sigma - 2P_\sigma \end{bmatrix} - \begin{bmatrix} \bar{C}_\sigma^T(t) \\ 0 \\ \bar{D}_{f\sigma}^T(t) \\ 0 \end{bmatrix} \begin{bmatrix} \bar{C}_\sigma(t) & 0 & \bar{D}_{f\sigma}(t) & 0 \end{bmatrix} \prec 0. \qquad (83)$$

This is apparently equal to

$$E_\sigma^{\perp T} \Delta_\sigma(t) E_\sigma^{\perp} \prec 0, \qquad (84)$$

with

$$E_\sigma^{\perp} = \begin{bmatrix} I & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & I & 0 \\ \bar{C}_\sigma(t) & 0 & \bar{D}_{f\sigma}(t) & 0 \\ 0 & 0 & 0 & I \end{bmatrix},$$

$$\Delta_\sigma(t) \triangleq \begin{bmatrix} \Psi_{\sigma 11}(t) & \Psi_{\sigma 12}(t) & \Psi_{\sigma 13}(t) & 0 & \Psi_{\sigma 14}(t) \\ * & \Psi_{\sigma 22}(t) & 0 & 0 & \Psi_{\sigma 24}(t) \\ * & 0 & \Psi_{\sigma 33}(t) & 0 & \Psi_{\sigma 34}(t) \\ 0 & 0 & 0 & -I & 0 \\ * & * & * & 0 & S_\sigma - 2P_\sigma \end{bmatrix}. \qquad (85)$$

Then, if we choose $E_\sigma = \begin{bmatrix} -\bar{C}_\sigma(t) & 0 & -\bar{D}_{f\sigma}(t) & I & 0 \end{bmatrix}$ as one annihilator for $E_\sigma^{\perp}$ and an arbitrary matrix $Y_\sigma = \begin{bmatrix} \begin{bmatrix} Y_{1\sigma} & Y_{2\sigma} \end{bmatrix} & 0 & Y_{3\sigma} & -I & 0 \end{bmatrix}^T$, using the similar technique as in [40, 32], and by Finsler's Lemma 6, we deduce that (84) holds if

$$T_\sigma(t) = \Delta_\sigma(t) + He(Y_\sigma E_\sigma) \prec 0, \qquad (86)$$

in which $Y_{i\sigma}$ $(i = 1, 2, 3)$ are some arbitrary tuning matrices.

Remark 6. Since a particular structure is chosen for the tuning matrix in Finsler's lemma in (86), it becomes a sufficient condition for satisfying (84). A more general form for the tuning matrix can also be chosen, but the complexity of the resulting LMIs will be increased.

Therefore, it is evident that $\dot{V}(\varsigma_t,\sigma)+\alpha_m V(\varsigma_t,\sigma)+I_{-}(t)\leq 0$ holds, if both (73) and the following inequality hold.

$$T_\sigma(t)=\begin{bmatrix} \Psi_{\sigma 11}(t)-He(\begin{bmatrix}Y_{1\sigma} & Y_{2\sigma}\end{bmatrix}^T\bar{C}_\sigma(t)) & \Psi_{\sigma 12}(t) & \Psi_{\sigma 13}(t)-\begin{bmatrix}Y_{1\sigma} & Y_{2\sigma}\end{bmatrix}^T\bar{D}_{f\sigma}(t)-\bar{C}_\sigma^T(t)Y_{3\sigma} & \begin{bmatrix}Y_{1\sigma} & Y_{2\sigma}\end{bmatrix}^T+\bar{C}_\sigma^T(t) & \Psi_{\sigma 14}(t) \\ * & \Psi_{\sigma 22}(t) & 0 & 0 & \Psi_{\sigma 24}(t) \\ * & 0 & \Psi_{\sigma 33}(t)-He(Y_{3\sigma}^T\bar{D}_{f\sigma}(t)) & \bar{D}_{f\sigma}^T(t)+Y_{3\sigma}^T & \Psi_{\sigma 34}(t) \\ * & 0 & * & -3I & 0 \\ * & * & * & 0 & -S_\sigma \end{bmatrix}<0. \tag{87}$$

Referring to Assumption 2, and pursuing the same line as in Theorem 1, we have

$$\Delta T_\sigma(t)=He(\Lambda_{m\sigma}(Q(t),Q(t),Q(t))\Gamma_{m\sigma}), \tag{88}$$

where $\Lambda_{m\sigma}$ is defined in (75) and

$$\Gamma_{m\sigma}\triangleq\begin{bmatrix} N_{\sigma 1} & 0 & 0 & 0 & N_{\sigma 4} & 0 & 0 & 0 \\ 0 & 0 & N_{\sigma 2} & 0 & 0 & 0 & 0 & 0 \\ N_{\sigma 1} & 0 & 0 & 0 & N_{\sigma 4} & 0 & 0 & 0 \end{bmatrix}, \tag{89}$$

and by using the generalized square inequality Lemma 4, i.e., (17), we obtain

$$\Delta T_\sigma(t)\leq\delta_{2\sigma}^{-1}\Lambda_{m\sigma}\Lambda_{m\sigma}^T+\delta_{2\sigma}\Gamma_{m\sigma}^T\Gamma_{m\sigma}. \tag{90}$$

And we can get

$$\Phi_{m\sigma}+\delta_{2\sigma}^{-1}\Lambda_{m\sigma}\Lambda_{m\sigma}^T\leq 0, \tag{91}$$

where $\Phi_{m\sigma}$ is defined in (74). Finally, using the Schur complement lemma (14), inequality (91) turns to (72).

The rest of the proof is omitted because it is similar to that of Theorem 1. This means that the switched system (8) satisfies the weighted H_{-} performance (α_m,γ_2) with

$$\gamma_2=\gamma_{20}\exp\left(-0.5\,\alpha_m\sum_{p=1}^{l}\tau_{ap}^m N_{0p}\right)$$

in (10). This completes the proof. □

3.3 The mixed weighted H_∞/H_{-} problem

In this section, the combination of both the problems of disturbance attenuation and fault sensitivity amplification is described by the following corollary as a mixed weighted H_∞/H_{-} problem. To solve this problem, an algorithm is also presented.

Corollary 1. By combining the results of Theorems 1 and 2 referring to the optimization problem defined in (11), the proposed IFDRC scheme can be summarized as follows:

Under the switching law $\sigma(t)$ with the defined MDADT in (92), if conditions (22)-(24) and (71)-(73) are satisfied, then the augmented system (8) is exponentially stable with the estimated state decay ratio in (61), and also satisfies mixed weighted H_∞/H_- performance indices (9) and (10).

$$\tau_{ai} \geq \max(\tau_{ai}^{m*}, \tau_{ai}^{M*}). \tag{92}$$

Moreover, the IFDRCU matrices can be constructed by (32).

Since (22)-(24) are in the LMI form, and (71)-(73) are BMI, the IFDRCU design problem yields the following two-step optimization algorithm [17].

Algorithm 1

0. Select the scalars $\alpha_M > 0$, $\mu_i^M \geq 1$, $\alpha_m > 0$, $\mu_i^m \geq 1$.
1. Solve (22)-(24) to obtain the minimum permitted level of disturbance attenuation, γ_1, which will lead to the appropriate robust controller to satisfy the H_∞ performance index (9).
2. Substitute the resulted controller gains from the first step into (71)-(73) and check the feasibility of these inequalities to find the maximum permitted level of fault sensitivity, γ_2, that satisfies the H_- performance index (10).

Also, compromising between the desired γ_1, γ_2 can be done by repeating the two aforementioned steps.

3.4 Residual signal evaluation

For successful fault detection and generating fault occurrence alarm, the last step after designing the residual generator is to evaluate the residual signal (Figure 1). This step includes two tasks:

- Producing an evaluation function ($J_{RMS}(L)$)

- Specifying a threshold (J_{th}).

By employing a similar method to the other fault detection literature [6, 14], which relaxes the necessity to estimate the fault signal, the following residual evaluation function is used:

$$J_{RMS}(L) = \|r(t)\|_2 = \left(\frac{1}{L}\int_{t_0}^{t_0+L} r^T(\tau)r(\tau)d\tau\right)^{\frac{1}{2}}, \tag{93}$$

where L is the evaluation time step and t_0 is the initial evaluation time instant.

To identify when a fault has occurred, this evaluation function can be compared to the threshold by the following rule:

$$J_{RMS}(L) - J_{th} = \begin{cases} > 0, & \text{fault occured} \;\Rightarrow\; \text{Alarm}, \\ < 0, & \text{No fault}. \end{cases} \tag{94}$$

As indicated in [9], the threshold can be chosen as

$$J_{th} = \sup_{\|\omega(t)\|_2 \leq \delta_\omega, f=0} J_{RMS}(L). \tag{95}$$

4 A Numerical Example

In this section, a numerical example is considered as a case study for simulating the proposed framework for the IFDRCU design technique to illustrate the effectiveness and applicability of the theoretical results.

The realization of this numerical example can be given by the Electrical Circuit system, which is shown in Figure 2.

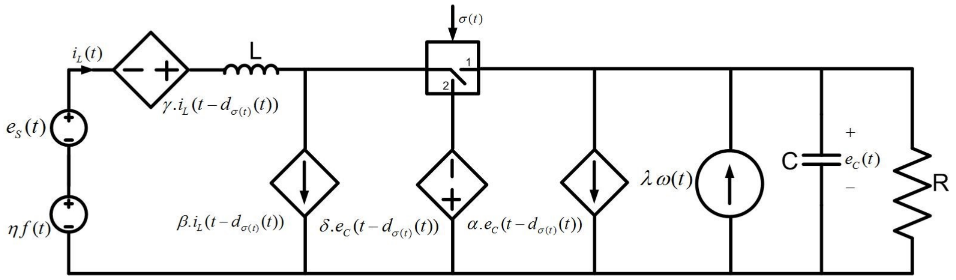

$\sigma(t)$
$i_L(t)$
L
1
2
$\gamma.i_L(t-d_{\sigma(t)}(t))$
$e_S(t)$
$\eta f(t)$
$\beta.i_L(t-d_{\sigma(t)}(t))$
$\delta.e_C(t-d_{\sigma(t)}(t))$
$\alpha.e_C(t-d_{\sigma(t)}(t))$
$\lambda\,\omega(t)$
C
+
$e_C(t)$
−
R

Figure 2: A sample Electrical Circuit switched system.

According to Kirchhoff's Circuit Law, for two switching modes of this Electrical Circuit, we have

$$\begin{aligned} KCL: C\frac{de_C}{dt} + \frac{e_C(t)}{R} &+ i_L(t)\cdot(\sigma(t)-2) + \alpha\cdot e_C(t-d_{\sigma(t)}(t)) \\ &- \beta\cdot i_L(t-d_{\sigma(t)}(t)).(\sigma(t)-2) - \lambda\cdot\omega(t) = 0, \\ KVL: L\frac{di_L}{dt} &- e_C(t)\cdot(\sigma(t)-2) - \delta\cdot e_C(t-d_{\sigma(t)}(t))\cdot(\sigma(t)-1) \\ &- \gamma\cdot i_L(t-d_{\sigma(t)}(t)) - e_s(t) - \eta\cdot f(t) = 0. \end{aligned} \tag{96}$$

The state-space representations of this circuit are given by

$$\dot{x}(t) = \begin{bmatrix} -\frac{1}{RC} & \frac{(2-\sigma(t))}{C} \\ \frac{(\sigma(t)-2)}{L} & 0 \end{bmatrix} x(t) + \begin{bmatrix} -\frac{\alpha}{C} & \frac{\beta.(\sigma(t)-2)}{C} \\ \frac{\delta.(\sigma(t)-1)}{L} & \frac{\gamma}{L} \end{bmatrix} x(t-d_{\sigma(t)}(t))$$

$$+\begin{bmatrix}0\\ \frac{1}{L}\end{bmatrix}u(t)+\begin{bmatrix}\frac{\lambda}{C}\\ 0\end{bmatrix}\omega(t)+\begin{bmatrix}0\\ \frac{\eta}{L}\end{bmatrix}f(t), \tag{97}$$

where $\begin{bmatrix}x_1(t) & x_2(t)\end{bmatrix}^T = \begin{bmatrix}e_C(t) & i_L(t)\end{bmatrix}^T$ and $u(t) = e_S(t)$ are the state vector and input signal, respectively.

When the parameters $\alpha, \beta, \delta, \gamma, \eta, \lambda$ are set to zero in this electrical circuit, this is equivalent to the Boost Converter switched system. As a typical circuit system, the Boost Converter is used to transform the source voltage into a higher voltage. This class of power converters has been modeled as switched systems. In recent years, the fault detection and control problems for such power converters have been widely studied in the literature [10, 26]. More details of this system are given in [36].

For $\alpha = -0.2, \beta = 0.3, \delta = 0.4, \gamma = -0.5, \lambda = -0.1, \eta = 0.4$ and $R = 1\,\Omega, L = 1\,H, C = 1\,F$ the following state-space matrices are obtained.

$$A_1 = \begin{bmatrix}-1 & 1\\ -1 & 0\end{bmatrix},\ A_{d1} = \begin{bmatrix}0.2 & -0.3\\ 0 & -0.5\end{bmatrix},\ B_1 = \begin{bmatrix}0\\ 1\end{bmatrix},\ B_{\omega 1} = \begin{bmatrix}0.1\\ 0\end{bmatrix},$$

$$B_{f1} = \begin{bmatrix}0\\ 0.4\end{bmatrix},\ C_1 = \begin{bmatrix}0.1 & 0.1\end{bmatrix},\ D_{\omega 1} = [0],\ D_{f1} = [0.1],$$

$$A_2 = \begin{bmatrix}-1 & 0\\ 0 & 0\end{bmatrix},\ A_{d2} = \begin{bmatrix}0.2 & 0\\ 0.4 & -0.5\end{bmatrix},\ B_2 = \begin{bmatrix}0\\ 1\end{bmatrix},\ B_{\omega 2} = \begin{bmatrix}0.1\\ 0\end{bmatrix},$$

$$B_{f2} = \begin{bmatrix}0\\ 0.4\end{bmatrix},\ C_2 = \begin{bmatrix}0.3 & 0.4\end{bmatrix},\ D_{\omega 2} = [0],\ D_{f2} = [0.1], \tag{98}$$

which are similar to the Boost Converter matrices in [10], except that it does not have state delay. Also, output matrices are considered the same as in [10].

For parameter uncertainties, the following real constant matrices and $Q(t) = \sin(3t)$ are considered:

$$M_1 = \begin{bmatrix}0.1\\ -0.2\end{bmatrix},\ M_2 = [0.1], \tag{99}$$

$$N_1 = \begin{bmatrix}-0.2 & 0.1\end{bmatrix},\ N_2 = \begin{bmatrix}0.1 & -0.1\end{bmatrix},\ N_3 = [0.2],\ N_4 = [-0.1].$$

Time-varying state delays for two subsystems are supposed to be $d_1(t) = 0.2 + 0.1\cos(t)$ and $d_2(t) = 0.3 - 0.2\sin(t)$. Therefore, the upper bound of delay and its derivative for two modes will be $d_1 = 0.3$, $\rho_1 = 0.1$ and $d_2 = 0.5$, $\rho_2 = 0.2$, respectively.

Given $\alpha_M = 0.1, \alpha_m = 0.3, \mu_{M1} = 1.01, \mu_{m1} = 1.1, \mu_{M2} = 1.02, \mu_{m2} = 1.5$, the allowed minimum MDADT for each subsystem could be obtained from (31) and (81), and (92) as $\tau^*_{a1} = \max(0.3177, 0.0995) = 0.3177$, $\tau^*_{a2} = \max(1.3516, 0.1980) = 1.3516$. By MDADT constraints $\tau_{a1} = 0.53, \tau_{a2} = 1.39$, the switching signal in Figure 3.a is chosen.

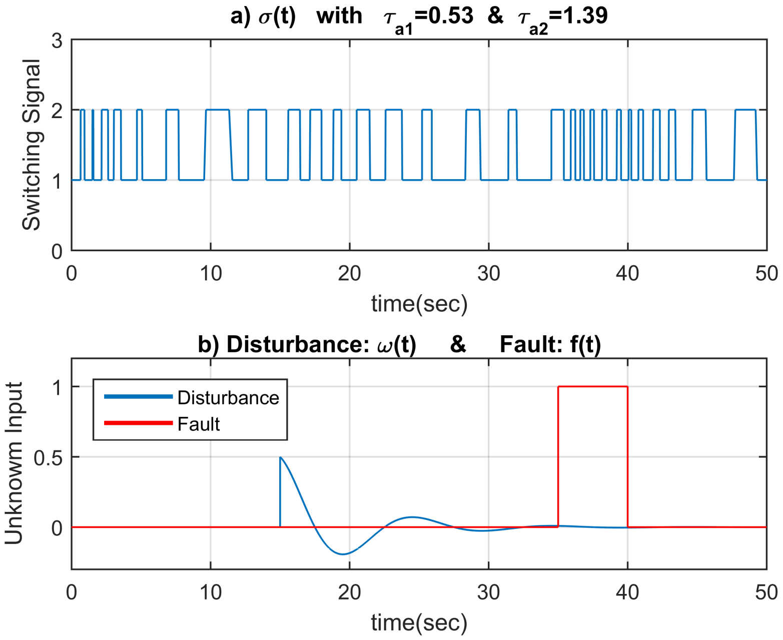


Figure 3: (a). Switching signal (b). Fault and disturbance signals.

Solving the LMIs in (22)-(24) by MOSEK solver [1] in MATLAB/YALMIP, results in the following controller/detector gains, and minimum disturbance attenuation level,$\gamma_1 = 0.187$.

$$\begin{bmatrix} A_{m1} & B_{m1} \\ C_{m1} & D_{m1} \\ K_{m1} & L_{m1} \end{bmatrix} = \begin{bmatrix} -0.7175 & 0.3226 & -0.0786 \\ 0.3226 & -0.7175 & -0.0786 \\ -0.0113 & -0.0113 & -0.4972 \\ -0.1252 & -0.1252 & -1.6310 \end{bmatrix},$$

$$\begin{bmatrix} A_{m2} & B_{m2} \\ C_{m2} & D_{m2} \\ K_{m2} & L_{m2} \end{bmatrix} = \begin{bmatrix} -0.7378 & 0.3134 & -0.0284 \\ 0.3134 & -0.7378 & -0.0284 \\ -0.0208 & -0.0208 & -0.2519 \\ 0.1272 & 0.1272 & -1.6611 \end{bmatrix}, \tag{100}$$

Then, solving the LMIs in (71)-(73) results in the fault sensitivity level, $\gamma_2 = 0.016$.

For simulation, we assume that the unknown bounded input, called disturbance, is given by $\omega(t) = 0.5\exp(-2(t-15))\cos(0.2\pi(t-15))u(t-15)$ with $\delta_\omega = 0.5$, and the fault occurs as a step in $t = 35s$ and remains for 5 seconds, while disturbance is present from $t = 15\ s$ as shown in Figure 3.b.

Choosing the initial state $x_0 = \begin{bmatrix} 0.6 & -0.4 \end{bmatrix}^T$, Figure 4.a and Figure 4.b show trajectories of the state responses of the system ($x(t)$) and its control input ($u(t)$), respectively, from which we can see that the closed-loop system is exponentially stable under the initial state and unknown disturbances.

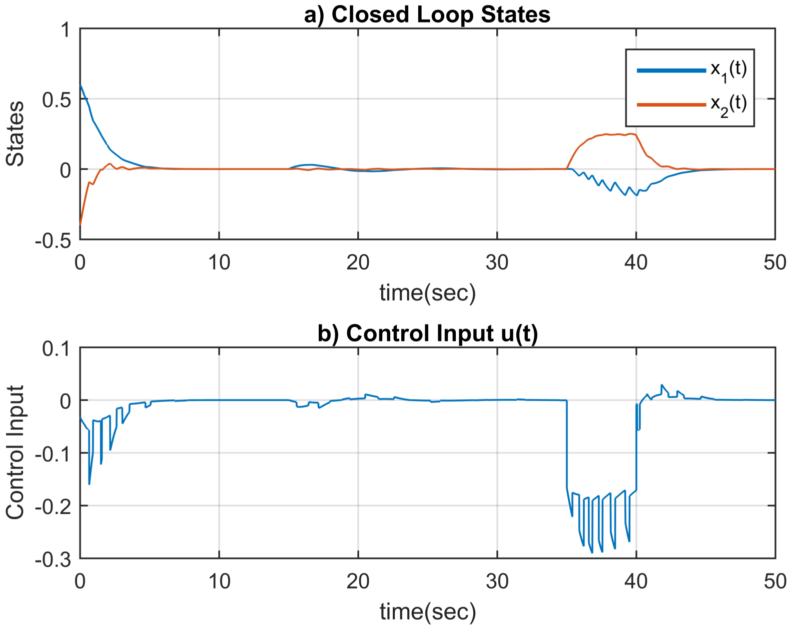


Figure 4: (a). State responses of the closed-loop system. (b). Control input.

The generated residual signal and the evolution of the residual evaluation function are shown in Figure 5.a and Figure 5.b.

Simulating the system in a fault-free case, the threshold can be determined as $J_{th} = 0.004$. It can be seen from Figure 5.b that fault is detected at $t = 35.2\ s$

Simulation results show that the early detection of fault can be achieved by the controller/detector immediately and effectively when faults occur, although disturbance input, mode-dependent time-varying state delay, and parameter uncertainties are present and the control loop is closed. The benefit of integrated fault detection and control design of the system is that fault occurrence cannot be hidden by the control action.

To illustrate the excellence of the proposed technique, it is compared with the existing method [10] in two cases; with and without state delay and parameter uncertainty. Comparing the disturbance attenuation level values (γ_1), as shown in Table 1, show that the proposed approach is less conservative. It has a better disturbance rejection capacity because the residual signal is less affected by the unknown input.

By comparing the minimum allowed average dwell time values in Table 1, the proposed approach has more flexibility in the switching times, since it admits different average dwell times for each subsystem. Note that since the compared paper did not

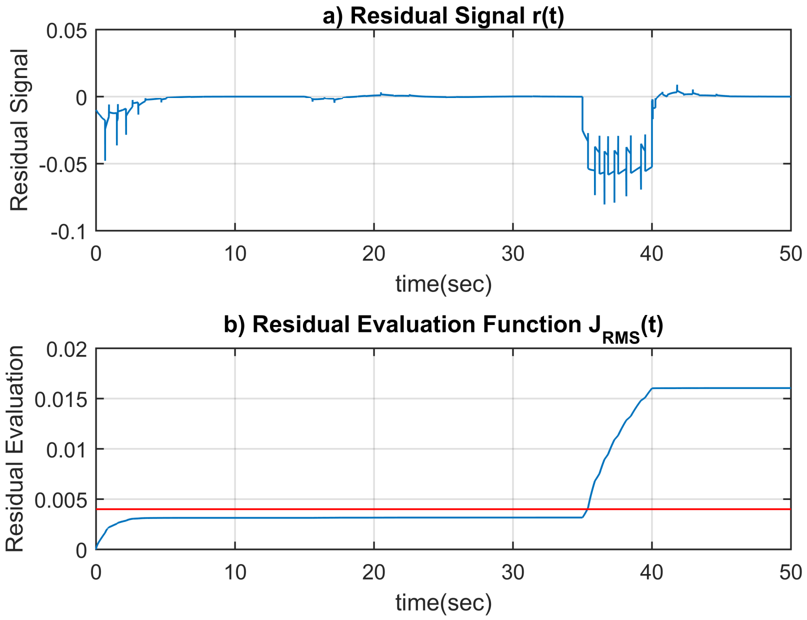


Figure 5: (a). Generated residual signal. (b). Residual evaluation function.

Table 1: Comparison with the existing results

Method	Delay	Uncertainty	Disturbance attenuation level (γ_1)	Average Dwell time#1	Average Dwell time#2
[10]	No	No	0.91	12.307	12.307
This paper	Yes	Yes	0.1871	0.3176	1.3516
This paper	No	No	0.1644	0.3176	1.3516

consider state-space delay and parameter uncertainties, our results were reported with and without state delay and parameter uncertainties.

5 Conclusion

The proposed MDADT switching strategy was less conservative and allowed lower and as well different ADTs for each subsystem compared with the general ADT switching method. The main objective of this paper was to propose a general framework for IFDRC of linear continuous-time switching systems suffering from mode-dependent time-varying state delay, parameter uncertainties, and input disturbance. Sufficient conditions for IFDRC design were derived based on the MDADT technique. Multiple Lyapunov-Krasovskii functions under the framework of mixed H_∞/H_-, and the fault detection filters and controllers were developed together. Finally, the proposed scheme was applied to a switched model of an Electrical Circuit system, and the simulation results indicated the effectiveness of the proposed technique.

References

[1] Andersen, E.D., Andersen, K.D. (2000). “The Mosek interior-point optimizer for linear programming: An implementation of the homogeneous algorithm”, in High Performance Optimization, 33, H. Frenk, K. Roos, T. Terlaky, S. Zhang, Eds. Boston, MA: Springer US, 197-232.

[2] Benzaouia, A., Eddoukali, Y. (2018). “Robust fault detection and control for continuous-time switched systems with average dwell time”, Circuits Syst Signal Process, 37(6), 2357-2373.

[3] Bhattacharya, R. “An introduction to linear matrix inequalities (LMI)”, Aerospace Engineering, Texas A& M University.

[4] Blondel V., Megretski, A., Eds. (2004). “Unsolved problems in mathematical systems and control theory”, Princeton, NJ: Princeton Univ. Press.

[5] Davoodi, M.R., Golabi, A., Talebi, H.A., Momeni H. R. (2012). “Simultaneous fault detection and control design for switched linear systems: A linear matrix inequality approach”, Journal of Dynamic Systems, Measurement, and Control, 134(6), 061010.

[6] Davoodi, M.R., Golabi, A., Talebi, H.A., Momeni H.R. (2013). “Simultaneous fault detection and control design for switched linear systems based on dynamic observer”, Optimal Control Applications and Methods, 34(1), 35-52.

[7] Davoodi, M., Meskin, N., Khorasani, K. (2018). “Integrated fault diagnosis and control design of linear complex systems”.

[8] Davoodi, M.R., Talebi, H.A., Momeni, H.R. (2011). “A novel simultaneous fault detection and control approach based on dynamic observer”, IFAC Proceedings Volumes, 44(1), 12036-12041.

[9] Du, D., Jiang, B., Shi, P., Karimi, H.R. (2014). “Fault detection for continuous-time switched systems under asynchronous switching”, International Journal of Robust Nonlinear Control, 24(11), 1694-1706.

[10] Eddoukali, Y., Benzaouia, A., Ouladsine, M. (2019). “Integrated fault detection and control design for continuous-time switched systems under asynchronous switching”, ISA Transactions, 84, 12-19.

[11] Elahi, A., Alfi ,A. (2017). “Finite-time H_∞ control of uncertain networked control systems with randomly varying communication delays”, ISA Transactions, 69, 65-88.

[12] Ghalehnoie, M. (2020). “Ultimate boundedness control for uncertain nonlinear impulsive switched systems: A fuzzy approach based on a complete Takagi–Sugeno structure”, Journal of Control, Automation and Electrical Systems, 31(2), 271-282.

[13] Hajshirmohamadi, S., Davoodi, M. (2015). “Comments on: Fault detection for continuous-time switched systems under asynchronous switching”, International Journal of Robust Nonlinear Control, 25(15), 2865-2868.

[14] Iftikhar, K., Khan, A.Q., Abid, M. (2015). “Optimal fault detection filter design for switched linear systems”, Nonlinear Analysis: Hybrid Systems, 15, 132-144.

[15] Ishihara, J.Y., Kussaba, H.T.M., Borges, R.A. (2017). “Existence of continuous or constant Finsler’s variables for parameter-dependent systems”, IEEE Transactions on Automatic Control, 62(8), 4187-4193.

[16] Li, Z., Mazars, E., Zhang, Z., Jaimoukha, I.M. (2012). “State-space solution to the H$-/H_\infty$ fault-detection problem”, International Journal of Robust Nonlinear Control, 22(3), 282-299.

[17] Li, J., Yang, G.H. (2013). “Simultaneous fault detection and control for switched systems under asynchronous switching”, Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering, 227(1), 70-84.

[18] Liberzon, D. (2003). “Switching in systems and control”, Boston, Birkhäuser.

[19] Liu, X. (2010). “Stabilization of switched linear systems with mode-dependent time-varying delays”, Applied Mathematics and Computation, 216(9), 2581-2586.

[20] Liu, X., Zhai, D., He, D.K., Chang, X.H. (2018). “Simultaneous fault detection and control for continuous-time Markovian jump systems with partially unknown transition probabilities”, Applied Mathematics and Computation, 337, 469-486.

[21] Ning, Z., Yu, J., Wang, T. (2017). “Simultaneous fault detection and control for uncertain discrete-time stochastic systems with limited communication”, Journal of the Franklin Institute, 354(17), 7794-7811.

[22] Ramezani-al, M. R., Kamyad, A. V., Pariz, N. (2016). “A new switching strategy design for uncertain switched linear systems based on min-projection strategy in guaranteed cost control problem”, IMA Journal of Mathematical Control and Information, 33(4), 1033-1049.

[23] Raza, M.T., Khan, A.Q., Mustafa, G., Abid, M. (2016). “Design of fault detection and isolation filter for switched control systems under Asynchronous Switching”, IEEE Transactions on Control Systems Technology, 24(1), 13-23.

[24] Shokouhi-Nejad, H., Ghiasi, A.R., Badamchizadeh, M.A. (2017). “Robust simultaneous finite-time control and fault detection for uncertain linear switched systems with time-varying delay”, IET Control Theory & Applications, 11(7), 1041-1052.

[25] Shokouhi-Nejad, H., Ghiasi, A.R., Badamchizadeh, M.A., Pezeshki, S. (2019). "H_∞/H-Simultaneous fault detection and control for continuous-time linear switched delay systems under asynchronous switching", Transactions of the Institute of Measurement and Control, 41(1), 263-275.

[26] Shokouhi-Nejad, H., Rikhtehgar Ghiasi, A., Badamchizadeh, M.A. (2017). "Robust simultaneous fault detection and control for a class of nonlinear stochastic switched delay systems under asynchronous switching", Journal of the Franklin Institute, 354(12), 4801-4825.

[27] Soltani, H., Naoui, S.B.H.A., Aitouche, A., Harabi, R.E., Abdelkrim, M.N. (2015). "Robust simultaneous fault detection and control approach for time-delay systems", IFAC-PapersOnLine, 48(21), 1244-1249.

[28] Su, Q., Fan, Z., Li, J. (2019). "H_∞/H-fault detection for switched systems with all subsystems unstable", IET Control Theory & Applications, 13(12), 1796-1803.

[29] Su, Q., Fan, Z., Li, J. (2019). "Observer-based fault detection for switched systems with all unstable subsystems", Journal of Control and Decision, 1-14.

[30] Wang, J., Shen, Y., Wang, Z. (2016). "$H-/H_\infty$ asynchronous fault detection filter design for switched systems with time-varying delays", in 2016 35th Chinese Control Conference (CCC), Chengdu, China, Jul., 6690-6697.

[31] Wu, L., Lam, J. (2009). "Weighted H_∞ filtering of switched systems with time-varying delay: Average Dwell time approach", Circuits System Signal Process, 28(6), 1017-1036.

[32] Yang, G.H., Wang, H. (2009). "Simultaneous fault detection and control for uncertain linear discrete-time systems", IET Control Theory & Applications, 3(5), 583-594.

[33] Zhai, D., Lu, A.Y., Li, J.H., Zhang, Q.L. (2016). "Simultaneous fault detection and control for switched linear systems with mode-dependent average dwell-time", Applied Mathematics and Computation, 273, 767-792.

[34] Zhao, X., Zhang, L., Shi, P., Liu, M. (2012). "Stability and stabilization of switched linear systems with mode-dependent average dwell-time", IEEE Transactions on Automatic Control, 57(7), 1809-1815.

[35] Zhao, X.Q., Zhao, J. (2015). "Asynchronous fault detection for continuous-time switched delay systems", Journal of the Franklin Institute, 352(12), 5915-5935.

[36] Zhang, L., Cui, N., Liu, M., Zhao, Y. (2011). "Asynchronous filtering of discrete-time switched linear systems with average dwell time", IEEE Transactions on Circuits and Systems I, 58(5), 1109-1118.

[37] Zhong, Y., Chen, T., Chen, C. (2015). "Robust fault estimation for uncertain switched linear systems with time-varying delay", Journal of Central South University, 22(11), 4254-4262.

[38] Zhong, G.X., Yang, G.H. (2015). “Robust control and fault detection for continuous-time switched systems subject to a dwell-time constraint”, International Journal of Robust and Nonlinear Control, 25(18), 3799-3817.

[39] Zhong, G.X., Yang, G.H. (2016). “Simultaneous control and fault detection for discrete-time switched delay systems under the improved persistent dwell-time switching”, IET Control Theory & Applications, 10(7), 814-824.

[40] Zhu, K., Zhao, J. (2017). “Simultaneous fault detection and control for switched LPV systems with inexact parameters and its application”, International Journal of Systems Science, 48(14), 2909-2920.

[41] Zhuang, G., Xia, J., Chu, Y., Chen, F. (2014). “H_∞ mode-dependent fault detection filter design for stochastic Markovian jump systems with time-varying delays and parameter uncertainties”, ISA Transactions, 53(4), 1024-1034.

How to Cite this Article:

Ejtahed, S.H., Pariz, N., Karimpour, A. (2022).“Integrated fault detection and robust control for linear uncertain switched systems with mode-dependent time-varying state delay”, Control and Optimization in Applied Mathematics, 7(2): 1-34. doi: 10.30473/COAM.2022.62848.1192

Control and Optimization in Applied Mathematics (COAM)
Vol. 7, No. 2, Summer-Autumn 2022 (35-52), ©2016 Payame Noor University, Iran

DOI. 10.30473/coam.2022.64268.1206 (Cited this article)

Research Article

A New Optimization Method Based on Dynamic Neural Networks for Solving Non-convex Quadratic Constrained Optimization Problems

Kobra Mohammadsalahi[1], **Farzin Modarres Khiyabani**[1*], **Nima Azarmir Shotorbani**[1]

[1]Department of Mathematics, Tabriz Branch, Islamic Azad University, Tabriz, Iran.

Received: June 02, 2022 ; **Accepted:** November 28, 2022 .

Abstract. This paper presents a capable recurrent neural network, the so-called μRNN for solving a class of non-convex quadratic programming problems. Based on the optimality conditions we construct a new recurrent neural network (μRNN), which has a simple structure and its capability is preserved. The proposed neural network model is stable in the sense of Lyapunov and converges to the exact optimal solution of the original problem. In a particular case, the optimality conditions of the problem become necessary and sufficient. Numerical experiments and comparisons with some existing algorithms are presented to illustrate the theoretical results and show the efficiency of the proposed network.

Keywords. Quadratic programming, Recurrent neural network, Non-convex optimization.

MSC. 90C34; 90C40.

* Corresponding author
iauasrb082@gmail.com, salahi763@gmail.com, azarmir$_n im@yahoo.com$
$http://mathco.journals.pnu.ac.ir$

1 Introduction

Quadratic programming problems arise in a wide variety of scientific and engineering applications including regression analysis, image and signal processing, parameter estimation, filter design, robot control, etc. See for example [3, 2, 40] and a study of piecewise linear-quadratic programs by Cui et al. [12]. Optimization problems with nonlinear objective functions are usually approximated by second-order (quadratic) systems and solved approximately by standard quadratic programming (QP) techniques [29, 30]. In modeling many scientific problems, quadratic problems are obtained, such as smoothing quadratic regularization methods [6], minimizing condition number [10] and machine learning [19].

In recent years, convex QP has been studied by many researchers and many good results have been obtained. For example, one can see [14, 16, 45, 48, 49] where several methods for solving degenerate QP, convex quadratic bilevel programming, and convex quadratic minimax problems have been proposed. A major difference between convex and non-convex quadratic programming ($NCQP$) problems is that for the former any local minimizer is also a global minimizer whereas the latter may have many local minimizers. An analytic method for $NCQP$ subject to a set of linear constraints is presented in [4, 8]. Jeyakumar et al. [23] establish Lagrange multiplier conditions for global optimality of general non-convex quadratic minimization problems with quadratic constraints. They also obtain necessary global optimality conditions, which are different from the Lagrange multiplier conditions for special classes of QPs (see [42, 43] for more details). Moreover, Huang et al. [21] and Kong et al. [26] have used faster gradient-free proximal stochastic methods to solve the non-convex non-smooth optimization. QPs can also be solved indirectly using unconstrained optimization methods and optimization algorithms [34, 35, 46]. There are several direct methods to solve CQP and $NCQP$, such as first-order methods [9], interior point algorithms [7], accelerated gradient method [20] and combining stochastic adaptive cubic regularization [39].

The neural networks for solving mathematical programming problems were first proposed by Tank and Hopfild [44, 17]. Their work has inspired many researchers to investigate other neural network models for solving programming problems. One of the efficient methods to solve CQP and $NCQP$ is a recurrent neural network (RNN). The main advantage of RNN to optimization is that they can solve optimization problems in running time at orders of magnitude much faster than the most traditional optimization algorithms [5, 50]. These networks have been used in many scientific applications, such as complex-variable programming problems [28], classifiers with low model complexity [38], and non-smooth constrained pseudo-convex optimization [47].

Xue and Bian [48] developed a project neural network for solving degenerate QP problems with general linear constraints. In the theoretical aspects, the proposed Neural Network (NN) is shown to have complete convergence and finite time convergence. Effati and Ranjbar [14] presented a new NN for solving CQP problems. This model has a simple form, furthermore, it has a good convergence rate with a less number calculation operation than the old models. Besides, Nazemi [37] has used a capable NN for solving strictly CQP problems with general linear constraints. Nonetheless, Malek and Hosseinipour-Mahani [31] in their paper demonstrated that the use of the

$RNNs$ to solve $NCQP$ is efficient. In this work, based on a generalized KKT method, a modified RNN model called $M.RNN$ for a class of $NCQP$ problems involving a so-called Z-matrix has been proposed. By the study of the resulting dynamic system, it is shown that under given assumptions, steady states of the dynamic system are stable [1, 25].

There is similar research in the field of nonlinear and non-convex programming via neural networks. Effati et al. [13] presented an efficient projection neural network for solving bilinear programming problems. Also, Eshaghnezhad et al. [15] used a neurodynamic model to solve the nonlinear pseudo-monotone projection equation and its applications. Nonetheless, there are other types of numerical approaches to solving optimization problems via neural networks. Mansoori and Effati [32, 33] applied a parametric NCP-based recurrent neural network model to solve fuzzy non-convex optimization problems. Leung and Wang [27] proposed a neurodynamic method to solve minimax and bi-objective portfolio problems.

In this paper, an RNN network is designed to solve the $NCQP$ problems. We call this network μRNN. μRNN is similar to $M.RNN$ in reference [31]. The authors in [31] could have designed the $M.RNN$ more easily, thus reducing calculations and proofs. Accordingly, we have designed an RNN as μRNN that is simple and highly efficient. μRNN is stable in the sense of Lyapunov and has a high speed of convergence. Thus μRNN not only solves CQP and $NCQP$, but also has the following advantages:

- μRNN has a simple structure, so it is easy to design and use.
- The convergence rate of μRNN is sometimes equal to the convergence rate of $M.RNN$, and somtimes it is better.

In terms of run time, the μRNN network is similar to the $M.RNN$ network. In some problems where it is necessary to choose a small Rung-Kutta numerical method step length, the run time increases in both methods (as in Example 2). But in many convex and non-convex problems, the run time is about a few seconds.

2 Preliminaries

This section provides the necessary mathematical background used to study the proposed method and its usage. We list some necessary notations and introduce some necessary preliminary results in this section.

- $\|.\|$ denotes the l_2-norm on $\mathbb{R}^n$ ($\|x\| = (\sum_{i=1}^{n} x_i^2)^{1/2}$) and e_i denotes the column vector with a 1 in the i-th coordinate and 0's elsewhere.
- The space of all $n \times n$ symmetric matrices is denoted by S^n.
- For $g : \mathbb{R}^n \to \mathbb{R}$, $\nabla g(x) \in \mathbb{R}$ and $\nabla^2 g(x) \in \mathbb{R}^{n\times n}$ stand for gradient and the Hessian of g at x.
- The notation $A \succcurlyeq 0$ ($A \preccurlyeq 0$) shows that the matrix A is positive (negative) semi-definite.

- If there exists a non-zero vector $x \in \mathbb{R}^n$ such that $x^T Ax < 0$ then $A \not\succeq 0$.

Consider the following smooth non-convex quadratic optimization problem.

$$\begin{cases} Min\ f(X) \\ s.t:\ \ g_i(X) \leq 0,\ \ i = 1,2,\dots,m, \end{cases} \tag{1}$$

where $f, g_i : \mathbb{R}^n \to \mathbb{R}$ are defined by

$$f(X) = \frac{1}{2}X^T A_f X + b_f^T X + c_f,\ \ g_i(X) = \frac{1}{2}X^T A_{g_i} X + b_{g_i}^T + c_{g_i},$$

and $S_0 = \{X \in \mathbb{R}^n | g_i(X) \leq 0, i = 1,2,\dots,m\}$ is the feasible set. We suppose that $A_f \not\succeq 0$, and define H_f, H_{g_i} for $i = 1,2,\dots,m$ by

$$H_f = \begin{pmatrix} A_f & b_f \\ b_f^T & 2c_f \end{pmatrix},\ H_{g_i} = \begin{pmatrix} A_{g_i} & b_{g_i} \\ b_{g_i}^T & 2c_{g_i} \end{pmatrix}. \tag{2}$$

Definition 1. A matrix $A \in S^n$ is called a Z-matrix if $a_{ij} \leq 0$ for all $i \neq j$. Therefore any diagonal matrix is a Z-matrix.

In this paper, the RNN will be designed based on the following statement.

Proposition 1. (Jeyakumar et al. [23]) For general non-convex quadratic programming problem (1), let $X^* \in S_0$. If there exists $\lambda = (\lambda_1, \dots, \lambda_m)^T \in \mathbb{R}_+^m - \{0\}$ such that the conditions

$$\begin{cases} (a) & A_f + \sum_{i=1}^m \lambda_i A_{g_i} \geq \mathbf{O}, \\ (b) & (A_f x^* + b_f) + \sum_{i=1}^m (\lambda_i A_{g_i} x^* + \lambda_i b_{g_i}) = O, \\ (c) & \sum_{i=1}^m \lambda_i g_i(x^*) = 0, \end{cases} \tag{3}$$

hold, then X^* is a global minimizer of (1).

Remark 1. In the problem (1) when $m = 1$ and the strict feasibility condition holds, conditions (3) are necessary and sufficient conditions [24]. Also, for $m > 1$ the condition (a) of (3) is just a sufficient (not necessary) global optimality condition [31].

Theorem 1. (Jeyakumar et al. [22]) For the non-convex quadratic problem (1), suppose that H_f and H_{g_i}, $i = 1,\dots,m$ are Z-matrices and the Slater condition holds, that is, there exists $X_0 \in \mathbb{R}^n$ such that $g_i(X_0) < 0, i = 1,\dots,m$. Then a feasible point X^* is a globally optimal solution if and only if the conditions (3) hold.

Lemma 1. If A is a real square matrix, then:

$$X^T AX = X^T A^T X.$$

Proof. We know that any square matrix A can be written as [18]

$$A = \frac{1}{2}(A + A^*) + \frac{1}{2}(A - A^*) \equiv B + C,$$

where $B = \frac{1}{2}(A+A^*)$ is the Hermitian part of A, and $C = \frac{1}{2}(A-A^*)$ is the skew-Hermitian part of A. We have

$$X^T AX = X^T BX + X^T CX,$$

where $X^T CX = 0$. Thus, if the matrix A is real,

$$X^T AX = \frac{1}{2}X^T(A + A^T)X \Rightarrow X^T AX = X^T A^T X.$$

□

Corollary 1. Without loss of generality, in (1) assume that the matrices A_f and A_{g_i}, $i = 1,\dots,m$ are symmetric. If the matrix A in $X^T AX$ is not symmetric then we can replace it with $\frac{1}{2}(A + A^T)$.

Consider the following differential equation:

$$\dot{X}(t) = f(X(t)), \ X(t_0) = X_0 \in \mathbb{R}^n. \tag{4}$$

The following classical result on the existence and uniqueness of the solution to (4) holds.

Theorem 2. (Uniqueness and Existence) Assume that g is a continuous mapping from $\mathbb{R}^n$ to $\mathbb{R}^n$. Then for arbitrary $t_0 \geq 0$ and $X_0 \in \mathbb{R}^n$ there exists a local solution $X(t)$, $t \in [t_0, \tau)$ to (4) for some $\tau > t_0$. If g is locally Lipschitzian continuous at X_0 then the solution is unique, and if g is Lipschitzian continuous in $\mathbb{R}^n$ then τ can be extended to ∞.

Proof. See [11]. □

3 Recurrent Neural Network

Based on the optimization conditions (3), we design an RNN that converges to the optimal solutions to the problem (1). Consider

$$\begin{cases} \frac{dX}{dt} = -A_f X - b_f - \frac{1}{2}\sum_{i=1}^m \mu_i^2 (A_{g_i} X + b_{g_i}), \\ \frac{d\mu}{dt} = diag(\mu_1, \dots, \mu_m).g(X), \end{cases} \tag{5}$$

where $g(X) = (g_1(X), \cdots, g_m(X))^T$ and $\mu = (\mu_1, \cdots, \mu_m)^T$. Assuming $y = (X^T, \mu^T)^T$ and

$$\nabla g(X) = (\nabla g_1^T(X), \cdots, \nabla g_m^T(X))^T, \ \nabla g_i = \left(\frac{\partial g_i}{\partial x_1}, \frac{\partial g_i}{\partial x_2}, \cdots, \frac{\partial g_i}{\partial x_n}\right)^T.$$

We call system (5), μRNN and it is summarized as follows

$$\dot{y} = K\varphi(y), \tag{6}$$

where

$$\varphi(y) = \begin{pmatrix} -\nabla f(X) - \frac{1}{2}\sum_{i=1}^m \mu_i^2 \nabla g_i \\ diag(\mu_1, \cdots, \mu_m) g^T(X) \end{pmatrix}, \ y(t_0) = y_0, \tag{7}$$

and K is an adjusted parameter. A sufficiently large K could accelerate the μRNN.

Lemma 2. If $A_{n\times n}$ and $B_{m\times m}$ are negative definite, then the following matrix is negative definite.

$$\mathcal{F}=\left[\begin{array}{ccc} A_{n\times n} & | & C_{n\times m} \\ -- & -- & -- \\ -C^T_{m\times n} & | & B_{m\times m} \end{array}\right]_{(m+n)\times(m+n)}. \tag{8}$$

Proof. For all $X=(x_1,\cdots,x_n,x_{n+1},\cdots,x_{m+n})^T$ we have:

$$\begin{aligned} X^T\mathcal{F}X =& X^T\left[\begin{array}{ccc} A & | & C \\ - & - & - \\ -C^T & | & B \end{array}\right]X \\ =& (x_1,\ldots,x_n)A\begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}+(x_{n+1},\ldots,x_{m+n})B\begin{pmatrix} x_{n+1} \\ \vdots \\ x_{m+n} \end{pmatrix} \\ &\underbrace{+(x_1,\ldots,x_n)C\begin{pmatrix} x_{n+1} \\ \vdots \\ x_{m+n} \end{pmatrix}-(x_{n+1},\ldots,x_{m+n})C^T\begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}}_{=0}\le 0, \end{aligned}$$

so the matrix $\mathcal{F}$ is negative definite. □

For simplicity of our analysis, we let $K=1$. An indication of how the neural networks (6) and (7) can be implemented on hardware is provided in Figure 1.

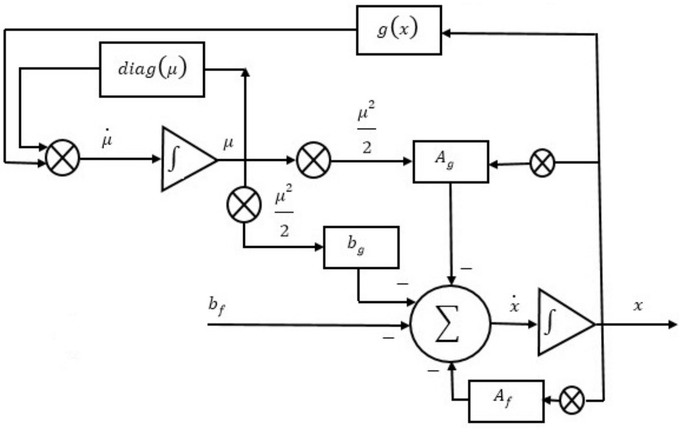


Figure 1: A simplified block diagram for the neural networks (6) and (7).

Theorem 3. Let $\Omega^*\subset\mathbb{R}^{n+m}$ be the set of equilibrium points of the system (6). If $y^*=(X^{*T},\mu^{*T})^T\in\Omega^*$ where $\mu^*=(\mu_1^*,\cdots,\mu_m^*)^T$, and

$$A_f+\sum_{i=1}^{m}\frac{\mu_i^{*2}}{2}A_{g_i}\geqslant 0,$$

then X^* is a global optimal solution of (1). Also, if H_f and $H_{g_i}, i = 1,\dots,m$ are Z-matrices and x^* is a global optimal solution of (1), then there exists $\mu^* \in \mathbb{R}^+ - \{O\}$ such that $(X^{*T}, \mu^{*T})^T$ is an equilibrium point of the μRNN.

Proof. Since y^* is an equilibrium point of the μRNN,

$$-A_f X^* - b_f - \frac{1}{2}\sum_{i=1}^{m} \mu_i^2 (A_{g_i} X^* + b_{g_i}) = 0, \forall i = 1,\dots,m; \mu_i g_i(X^*) = 0. \tag{9}$$

Let $\lambda_i^* = \frac{\mu_i^{*2}}{2}$, (9) satisfies assumptions (3), therefore X^* is a global optimal solution of (1). On the other hand, if H_f and H_{g_i} are Z-matrices then by using Theorem 1, conditions (3) are necessary and sufficient for the optimality of x^*. Therefore, if x^* is a global optimal solution of (1), then there exists $\lambda = (\lambda_1,\dots,\lambda_m) \geq 0,\ \lambda \neq 0$ such that conditions (3) hold. Substituting $\lambda_i = \frac{\mu_i^2}{2}$ into (3), we have

$$\begin{cases} (a1) & A_f + \sum_{i=1}^{m} \frac{\mu_i^2}{2} A_{g_i} \geq \mathbf{O}, \\ (b1) & (A_f X^* + b_f) + \sum_{i=1}^{m} \frac{\mu_i^2}{2}(A_{g_i} X^* + b_{g_i}) = O, \\ (c1) & \sum_{i=1}^{m} \frac{\mu_i^2}{2} g_i(X^*) = 0. \end{cases} \tag{10}$$

Since for all $X \in \Omega^*$, we have $g_i(X) \leq 0,\ \mu \geq 0$, so there are two cases, one $\mu_i = 0$ and the other $\mu_i \neq 0$. If $\mu_i = 0$ then $\mu_i g_i(X) = 0$, and if $\mu_i \neq 0$, then $\mu_i g_i(X) = 0$ is established again (from condition $(c1)$). Thus $(X^{*T}, \mu^{*T})^T$ is an equilibrium point of the μRNN. □

4 Stability and Convergence Analysis

In this section, the stability and convergence properties of the μRNN are exactly analyzed. It is clear that φ is continuously differentiable. Thus φ is locally Lipschitz continuous in $\mathbb{R}^{n+m}$ with positive constant $\|\nabla\varphi\|$ where $\nabla\varphi$ is the Jacobian matrix for $\varphi(y)$. So, by Theorem 2 the solution $y(t)$, for $t \in [t_0, \tau)$ to the μRNN, for some $\tau > t_0$ is unique as $\tau \to \infty$.

Definition 2. ([25]) The equilibrium point y^* is Lyapunov stable if, for each $\epsilon > 0$, there is $\delta > 0$ such that if $\|y(t_0) - y^*\| < \delta$, then $\|y(t) - y^*\| < \epsilon$, for all $t \geq t_0$.

Definition 3. ([41]) A set G is an invariant set for a dynamic system if every system trajectory which starts from a point in G remains in G for all future times.

Theorem 4. (Local Invariant Set Theorem) Consider an autonomous system of the form $\dot{X} = f(X)$, with f continuous, and let $V(X)$ be a scalar function with continuous first partial derivatives. Assume that

- for some $l > 0$, the region Ω_l defined by $V(X) < l$ is bounded.
- $\frac{d}{dt}V(X) \leq 0$ for all X in Ω_l.

Let R be the set of all points within Ω_l where $\frac{d}{dt}V(X) = 0$, M be the largest invariant set in R. Then, every solution $X(t)$ originating in Ω_l tends to M as $t \to \infty$.

Proof. See [41]. □

Theorem 5. Let y^* be an equilibrium point for (5) and $D \subset \mathbb{R}^{n+m}$ be a domain containing y^*. Let $V : D \to \mathbb{R}$ be a continuously differentiable function such that

$$V(y^*) = 0, \;\; V(y) > 0, \;\; \text{in } D - \{y^*\}, \tag{11}$$

$$\frac{dV}{dt}(y) \le 0, \;\; \text{in } D, \tag{12}$$

then, y^* is stable. Moreover, if

$$\frac{dV}{dt}(y) < 0, \;\; \text{in } D - \{y^*\},$$

then, y^* is asymptotically stable.

Proof. See [25]. □

Theorem 6. If $y = (X^T, \mu^T)^T \in \mathcal{M} \subset \mathbb{R}^{n+m}$, $X \in S_0$ and assume that

$$\mathcal{A} = A_f + \sum_{i=1}^{m} \frac{\mu_i^2}{2} A_{g_i} \succcurlyeq 0.$$

Then the Jacobian matrix $\nabla\varphi(y)$ of the mapping φ defined in (6) is a negative semi-definite matrix for all $y \in \mathcal{M}$.

Proof. It can be proved that the Jacobian matrix of φ is

$$\nabla\varphi = \left(\begin{array}{ccc} \left[-A_f - \sum_{i=1}^{m} \frac{\mu_i^2}{2} A_{g_i}\right]_{(n\times n)} & | & \left[-\nabla g^T(x).diag(\mu_1,\dots,\mu_m)\right]_{(n\times m)} \\ - & - & - \\ \left[diag(\mu_1,\dots,\mu_m).\nabla g(x)\right]_{(m\times n)} & | & \left[diag(g_1(x),\dots,g_m(x))\right]_{(m\times m)} \end{array}\right),$$

where $\nabla\varphi$ is an $(n+m)\times(n+m)$ matrix. $\nabla\varphi$ is exactly in the form of matrix $\mathcal{F}$ in Lemma 2. Since for any feasible point $y = (X^T, \mu^T)^T$ we have $g_i(X) \le 0$, $i = 1, \dots, m$ and using the assumption and Lemma 2 we obtain that $\nabla\varphi$ is a negative semi-definite matrix. □

Theorem 7. Let the assumptions of Theorem 6 hold. If $\Omega^* \subset \mathcal{M} \subset \mathbb{R}^{n+m}$ and S_0 is the feasible set of (1), then system (6) satisfies the following statements:

(i) Equilibrium points of (6) are stable in the sense of Lyapunov,

(ii) For all points $y(t_0) = (X_0^T, \mu_0^T)^T \in \mathcal{M}$ where $X_0 \in S_0$ the trajectory of $y(t)$ starting from $y(t_0)$ tends to Ω^* as $t \to \infty$.

(iii) For all $\hat{y} = (\hat{X}^T, \hat{\mu}^T)^T \in \Omega^*$ there exists a trajectory $y(t)$ with initial point $y(t_0) \in \mathcal{M}$ converges to $\hat{y}$, where $\hat{X}$ is a global optimal solution of problem (1).

Proof. (i). In this case, we prove that the equilibrium point is stable in the sense of Lyapunov. Consider the Lyapunov function $L : \mathcal{M} \to \mathbb{R}$ as

$$L(y) = \|y - \hat{y}\|^2, \tag{13}$$

where $\hat{y} \in \Omega^*$, and $y(t)$ is a trajectory obtained for system (6) starting from $y(t_0)$. Taking the time derivative from (13) and using (6), we have

$$\frac{dL(y)}{dt} = 2(y - \hat{y}).\frac{dy}{dt} = 2(y - \hat{y})\varphi(y).$$

Now by using the mean value theorem, there exists $\tilde{y}$ between y and $\hat{y}$, such that

$$\varphi(y) - \varphi(\hat{y}) = \nabla\varphi(\tilde{y})(y - \hat{y}).$$

By using Theorem 6 $\nabla\varphi(y)$ is negative definite, by multiplying both sides of the above equation by $(y - \hat{y})^T$, we have

$$(y - \hat{y})^T(\varphi(y) - \varphi(\hat{y})) = (y - \hat{y})^T\nabla\varphi(y)(y - \hat{y}).$$

Since the right side of the above equation is negative and $\varphi(\hat{y}) = 0$, we have

$$(y - \hat{y})^T\varphi(y) \leq 0 \Longrightarrow \frac{dL(y)}{dt} \leq 0.$$

(ii). To prove this part, we use Theorem 4. Note that system (6) is autonomous. The function $L : \mathcal{M} \to \mathbb{R}$, as stated in (13), is a scalar function with continuous first-order partial derivatives. For all $l > 0$, the following set

$$\Omega_l = \{y \in \mathcal{M} | L(y) \leq l\},$$

is bounded and for all $y \in inn(\Omega_l)$, we have $\frac{dL(y)}{dt} \leq 0$ where $inn(\Omega_l)$ is the interior of $\Omega(y)$. Now, Theorem 4 implies that every solution $y(t)$ of (6) starting from an arbitrary point belongs to $\mathcal{M}$, converges to a set of $M = \Omega^*$. It should be noted that in this discussion, the two sets R and M in Theorem 4 are the same as Ω^*.

(iii). If $y(t_0)$ is a feasible point, then the trajectory $y(t)$ obtained for the system of (6) starting from $y(t_0)$ cannot be unbounded, otherwise, we have $\lim_{k\to+\infty} \|y(t_k) - \hat{y}\|^2 = +\infty$ which leads to a contradiction with Lyapunov stability of system (6). As a result $\lim_{k\to+\infty} \|y(t_k) - \hat{y}\|^2 = 0$ or $M > 0$, if $\lim_{k\to+\infty} \|y(t_k) - \hat{y}\|^2 = 0$, then $\lim_{k\to\infty} y(t_k) = \hat{y}(t)$. If not, we have:

$$\lim_{k\to+\infty} \|y(t_k) - \hat{y}\|^2 = M > 0,$$

which implies that $\lim_{k\to+\infty} y(t_k) = \bar{y}$, where $\bar{y} \in \Omega^*$ and $\|\bar{y} - \hat{y}\|^2 = M$. Therefore, there is a trajectory $\{y(t_k)\}_{k=1}^{+\infty}$ that converges to an equilibrium point of system (6). According to Theorem 3, each equilibrium point of the system (6) is a global optimal solution of the problem (1). Therefore, the statement in item (iii) is true. □

In fact, all RNNs are autonomous.

Remark 2. In many problems, the attraction region of an equilibrium point is large, so the system (6) can be stable by starting from outside the feasible set.

5 Numerical Examples

In this section, some experiments are given to illustrate the efficiency and good performance of μRNN for solving optimization problems (1). The numerical testing was carried out on a Dell laptab (2.1 GHz, 2.00 GB of RAM) with the use of MATLAB (2008). The first three examples are for the non-convex problems and then the two examples are written for the convex problems. The numerical results of the non-convex examples are compared to the numerical results of Malk et al. [31] and convex examples are compared with Nazemi [36, 37].

Example 1. Consider the following non-convex quadratic optimization problem [31, 49].

$$\begin{array}{ll} \min & 8x_1x_2 + 3x_2^2 + 14x_1 + 12x_2 \\ s.t. & \quad 18x_1^2 + 8x_2^2 + 2x_1 - 1 \le 0, \\ & \quad 13x_1^2 - 4x_1x_2 + 8x_2^2 + 4x_2 - 1 \le 0, \\ & \quad 5x_1^2 - 10x_1x_2 + 5x_2^2 + 16x_1 + 18x_2 - 1 \le 0. \end{array} \tag{14}$$

By performing Network μRNN and starting from random initial points

$$y(t_0) = [rand, rand, rand, rand, rand\,]^T,$$

we obtain $X^* = (-0.21901076, -0.26801087)^T$ and $\mu^* = (2.00739048, 0.0001376, 0)^T$. From $\lambda_i = \frac{\mu_i^{*2}}{2}$ we have $\lambda^* = (2.014807306, 0, 0)^T$. We note that

$$\mathcal{A} = A_f + \frac{1}{2}\sum_{i=1}^{3} \mu_i^{*2} A_{g_i} = \begin{pmatrix} 72.5330630160 & 8 \\ 8 & 38.236916895 \end{pmatrix} \succ 0,$$

and $det(\mathcal{A}) = 2709.44 > 0$. In the system (6) a sufficiently large K could accelerate the μRNN. In other words, as the amount of K increases, the settling time of the system decreases significantly. Figure 2 presents the state trajectories of network μRNN with five random initial points and $K = 1$, $K = 1000$. According to the numerical results, it can be said that the convergence rate of networks μRNN and $M.RNN$ are equal.

Example 2. (Global solution for the CDT problem, [31]) Consider the following problem:

$$\begin{array}{ll} \min & f(d) = \frac{1}{2} d^T B d + b^T d \\ s.t. & \|A^T d + a\| \le \theta, \\ & \|d\| \le \delta, \end{array} \tag{15}$$

where $B \in S^n$, $A \in \mathbb{R}^{n\times m}(m \le n)$, $b \in \mathbb{R}^n$, $a \in \mathbb{R}^m$, $\theta > 0$ and $\delta > 0$. The problem (15) comes from applying the successive quadratic programming method and the trust-region technique to minimize a general function $q(X)$ subject to $h(X) = 0$ (for the details

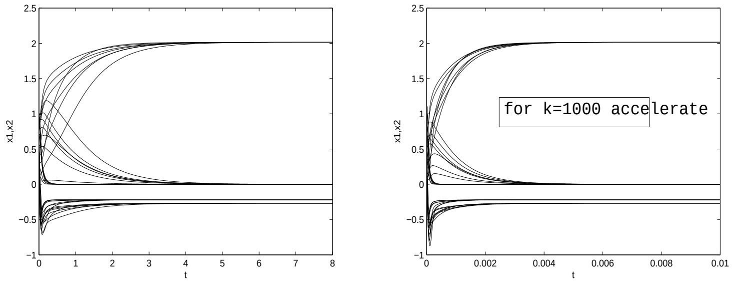


Figure 2: The transient behavior of the μRNN with different initial points for Example 1, $K = 1$ left side and $K = 1000$ right side.

see [31]). To solve the CDT problem (15) by the theory developed in this paper, we replace (15) by:

$$\begin{array}{ll} \min & f(d) = \frac{1}{2}d^TBd + b^Td \\ s.t. & g_1(d) = \|A^Td + a\|^2 - \theta^2 \leq 0, \\ & g_2(d) = \|d\|^2 - \delta^2 \leq 0.\delta, \end{array} \tag{16}$$

where for $n = m = 2$,

$$B = \begin{pmatrix} -2 & 0 \\ 0 & 2 \end{pmatrix}, \quad A = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \; a = (0, -6)^T, \; b = (0, -6)^T, \; \delta = 5, \; \theta = 5,$$

$$H_f = \begin{pmatrix} B & b \\ b & 0 \end{pmatrix}, \; H_{g_1} = 2\begin{pmatrix} AA^T & Aa \\ (Aa)^T & \|a\|2 - \theta^2 \end{pmatrix}, \; H_{g_2} = \begin{pmatrix} I_n & 0 \\ 0 & -\delta^2 \end{pmatrix}.$$

By performing network μRNN and starting from random initial points, we obtain

$$d_I^* = \begin{pmatrix} 3.98163 \\ 3.00000 \end{pmatrix}, \; \mu_I^* = \begin{pmatrix} 1.82558 \\ 1.82558 \end{pmatrix}, \tag{17}$$

and

$$d_{II}^* = \begin{pmatrix} -3.98163 \\ 3.00000 \end{pmatrix}, \; \mu_{II}^* = \begin{pmatrix} 1.82558 \\ 1.82558 \end{pmatrix}, \tag{18}$$

are two different global optimal solutions for Example 2.

Note that

$$\mathcal{A} = A_f + \sum_{i=1}^{2} \frac{\mu_i^{*2}}{2} A_{g_i} = \begin{pmatrix} 4.6654 & 0 \\ 0 & 8.6654 \end{pmatrix} \succ 0.$$

Figure 3 presents the state trajectories of network μRNN with 2 random initial points and $K = 100$.

Example 3. Consider the following non-convex programming problem ([31]).

$$\begin{array}{ll} \min & -3x_1^2 + x_2^2 + \frac{3}{2}x_3^2 + 2x_4^2 + 3x_5^2 \\ s.t. & \frac{1}{4}(x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 - 14) \leq 0, \\ & \frac{1}{4}(x_1^2 + x_2^2 + x_3^2 + (x_4 - 3)^2 + x_5^2 - 17) \leq 0, \\ & -x_2x_3 - 0.5x_3^2 - 1.5x_4 + x_5^2 - 2.5 \leq 0, \\ & -2x_2x_3 + 0.5x_4^2 - 9x_5 \leq 0, \quad -x_2x_3 - 9x_5 \leq 0. \end{array} \tag{19}$$

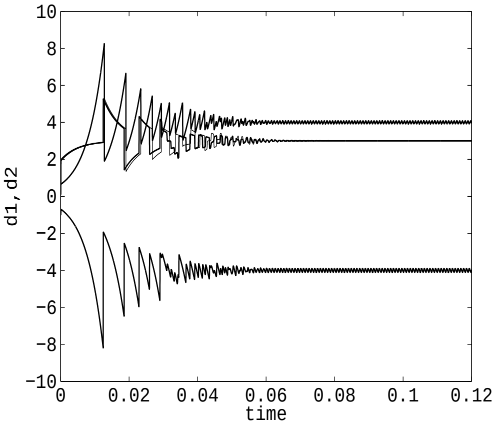


Figure 3: The transient behavior of the μRNN with different initial points for Example 2 for $K = 100$.

Using the network $M.RNN$, Malek and Hossinipour [31] obtained the optimal solutions as follows

$$X^* = (3.6051, 0.0000, 0.0000, 1.0000, 0.0556)^T,$$

$$X^{**} = (-3.6051, 0.0000, 0.0000, 1.0000, 0.0556)^T,$$

with

$$\lambda^* = \lambda^{**} = (5.2840, 6.7160, 0.0000, 0.0741, 0.0000)^T.$$

Now by using network μRNN, we obtain

$$x_I^* = \begin{pmatrix} 3.606758 \\ 0 \\ 0 \\ 0.999630707 \\ 0.05558959 \end{pmatrix}, \quad \mu_I^* = \begin{pmatrix} 3.249422 \\ 3.663398 \\ 0 \\ 0.3848944 \\ 0 \end{pmatrix}, \tag{20}$$

and

$$x_{II}^* = \begin{pmatrix} -3.605471 \\ 0 \\ 0 \\ 0.999979 \\ 0.0555559 \end{pmatrix}, \quad \mu_{II}^* = \begin{pmatrix} 3.251679 \\ 3.666151 \\ 0 \\ 0.384872 \\ 0 \end{pmatrix}. \tag{21}$$

Moreover, we get

$$\mathcal{A} = \begin{pmatrix} 18.014 & 0 & 0 & 0 & 0 \\ 0 & 26.014 & -0.1481 & 0 & 0 \\ 0 & -0.1481 & 27.0140 & 0 & 0 \\ 0 & 0 & 0 & 28.0881 & 0 \\ 0 & 0 & 0 & 0 & 30.0140 \end{pmatrix}. \tag{22}$$

In this problem, since H_f and H_{g_i}, $i = 1,\dots,m$ are Z-matrices, we can conclude by Theorem 1 X_I^* and X_{II}^* are two different global solutions of this problem. Figures 4 and 5 present the state trajectories of network μRNN with two random initial points and $K = 1$, $K = 300$, $K = 1300$. According to the numerical results, it can be said that the convergence rates of network μRNN are better than the network $M.RNN$.

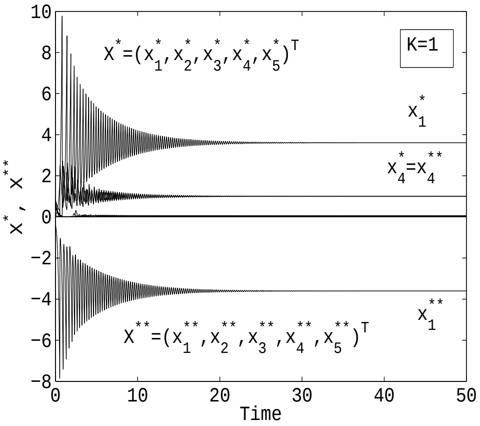


Figure 4: The transient behavior of the μRNN with different initial points for Example 3 for $K = 1$.

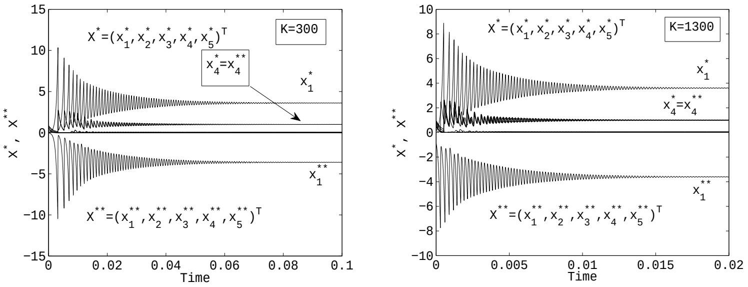


Figure 5: The transient behavior of the μRNN with different initial points for Example 3, $K = 300$ left side and $K = 1300$ right side.

To check the μRNN performance, we want to solve two examples for convex problems.

Example 4. Consider the following convex nonlinear optimization problem [36].

$$\begin{array}{ll} \min & x_1^2 + 2x_2^2 + 2x_1x_2 - 10x_1 - 12x_2 \\ s.t. & z_1 + 3x_2 \leq 8, \\ & x_1^2 + x_2^2 + 2x_1 - 2x_2 \leq 3. \end{array} \tag{23}$$

The exact solution is $X^* = (1,2)^T$. By performing the network μRNN and starting from 5 random initial points we get

$$x^* = \begin{pmatrix} 1.000075 \\ 2.000020 \end{pmatrix}, \quad \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix} = \begin{pmatrix} 0.001533 \\ 1.414195 \end{pmatrix}, \tag{24}$$

and note that

$$\mathcal{A} = A_f + \lambda_1 A g_1 + \lambda_2 A g_2 = \begin{pmatrix} 3.99994 & 2.0000 \\ 2.0000 & 5.999947 \end{pmatrix} \succ 0. \tag{25}$$

Figure 6 displays the transient behavior based on the network μRNN with 5 random initial points. All trajectories of the network converge to $X^* = (1, 2)^T$. Moreover, when the initial point is chosen as an infeasible point, the trajectory of the network μRNN still converges to X^*.

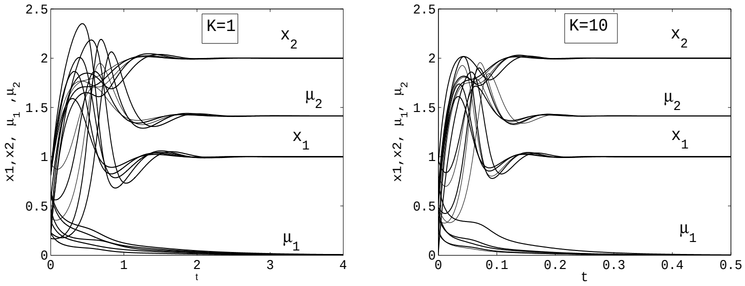


Figure 6: Transient behaviors of $y(t) = (X^T, \mu^T)^T$ of network μRNN with 5 various initial points in Example 4 for $K = 1$ and $K = 10$.

Example 5. Consider the following convex nonlinear optimization problem [37]:

$$\begin{array}{ll} \min & 10x_1^2 + 2x_2^2 + 2x_3^2 - 2(x_1x_2 + 3x_1x_3 - x_2x_3) \\ s.t. & -1 \le x_2 - x_1 \le 0, \\ & -1 \le x_3 - 3x_1 \le 1, \\ & 1 \le x_2 + x_3 \le 2. \end{array} \tag{26}$$

The first constraint is equivalent to $(x_2 - x_1)(x_2 - x_1 + 1) \le 0$. Similarly, the constraints of the problem (26) are equivalent to the following constraints.

$$\begin{cases} x_1^2 + x_2^2 - 2x_1x_2 + x_2 - x_1 \le 0, \\ 9x_1^2 + x_3^2 - 6x_1x_3 - 1 \le 0, \\ x_2^2 + x_3^2 + 2x_2x_3 - 3x_2 - 3x_3 + 2 \le 0. \end{cases}$$

Since the problem is convex, the unique exact solution is $X^* = (\frac{1}{4}, \frac{1}{4}, \frac{3}{4})^T$. By performing the network μRNN and starting from 4 random initial points we obtain

$$x^* = \begin{pmatrix} 0.250452 \\ 0.248640 \\ 0.75135 \end{pmatrix}, \quad \begin{pmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \end{pmatrix} = \begin{pmatrix} 0.146195 \\ 0 \\ 1.999987 \end{pmatrix}. \tag{27}$$

Note that

$$\mathcal{A} = A_f + \sum_{i=1}^{3} \lambda_i A g_i = \begin{pmatrix} 5.02 & -0.50 & -1.50 \\ 2.00 & 5.99 & 2.01 \\ -1.50 & 2.01 & 2.51 \end{pmatrix} \succ 0, \tag{28}$$

and $det(\mathcal{A}) = 8.128$. Figures 7 and 8 show that the trajectories of the network μRNN to solve the above problem with 4 random initial points and $K = 15, 120, 1000$, converge to the optimal solution of this problem. It is seen that the proposed network converges to the exact solution X^* independent of the way that we may choose the starting points.

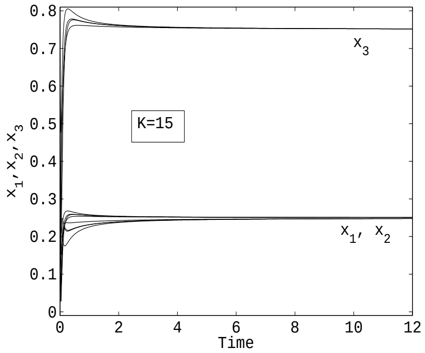


Figure 7: The transient behavior of the μRNN with 4 initial points for Example 5 for $K = 15$.

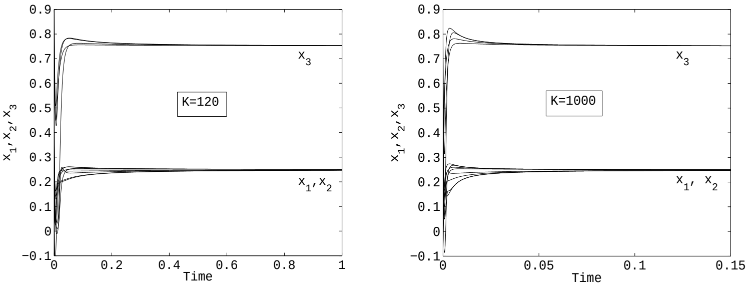


Figure 8: The transient behavior of the μRNN with 4 initial points for Example 5, $K = 120$ left side and $K = 1000$ right side.

6 Conclusion

In this paper, we presented a recurrent neural network (μRNN) for solving non-convex quadratic problems based on the Lyapunov theory. This network is a modified $M.RNN$ and has a simpler structure compared to that. The capability of this network is equal to and sometimes better than $M.RNN$. Finally, to show the efficiency of the proposed network, some numerical examples (convex and non-convex) were presented.

References

[1] Bacciotti, A. (1992). "Local stabilizability of nonlinear control systems", Advances in Mathematics for Applied Sciences.

[2] Bazaraa, M.S., Shetty, C.M. (1990). "Nonlinear programming theory and algorithms", Wiley and Sons, New York.

[3] Bertsekas, D.P. (1989). "Parallel and distributed numerical methods", Prentice-Hall, Englewood, Cliffs, NJ.

[4] Best Michael, J. (2017). "Quadratic programming with computer programs", Advances in Applied Mathematics, University of Waterloo, Canada, CRC Press.

[5] Beyer, D., Ogier, R. (2000). "Tabu learning: A neural networks search method for solving non-convex optimization problems", IEEE International Joint Conference Neural Networks 2, 953-961.

[6] Bian, W., Chen, X. (2013). "Worst-case complexity of smoothing quadratic regularization methods for non-Lipschitz optimization", SIAM, Journal of Optimization, 23 (3), 1718-1741.

[7] Bian, W., Chen, X., Ye, Y. (2014). "Complexity analysis of interior point algorithms for non-Lipschitz and non-convex minimization", Mathematical Programming, 149(1-2), 301-327.

[8] Boob, D.P. (2020). "Convex and structured non-convex optimization for modern machine learning: Complexity and algorithms", Georgia Institute of Technology.

[9] Carmon, Y., Duchi, J.C. (2020). "First-order methods for non-convex quadratic minimization", SIAM Review, 62(2), 395-436.

[10] Chen, X., Womersley, R., Ye, J. (2011). "Minimizing the condition number of a Gram matrix", SIAM Journal on Optimization, 21, 127-148.

[11] Chicone, C. (2006). "Ordinary differential equations with applications"; Second edition, Springer-Verlag, New York.

[12] Cui, Y., Chang, T. H., Hong, M., Pang, J.S. (2020). "A study of piecewise linear-quadratic programs", Journal of Optimization Theory and Applications, 1-31.

[13] Effati, S., Mansoori, A., Eshaghnezhad, M. (2015). "A projection neural network for solving bilinear programming problems", Neurocomputing, 1-20.

[14] Effati, S., Ranjbar, M. (2011). "A novel recurrent nonlinear neural network for solving quadratic programming problems", Applied Mathematical Modelling, 33, 1688-1695.

[15] Eshaghnezhad, M., Effati, S., Mansoori, A. (2016). "A neurodynamic model to solve nonlinear pseudo-monotone projection equation and its applications", IEEE Transactions on Cybernetics, 1-13.

[16] Gao, X.B., Liao, L.Z., Xue, W. (2004). "A neural network for a class of convex quadratic minimax problems with constraints", IEEE Transactions on Neural Networks, 15(3), 622-628.

[17] Hopfield, J.J., Tank, D. (1985). "Neural computation of decisions in optimization problem", Biod. Cybern., 52, 141-152.

[18] Horn, R.A., Johnson, C.R. (1990). "Matrix Analysis", Cambridge University Press.

[19] Huan, L., Fang, C., Zhouchen, L. (2020). "Accelerated first-order optimization algorithms for machine learning", Proceedings of the IEEE, doi: 10.1109/JPROC.2020.3007634.

[20] Huan, L., Lin, Z. (2019). "Provable accelerated gradient method for non-convex low-rank optimization", Machin Learning.

[21] Huang, F., Gu, B., Huo, Z., Xhen, S., Huang, H. (2019). "Faster gradient-free proximal stochastic methods for non-convex nonsmooth optimization", The Thirty-Third AAAI Conference on Artificial Intelligence (AAAI-19), 1503-1510.

[22] Jeyakumar, V., Lee, G.M., Li, G.Y. (2009). "Alternative theorems for quadratic inequality systems and global quadratic optimization", SIAM Journal on Optimization, 20(2), 983-1001.

[23] Jeyakumar, V., Rubinov, A.M., Wu, Z.Y. (2007). “Non-convex quadratic minimization problems with quadratic constraints: Global optimality conditions”, Mathematical Programming, series, A 110, 521-541.

[24] Jeyakumar V., Srisatkunarajah S. (2009). “Lagrange multiplier necessary condition for global optimality for non-convex minimization over a quadratic constraint via S-lemma”, Optimization Letters, 3, 23-33.

[25] Khalil, H.K. (2002). “Nonlinear systems”, Prentice Hall, Third edition.

[26] Kong, W., Melo, J.G., Monteiro, R.D.C. (2019). “An efficient adaptive accelerated inexact proximal point method for solving linearly constrained non-convex”, Composite Problems, 1-28.

[27] Leung, M.F., Wang, J. (2019). “Minimax and bi-objective portfolio selection based on collaborative neurodynamic optimization”, IEEE Transactions on Neural Networks and Learning Systems, 1-12.

[28] Liu, S., Jiang, H., Zhang, L., Mei, X. (2020). “A neurodynamic optimization approach for complex-variables programming problem”, Neural Networks, 129, 280-287.

[29] Lu, S. (2018). “First-Order methods of solving non-convex optimization problems: Algorithms, convergence, and optimality”, Electrical and Electronics Commons.

[30] Luenberger, D.G. (1948). “Introduction to Linear and Nonlinear Programming”, Reading MA: Addison-Wesley.

[31] Malek, A., Hosseinipour-Mahani, N. (2015). “Solving a class of non-convex quadratic problems based on generalized KKT conditions and neurodynamic optimization technique”, Kybernetika, 51, 890-908.

[32] Mansoori, A., Effati, S. (2019). “An efficient neurodynamic model to solve nonlinear programming problems with fuzzy parameters”, Neurocomputing, 334, 125-133.

[33] Mansoori, A., Effati, S. (2019). “Parametric NCP-based recurrent neural network model: A new strategy to solve fuzzy non-convex optimization problems”, IEEE Transactions on Systems, Man, and Cybernetics: Systems, 1-10.

[34] Modarres, F., Hassan, M.A., Leong, W.J. (2011). “A symmetric rank-one method based on extra updating techniques for unconstrained optimization”, Computers and Mathematics with Applications, 62, 392-400.

[35] Nasiri, J., Modarres Khiyabani, F. (2018). “A whale optimization algorithm (WOA) approach for clustering”, Cogent Mathematics and Statistics, doi.org/10.1080/25742558.2018/1483656.

[36] Nazemi, A.R. (2012). “A dynamic system model for solving convex nonlinear optimization problems”, Communications in Nonlinear Science and Numerical Simulation, 17, 1696-1705.

[37] Nazemi, A.R. (2014). “A neural network model for solving convex quadratic programming problems with some applications problems”, Engineering Applications of Artificial intelligence, 32, 54-62.

[38] Pant, H., Soman Jayadeva, S., Bhaya, A. (2020). “Neurodynamical classifiers with low model complexity”, Neural Networks, 132, 405-415.

[39] Park, S., Jung, S.H., Pardalos, P.M. (2020). “Combining stochastic adaptive cubic regularization with negative curvature for non-convex”, Journal of Optimization Theory and applications, 184 (3), 953-071.

[40] Rudnick-Cohen, E., Herrmann, J.W., Azarm, S. (2020). “Non-convex feasibility robust optimization via scenario generation and local refinement”, Journal of Mechanical Design, 142 (5), 1-10.

[41] Slotine, J.J.E., Li, W. (1990). “Applied Nonlinear Control”, Wiley and Sons, New York.

[42] Strekalovsky, A.S. (2018). “On non-convex optimization problems with D. C. equality and inequality constraints”, IFAC papers online, 51-32, 895-900.

[43] Strekalovsky, A. (2019). “Nonconvex optimization: From global optimality conditions to numerical methods”, AIP Conference Proceedings 2070, 020015, 1-4.

[44] Tank, D.W., Hopfield, J.J. (1986). “Simple neural optimization networks: On A/D converter, signal decision circuit and a linear programming circuit”, IEEE Transactions on Circuits and Systems, 33, 533-541.

[45] Tian, Y., Lu, C. (2011). “Nonconvex quadratic formulations and solvable conditions for mixed integer quadratic programming problems”, Journal of Industrial and Management Optimization, 7 (4), 1027-1039.

[46] Valizadeh Oghani, A., Khiabani, F. M., Farahmand, F.H. (2020). “Data envelopment analysis technique to measure the management ability: Evidence from Iran cement industry”, Cogent Business and Management, doi.org/10.1080/23311975.2020.1801960.

[47] Xu, C., Chai, Y., Qin, S., Wang, Z., Feng, J. (2020). “A neurodynamic approach to nonsmooth constrained pseudo convex optimization problem”, Neural Networks, 124, 180-192.

[48] Xue, X., Bian, W. (2007). “A project neural network for solving degenerate convex quadratic program”, Neurocomputing, 70, 2449-2459.

[49] Yan, Y. (2014). “A new nonlinear neural network for solving quadratic programming problems”, Springer International Publishing Switzerland, 347-357.

[50] Yang, Y., Cao, J., Xu, X., Liu, J. (2012). “A generalized neural network for solving a class of minimax optimization problems with linear constraints”, Applied Mathematics and Computation, 218, 7528-7537.

How to Cite this Article:

Mohammadsalahi, K., Modarres Khiyabani, F., Azarmir Shotorbani N., (2022). “A new optimization method based on dynamic neural networks for solving non-convex quadratic constrained optimization problems”, Control and Optimization in Applied Mathematics, 7(2): 35-52. doi: 10.30473/coam.2022.64268.1206

Control and Optimization in Applied Mathematics (COAM)
Vol. 7, No. 2, Summer-Autumn 2022 (53-75), ©2016 Payame Noor University, Iran

DOI. 10.30473/coam.2022.62968.1195 (Cited this article)

Research Article

Solving Nonlinear Hydraulic Equations of Water Distribution Networks by Using a Trust-Region Method

Mahdi Ahmadnia[1], **Reza Ghanbari**[1,*], **Khatere Ghorbani-Moghadam**[2]

[1]Faculty of Mathematical Sciences, Department of Applied Mathematics, Ferdowsi University of Mashhad, Mashhad, Iran.
[2]Mosaheb Institute of Mathematics, Kharazmi University, Tehran, Iran.

Received: February 10, 2022; **Accepted:** November 27, 2022 .

Abstract. In a water distribution network, in order to analyze and determine its parameters such as head and flow rate, we have to solve nonlinear hydraulic equations in each component of the network. Contrary to most of the water distribution network simulation software, solving these equations by using the gradient method, we propose a trust-region method to solve them, as the trust-region method is newer than the gradient method and has well worked in mathematical problems. To prove the effectiveness of our method, we made a comparison between our proposed method and the well-known gradient method. The results show that the trust-region method is convergent in all instances, but the gradient method diverges when the dimension of nonlinear hydraulic equations of water distribution networks increases. In addition, our results convince us that the solution obtained from the trust-region method is more accurate compared to the gradient method. Thus, using the trust-region method in solving the network equations can lead to a better hydraulic analysis of the network.

Keywords. Water distribution network, Hydraulic equations, Nonlinear equation, Trust-Region method.

MSC. 90C34; 90C40.

* Corresponding author
ahmadnia.mahdi74@gmail.com, rghanbari@um.ac.ir, kh.ghorbani@khu.ac.ir

1 Introduction

Head and flow regulation in water distribution networks (WDNs) is a significant concern for water utilities. Effective head and flow control throughout pipe networks are essential to ensure rational sufficient service levels to customers for daily fluctuating demand patterns. Simulation models are applied to estimate the distribution of pipe flow rates and residual nodal heads (pressures) within pipe networks, in which these hydraulic parameters have to be computed for different loading and operating conditions [3]. For finding head, flow, and also hydraulic analysis some nonlinear equations have to be solved [15, 52].

There are many methods done to solve the nonlinear equations in WDNs [40]. An iterative method for solving these equations was first proposed by Cross [12] (This method was also used for solving gas network equations; see [8]). Cross proposed an approach that is used to solve the equations Q, ΔQ, and H related to WDNs. The number of calculations required for convergence in the Cross method depends on the convergence criterion (accuracy of the solution), the initial solution, the flow rate of the pipes, and also the resistance of the pipes (R).

Cross's method [12] solves only one equation at a time with some assumptions, such as ignoring the effect of adjacent loops. Martin and Peter [34] proposed the Newton–Raphson method for solving nonlinear WDNs equations, which solves all equations simultaneously. This method is used to solve flow and node equations. However, their approach works better in solving node equations than flow equations. Shamir and Howard [45] and Zarghamee [56] used the Newton–Raphson method for networks with valves and pumps. They investigated the convergence conditions of the Newton–Raphson method and the possibility of insolvable problems. In each iteration of the Newton–Raphson method, in order to determine the correction of the pipe discharge values, a linear equation system must be solved. This linear system is formed by the Jacobian matrix in each iteration. Liu [30] modified the Jacobian matrix to a diagonal matrix. He demonstrated that by using the diagonal matrix, the speed in solving the linear equation is accelerated fast in each iteration. Moosavian and Jaefarzadeh [36] illustrated that the approach proposed by Liu has two disadvantages. One of them is the lack of convergence in large WDNs, and the other is high fluctuations to achieve convergence. So, they suggested that some network pipes must be temporarily removed during the analysis process. They also halved the amount of correction per repetition to reduce fluctuations, but they increased the number of repetitions until the final solution was reached; see [36]. It is also important how to choose the initial solution in this method. If the wrong initial solution is chosen, then the Newton–Raphson method diverges. There are more suggestions for improving the convergence of the Newton–Raphson method. Most of these suggestions correct the pipe flow rates per repetition (see [4, 14, 28, 29, 39, 43, 44, 46, 47]). Based on the Broyden method [9], Tabesh [49] provided a relation for finding the correction rate of flow rate in each iteration. Tanyimboh et al. [50] proposed a line search method in order to accelerate the convergence.

Wood and Charles [53] used the linear theory method to solve flow equations. They showed that their proposed method is too fast and independent of the initial solution.

Also, Collins and Johnson [10] and Isaacs and Mills [22] used the linear theory for solving node equations, which is usually better for solving flow equations than node equations. Each iteration of this method assumes a value for the flow rate of the pipes based on the flow rate obtained from the previous iterations. Using this hypothetical value, a linear equation system that is approximately equivalent to a network equation system is solved. During the convergence process of this method, fluctuations occur. These fluctuations reduce the convergence rate of the linear theory method. Due to these fluctuations, Nielsen [37] proposed using a combination of linear theory and the Newton–Raphson method, so the initial solution of the Newton–Raphson method is produced by linear theory. For the purpose of increasing the speed of convergence, Bhave [6] provided a method for determining the hypothetical flow rates of each iteration. He suggested using the hypothetical mean flow rate of mth and the flow rate obtained in the mth repeat as the hypothetical flow rate of $m+1$.

The most common method currently used in many network simulation software, such as EPANET, is the gradient method. Todini and Pilati [51] introduced this method for WDNs. The gradient method finds the solution of the equations in each iteration solving a linear equation system. Although more equations need to be solved in this method, Todini and Pilati [51] have shown that this method is very computationally robust. Powell [43] solved this algorithm by using Lagrange coefficients for optimization problems with equality constraints. The gradient method is somewhat independent of the initial solution, but if the initial solution is close to the final solution, then the degree of convergence of this method is at least two [5]. See other works for solving WDNs in [2, 7, 13, 18, 20, 21, 26, 24, 32, 33]. See the summary of literature review in Table 1.

The trust region is a newer method compared to the gradient method. So far, the trust-region method for solving equations of WDNs has not been investigated and we use the trust-region method to solve WDN equations. The results show that the trust-region method is more accurate in solving WDN equations compared to the gradient method. So, the trust-region method can provide a better hydraulic analysis of WDN. Here, we use the trust-region method for solving hydraulic equations in a WDN.

The rest of our work is organized as follows. In Section 2, we provide the necessary definitions. We propose a trust-region method for solving flow equations in Section 3. In Section 4, we implement our proposed algorithm on several test problems and compare them with the gradient method. Finally, conclusion will be resented in Section 5.

2 Preliminaries

WDNs are designed in different types. Serial networks, branching networks, looped networks, and composite networks are among the types of WDNs. Here, we give some basic definitions of WDNs.

Definition 1 (Node)**.** [48] The point of intersection of several pipes, as well as the starting and endpoints of each pipe, is called a node.

Table 1: Survey of literature review

Names of authors	**Type of problem**	**Solving methods**
Cross's method [12]	Nonlinear WDNs equations	Iterative method
Martin and Peter [34]	Nonlinear WDNs equations	Newton–Raphson method
Shamir and Howard [45]	Nonlinear WDNs equations	Newton–Raphson method
Zarghamee [56]	Nonlinear WDNs equations (Networks with valves and pumps)	Newton–Raphson method
Liu [30]	Nonlinear WDNs equations	Linearization approach
Moosavian and Jaefarzadeh [36]	Nonlinear WDNs equations	Modified Liu's method
Tabesh [49]	Nonlinear WDNs equations	Iterative algorithm based on the Broyden method
Tanyimboh et al. [50]	Nonlinear WDNs equations	Iterative algorithm based on line search method
Wood and Charles [53]	Nonlinear WDNs equations	Linear theory method to solve flow equations
Collins and Johnson [10]	Nonlinear WDNs equations	Linear theory for solving node equations
Isaacs and Mills [22]	Nonlinear WDNs equations	Linear theory for solving node equations
Nielsen [37]	Nonlinear WDNs equations	Hybrid algorithm based on the linear theory and the Newton–Raphson method
Bhave [6]	Nonlinear WDNs equations	Iterative algorithm for determining the hypothetical flow rates of each iteration
Todini and Pilati [51]	Nonlinear WDNs equations	Gradient method
Powell [43]	Nonlinear WDNs equations	Iterative algorithm by using Lagrange coefficients

Definition 2 (Consumption node)**.** [48] The nodes from which water is removed are called consumption nodes.

Definition 3 (Source node)**.** [48] The nodes through which water enters the network are called source nodes.

Definition 4 (Loop)**.** [48] The closed environment that creates several interconnected pipes is called a loop.

Each WDN consists of different components, such as storage tanks, pipes, valves, pumps, and so on; see [48]. Each of these components can affect the head and flow.

The characteristics of each of these components are described by the head-flow relationship in that component. For example, the head-flow relationship for network pipes is obtained from the following relationship (see [48]):

$$h_{ij} = H_i - H_j = R_{ij} Q_{ij} |Q_{ij}|^{n-1}, \tag{1}$$

where h_{ij} is the decrease of energy (h_{ij} shows the amount of head loss in the pipe ij). Moreover, Q_{ij} and R_{ij} represent the current passing through the pipe ij and the resistance constant of the pipe ij, respectively. Also, H_i and H_j are equal to the head in nodes i and j, respectively. The direction of water flow in the pipes of a distribution network is always from more heads to fewer heads. When water is transferred from one node to another through a pipe, its hydraulic energy is reduced due to friction [27]. This shows the decrease in energy in relation to (1), which is denoted by the symbol h_{ij}. In other words, h_{ij} is equal to the amount of head loss in the pipe ij. Moreover, n is the power of water flow, and to calculate it, we use the Hazen–Williams method. In the Hazen–Williams method, the value of n is considered equal to $1,852$, and the value of R_{ij} is obtained from the following equation:

$$R_{ij} = \frac{\alpha . L_{ij}}{C_{HW_{ij}}^{1.852} . D_{ij}^{4.87}}, \tag{2}$$

where α is equal to 10.675 (in the metric system) and L_{ij}, $C_{HW_{ij}}$, and D_{ij} indicate the length of the pipe ij (in meters), the Hazen–Williams coefficient of the pipe ij, and the diameter of the pipe ij (in meters), respectively. The Hazen–Williams coefficient depends on the characteristics of the pipe, such as the material, age, and so on, and it is determined and announced by the pipe's manufacturers [23].

If the head is specified at both ends of a pipe, then the value of Q_{ij} is calculated based on (1) as follows:

$$Q_{ij} = \left(\frac{\left| H_i - H_j \right|}{R_{ij}} \right)^{\left(\frac{1}{n}\right)} sgn(H_i - H_j), \tag{3}$$

where sgn is the sign function. For the hydraulic analysis, as well as determining the parameters of a WDN, the nonlinear equations in the network components must be solved. These equations are obtained according to the network components and using the two basic laws of continuity and energy survival.

Definition 5 (Continuity rule)**.** [48] According to this rule, the sum of the input current values in each node is equal to the sum of the output current values from that node. In other words,

$$\left(\sum_{ij \in IJ_j} Q_{ij} \right)_{in} - \left(\sum_{ij \in IJ_j} Q_{ij} \right)_{out} = q_j, \qquad \text{for all } j = 1, \ldots, NJ, \tag{4}$$

where q_j is the input or output flow rate of node j, IJ_j represents all the pipes connected to node j, and NJ is the number of nodes in the network. Some of the equations obtained from this relation may not be independent. In fact, the number of independent equations obtained from (4) is equal to the number of consumption nodes. Therefore, (4) is not used for source nodes.

Definition 6 (Energy rule)**.** [48] This rule is used for all loops in the distribution network to write equations. According to this law, the total head loss inside a loop is considered equal to zero, that is,

$$\sum_{ij \in IJ_L} h_{ij} = \sum_{ij \in IJ_L} R_{ij} Q_{ij} |Q_{ij}|^{n-1} = 0, \qquad \text{for all } L = 1, \ldots, NL, \tag{5}$$

where, IJ_L represents all loop pipes L and NL is equal to the number of network loops. If the direction for water flow in the pipe is clockwise, then the head loss sign for that pipe in (5) will be positive; otherwise, it will be negative (In the equations written in terms of flow, the mentioned symbol is included in the coefficient of resistance of the pipe).

Note: The direction of water flow in the pipes cannot be determined before solving the network equations. For this reason, first, the direction of flow in the pipes is hypothetically determined, and the network equations are written based on it. After solving the equations, whenever the flow of a pipe is obtained as a negative number, it means that the direction of flow in this pipe is assumed to be the opposite. However, the amount of current is correct, and there is no need to solve the equations again.

According to Definitions 5 and 6, different equations can be written for the analysis of WDNs, including Flow, node, ring equations, ΔH equations, and head-flow equations. Here, we solve equations of the flow system. The number of flow equations is equal to the number of pipes in the network, and the unknown of these equations is the flow rate of the pipes. Flow equations are written by using both the laws of continuity and energy. In the flow equations, the equations derived from the law of continuity are all linear, and the equations derived from the law of energy are all nonlinear. By combining (4) and (5), the system of flow equations is obtained.

Now, using the laws of continuity and energy, we write the flow equations for the network in Figure 1 as follows:

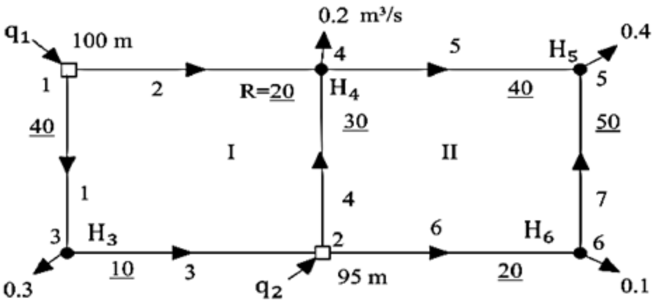


Figure 1: A small sample of WDN [51].

Node 3 : $Q_1 - Q_3 - 0.3 = 0,$

$$
\begin{aligned}
&\text{Node } 4: Q_2 + Q_4 - Q_5 - 0.2 = 0,\\
&\text{Node } 5: Q_5 + Q_7 - 0.4 = 0,\\
&\text{Node } 6: Q_6 - Q_{65} - 0.1 = 0,\\
&\text{Loop I}: 20Q_2|Q_2|^{n-1} - 30Q_4|Q_4|^{n-1} - 10Q_3|Q_3|^{n-1} - 40Q_1|Q_1|^{n-1} = 0,\\
&\text{Loop II}: 40Q_5|Q_5|^{n-1} - 50Q_7|Q_7|^{n-1} - 20Q_6|Q_6|^{n-1} + 30Q_4|Q_4|^{n-1} = 0.
\end{aligned} \tag{6}
$$

By solving a system of equations (6), the flow rate of all network pipes is obtained. Methods such as Cross, Newton-Raphson, linear theory, and Gradient have been used to solve the equations of water supply networks. In this paper, the trust-region method is used to solve these equations. We will explain more about this in the following section.

3 Trust-Region Method for Solving the System of Flow Equations

As a kind of numerical method for solving nonlinear optimization problems, the trust-region method has been widely studied in recent decades [55]. The trust-region method was first used to solve unconstrained optimization problems by Powell [43], of which the distance between the iteration points in the current iteration cycle and the cycle before should be limited. In this method, by applying the Taylor-series expansion, a quadratic model is used to approximate the objective function. It can be thought that there is a neighborhood around the current iteration point within which we trust the surrogate model. Such a neighborhood is called a trust region. The size of the trust region is tuned by the performance of the algorithm in the previous search; see [38]. See other works [11, 16, 35, 54].

The equations of WDNs are nonlinear. So far, different methods have been proposed to solve nonlinear equations; see [1, 17, 19, 25, 41]. One of the desirable and efficient methods to solve the system of nonlinear equations is to use the existing methods in optimization. In other words, solving the system of equations is equivalent to solving an optimization problem. Therefore, instead of solving the system of equations, the equivalent optimization problem can be solved. To use these methods, the system of equations must first be turned into an optimization problem. Then, using the trust-region method solves it.

In this section, first, we explain how to convert a system of equations into an optimization problem and then propose a trust-region algorithm to solve it.

3.1 Converting the System of Equations into an Optimization Problem

In this section, we explain how to convert a system of equations to an optimization problem. Consider the system of equations $r(x) = 0$, in which $r : \mathbb{R}^n \to \mathbb{R}^n$ is a vector function as follows:

$$r(x) = \begin{bmatrix} r_1(x) \\ \vdots \\ r_n(x) \end{bmatrix}, \tag{7}$$

where $r_i : \mathbb{R}^n \to \mathbb{R}$, for $i = 1, 2, \dots, n$, is the ith equation of the system $r(x) = 0$. The following lemmas can be obtained easily.

Lemma 1. The vector x^* is the solution to the system of equations $r(x) = 0$ if and only if $\|r(x^*)\| = 0$.

Lemma 2. The vector x^* is the solution to the system $r(x) = 0$ if and only if the optimal solution to the problem $\min \|r(x)\|$ be zero.

Proof. Suppose that x^* is the solution to $r(x) = 0$, based on Lemma 1, $\|r(x^*)\| = 0$. On the other hand, $\|r(x)\|$ is always positive, so, the solution to the problem $\min \|r(x)\|$ is zero.
The converse of the theorem follows similarly. □

Therefore, instead of solving the system of nonlinear equations, the equivalent optimization problem can be solved. If the solution to the optimization problem is zero, then the system of equations will have a solution, and the solution is equal to the solution to the optimization problem. Indeed if the solution to the optimization problem is a nonzero number, then the system of equivalent equations will not have a solution. The system of equations may also have more than one solution, in this case, the equivalent optimization problem will have several optimal solutions. Hence, the following optimization problem can be considered equivalent to solving the system $r(x) = 0$:

$$\min \ \frac{1}{2}\|r(x)\|^2. \tag{8}$$

Now, considering the system of flow equations as $r(Q) = 0$, according to (8), we have

$$\min_{Q \in \mathbb{R}^n} \frac{1}{2}\|r(Q)\|^2. \tag{9}$$

Problem (9) can also be written according to the flow equations as follows:

$$\min_{Q \in \mathbb{R}^n} \frac{1}{2}\left[\sum_{j=1}^{NJ}\left(\sum_{ij \in IJ_j} Q_{ij} + q_j\right)^2 + \sum_{L=1}^{NL}\left(\sum_{ij \in IJ_L} R_{ij} Q_{ij}^n\right)^2\right], \tag{10}$$

where R_{ij} is the constant of ijth pipe resistance constant and IJ_j and IJ_L, respectively, represent the pipes connected to node j and the pipes in the L ring. Also, NJ and NL are equal to the number of nodes and network pipes, respectively. Therefore, model (10) is an unconstrained and nonlinear optimization problem. This model can be solved by different optimization methods. Here, we use the trust-region method to solve (10). In what follows, we describe this method.

3.2 Trust-Region Method

Algorithms solving optimization problems usually start from an initial solution and then improve the current point in each iteration. For this reason, these algorithms are also known as iterative algorithms. The strategy of transition from one iteration to another is a factor that distinguishes iterative algorithms. In general, iterative methods are divided into two main categories [38]:

- Line search methods,
- Trust-region methods.

In the line search methods, first, the direction of movement is determined and then the length of the step is decided. Indeed in the methods of the trust-region, first, the length of the movement step is determined and then the direction is decided according to the selected step length. We explain the trust-region method for solving the optimization problem (9) assuming $f(Q) = \frac{1}{2}\|r(Q)\|^2$ as follows.

Suppose that Q_k is the flow rate in the kth iteration. In iterative methods for solving optimization problems, Q_{k+1} is updated as follows:

$$Q_{k+1} = Q_k + p_k, \tag{11}$$

where the vector p_k is selected in such a way that the maximum improvement for the problem objective function occurs. In each iteration of the trust-region method, for finding p_k, by using the Taylor series, an approximation of the objective function is obtained as follows:

$$f(Q_k + p) = f_k + g_k^T p + \frac{1}{2} p^T \nabla^2 f(Q_k + tp)p, \tag{12}$$

where $f_k = f(Q_k)$, $g_k = \nabla f(Q_k)$, and t is a number in the range $(0,1)$. The Jacobian matrix (J) can be used as a suitable approximation instead of ∇f and $\nabla^2 f$ [38]. By replacing $\nabla f = J_k^T r_k$ and $\nabla^2 f = J_k^T J_k$, a suitable approximation is obtained in each iteration of the objective function of the problem. Also,

$$m_k(p) = f_k + p^T J_k^T r_k + \frac{1}{2} p^T J_k^T J_k p = \frac{1}{2}\|r_k + J_k p\|^2, \tag{13}$$

where $m_k(p)$ is an approximation of the function $f(Q)$ around the point Q_k. The difference between the approximate function $m_k(p)$ and the function $f(Q_k + p)$ is equal to $O(\|p\|^2)$. If the value of p is small, then the difference between the two functions will be small. Therefore, in each iteration of the trust-region method, to find the suitable direction, the following optimization problem must be solved:

$$\min_{p \in \mathbb{R}^n} \; m_k(p) = f_k + p^T J_k^T r_k + \frac{1}{2} p^T J_k^T J_k p \quad \text{s.t. } \|p\| \leq \Delta_k, \tag{14}$$

where Δ_k is the radius of the trust-region in the kth iteration. The value of Δ_k should be chosen such that $m_k(p)$ and $f(Q)$ are approximately equal in this region. Moreover,

p_k is the solution of (14). In fact, p_k is the direction in which the most reduction for $m_k(p)$ occurs. As mentioned, if p is small, then the value of $m_k(p)$ and $f(Q)$ will be close to each other. If small Δ_k with $\|p\| \leq \Delta_k$ is selected, then the two functions $m_k(p)$ and $f(Q)$ in this region behave similarly. Hence, for $f(Q)$, the largest possible reduction occurs by moving from the point Q_k in the direction of p_k. Therefore, in each iteration of the trust-region method, instead of solving the main problem, problem (14) will be solved.

As mentioned, choosing the radius of the trust region is important. If the performance of the algorithm is good, then the radius of the region must be increased in order to have a better speed of convergence. If the algorithm performance is poor, then the trust region decreases for greater accuracy. The performance of the algorithm in each iteration is determined by the following formula:

$$\rho_k = \frac{f(Q_k) - f(Q_k + p_k)}{m_k(0) - m_k(p_k)} = \frac{\|r(Q_k)\|^2 - \|r(Q_k + p_k)\|^2}{\|r(Q_k)\|^2 - \|r(Q_k) + J(Q_k)p_k\|^2}. \tag{15}$$

If $\rho_k = 1$, then the approximate function m_k and f will be closer to each other in the existing area. In other words, if the value is closer to one, then the algorithm has a better performance. The main steps of trust-region can be summarized by a pseudo code as Algorithm 2 below.

Algorithm 2 Trust-region algorithm

```
Input: $\hat{\Delta} > 0$, $\Delta_0 \in (0, \hat{\Delta})$, and $\eta \in [0, \frac{1}{4})$.

1: For $k = 0, 1, 2, \ldots$, do the following operations:
    1-1 : Obtain the value of $p_k$ by solving (14).
    1-2 : Calculate the value of $\rho_k$ using relation (15).
    1-3 : Find $\Delta_{k+1}$, using $\rho_k$ by
        1-3-1: If $\rho_k < \frac{1}{4}$, then $\Delta_{k+1} = \frac{1}{4}\Delta_k$.
        1-3-2: If $\rho_k > \frac{3}{4}$ and $\|p_k\| = \Delta_k$, then $\Delta_{k+1} = \min(2\Delta_k, \hat{\Delta})$; else $\Delta_{k+1} = \Delta_k$.
    1-4: Find $Q_{k+1}$, using $\rho_k$ and $\eta$ by
        1-4-1: If $\rho_k > \eta$, then $Q_{k+1} = Q_k + p_k$; else $Q_{k+1} = Q_k$.

Output: $Q_k$ and $f(Q_k)$.
```

The details of the steps associated with Algorithm 2 are described next.
The maximum trust-region radius, the trust-region radius for the first iteration, and η should be given as input to the algorithm. The parameters of our proposed algorithm are set by IRACE PACKAGE [31] to ensure fair space, $\hat{\Delta} = 0.9$, $\Delta_0 = 0.9$ and $\eta = 0.2$.

In step 1-2, the optimal solution is obtained through an iterative process. For this purpose, in each iteration in step 1-1, a quadratic approximation (14) of the original objective function $(\frac{1}{2}\|r(Q)\|^2)$ is calculated based on the solution of the previous iteration.

Then by solving the updated equation (14), the previous iteration solution improves. To solve equation (14), the Dogleg algorithm (The Dogleg algorithm is described below) is used. In step 1-2, based on the solution obtained in step 1-1, the value of ρ_k is calculated. A larger ρ_k (close to 1) indicates that the approximate function of (14) and the original objective function are close to each other.

Based on the calculated ρ_k in step 1-3, the trust-region radius of the next iteration is decided. Also, $\rho_k < \frac{1}{4}$ indicates that the approximation obtained from step 1-1 is not an appropriate approximation for the original objective function. In this case, in order to increase accuracy, the trust-region radius becomes smaller. In addition, $\rho_k > \frac{3}{4}$ indicates that the approximation obtained from step 1-1 is a very good approximation for the original objective function. In this case, in order to increase the speed of convergence, the trust-region radius increases. Moreover, $\frac{1}{4} < \rho_k < \frac{3}{4}$ indicates that the approximation obtained from step 1-1 is a normal approximation for the original objective function. In this case, the trust-region radius does not change.

In step 1-4, a decision is made based on ρ_k whether or not to accept the current iteration solution. Also, $\rho_k < \eta$ indicates that the approximation obtained from step 1-1 is not a suitable approximation for the original objective function. Therefore, accepting the solution obtained from the approximate function may complicate the convergence process of the algorithm. For this reason, in this case, the solution obtained from step 1-1 will not be accepted. In this case, step 1-1 is repeated with the same approximation function as the previous one, except that the trust-region radius is reduced in step 1-3. Hence, it is expected that repeating step 1-1 will lead to a more accurate solution.

The parameters and variables of the proposed method for solving the equations of the WDN are reported in Table 2.

Table 2: Parameters and variables of the proposed method for solving the equations of the WDN

symbol	Type	Expression
R	Parameter	The resistance constant of the pipe;
Q_0	Parameter	The initial flow of pipes (initial solution);
q	Parameter	The flow that exits (or enters) the network at each node;
Δ_0	Parameter	The radius of the trust-region is the first iteration;
$\hat{\Delta}$	Parameter	Maximum radius of the trust-region;
η	Parameter	The value used to reject or confirm the solution to each iteration;
Q	Variable	The flow that passes through the network pipes.

3.3 Dogleg Algorithm

This section describes the Dogleg algorithm. The Dogleg algorithm first removes the quadratic phrase of the objective function (14) ($\frac{1}{2}p^TJ_k^TJ_kp$) and solves the following linear optimization problem:

$$\min_{p\in\mathbb{R}^n} f_k + p^TJ_k^Tr_k \quad \text{s.t. } \|p\| \leq \Delta_k. \tag{16}$$

The solution obtained from the problem (16) is called p_k^s. Since the objective function of problem (16) is linear, it can be solved easily. We know that the value of the objective function always decreases by moving in the direction $-g_k = -J_k^T r_k$. Therefore, the lowest value of the objective function (16) is obtained for $p_k^s = -\alpha J_k^T r_k$, so that the higher α, the lower the value of the objective function (16). Given that in problem (16) $\|p_k^s\| \leq \Delta_k$, α value is equal to $\frac{\Delta_k}{\|J_k^T r_k\|}$. Hence, the solution to problem (16) is obtained from

$$p_k^s = -\frac{\Delta_k}{\|J_k^T r_k\|} J_k^T r_k. \tag{17}$$

Linearizing the objective function of problem (14) reduces the accuracy of the obtained solution. For this reason, after calculating the solution to problem (16), the Dogleg algorithm solves the following quadratic problem in order to increase accuracy:

$$\min_{\tau \geq 0} \; m_k(\tau p_k^s) \quad \text{s.t. } \|\tau p_k^s\| \leq \Delta_k. \tag{18}$$

The solution obtained from the problem (18) is called τ_k. Since the objective function (18) is a univariate quadratic function, solving problem (18) is very simple (Set the differential of the function equal to zero and solve a simple equation). Accordingly, the least value of the objective function (18) occurs for $\tau_k = \frac{\|J_k^T r_k\|^3}{\Delta_k r_k^T J_k (J_k^T J_k) J_k^T r_k}$. According to the constraints of problem (18), it must be $\|\tau_k p_k^s\| \leq \Delta_k$. Since $\|p_k^s\| = \Delta_k$ (according to (17)), it must be $\tau_k \leq 1$. Thus τ_k is obtained from (19).

$$\tau_k = \min\{1, \frac{\|J_k^T r_k\|^3}{\Delta_k r_k^T J_k (J_k^T J_k) J_k^T r_k}\}. \tag{19}$$

So, if $\tau_k < 1$, then $\tau_k p_k^s$ is a better solution to the problem (14) compared to p_k^s. The $\tau_k p_k^s$ is called p_k^c. and it obtains accordingly as follows.

$$p_k^c = -\tau_k (\frac{\Delta_k}{\|J_k^T r_k\|}) J_k^T r_k. \tag{20}$$

If $\|p_k^c\| = \Delta_k$, then the Dogleg algorithm considers $p_k = p_k^c$ as the approximation solution to (14). If $\|p_k^c\| < \Delta_k$, then we provide another direction to calculate the solution to (14) for increasing the convergence speed. To determine this direction, the following problem must be solved unconstrained:

$$\min_{p \in \mathbb{R}^n} \; m_k(p) = f_k + p^T J_k^T r_k + \frac{1}{2} p^T J_k^T J_k p. \tag{21}$$

The solution to the problem (21) is called p_k^j. Problem (21) is an unconstrained quadratic problem. Therefore, to obtain p_k^j, it suffices to set the differential of the objective function to zero ($J_k^T r_k + \frac{1}{2} J_k^T J_k p = 0$). By solving this simple linear equation, we have

$$p_k^j = -(J_k^T J_k)^{-1}(J_k^T r_k) = -J_k^{-1} r_k. \tag{22}$$

For finding the approximation solution to problem (14), the Dogleg algorithm uses the combination of the two directions p_k^c and p_k^j. The main steps of the Dogleg algorithm can be summarized as Algorithm 3 below.

Algorithm 3 Dogleg algorithm

```
    Input: Trust-region radius ($\Delta_k$), Jacobian matrix ($J_k$) and Vector of transactions ($r_k$)
1:  Calculate the value of $p_k^c$ using relation (20).
    1-1: If $\|p_k^c\| = \Delta_k$, then set $p_k = p_k^c$.
    1-2: else, do the following:
      1-2-1: Calculate the value of $p_k^j$ using relation (22).
      1-2-2: Set $p_k = p_k^c + \tau(p_k^j - p_k^c)$.
      1-2-3: Calculate the maximum value of $\tau \in [0,1]$ as $\|p_k\| \leq \Delta_k$.
    Output: Vector $p_k$.
```

3.4 Convergence of the Proposed Method

In this paper, to solve equations of WDN, the optimization problem (10) is solved using the trust-region method. problem (10) is an unconstrained optimization problem. So if $\nabla(f(Q)) = 0$, then Q will be the optimal solution. In [38], it proved that the gradient sequence created in the trust-region method converges to zero (for $\eta > 0$). Hence, to solve the problem (10) the trust-region method is convergent. In addition, the convergence of the trust-region method in the general case has also been proved in [38].

4 Numerical Results

In the hydraulic analysis software of WDN, the use of the gradient method to solve network equations is popular. Therefore, in this section, we compare the performance of the proposed method with the gradient method using several numerical examples. All executions are done on a notebook with characteristics of CPU: intel core i5 2520M 2.5GHz with 8 GB RAM under Windows 7 home premium in MATLAB R2017b software. In the following, the gradient method for solving the equations of WDN is briefly explained. Then, study examples are introduced, and finally, the performance of the two methods of trust-region and gradient are compared in terms of accuracy and speed.

4.1 Gradient Method

In order to hydraulically analyze a WDN, its hydraulic equations must be solved. The gradient method is currently used in many popular commercial software, such as WATERGEMS and EPANET to solve these equations. For solving the equations of the WDN, the gradient method solves a linear equation system in each iteration. This system includes two types of equations. The first type is continuity equations (4) that do not need to be updated in each iteration. The second type is the below equations (23), which must be updated in each iteration according to the solution of the previous iteration.

$$H_{t+1,oi} - H_{t+1,oj} - (nR_{ox}|Q_{t,ox}|^{n-1})Q_{t+1,x} = (1-n)R_{ox}Q^{n}_{t,ox}, \qquad x = 1,\dots,NP, \tag{23}$$

where, $H_{t+1,oi}$ and $H_{t+1,oj}$ represent the head in nodes i and j in the iteration of $t+1$, respectively. Also, R_{ox} indicates the resistance of the pipe and $Q_{t,ox}$ the pipe flow x in the iteration of t. Thus the gradient method in each iteration forms a linear equation system and then solves it. For more information, we refer the reader to [51].

4.2 Examples

In this section, some study examples are introduced.

Example 1. [49] Figure 2 shows a simple WDN. This network has no valve and pump and also has two source nodes and four consumption nodes. The flow equation system of this network has seven equations. This system includes four linear equations and three nonlinear equations.

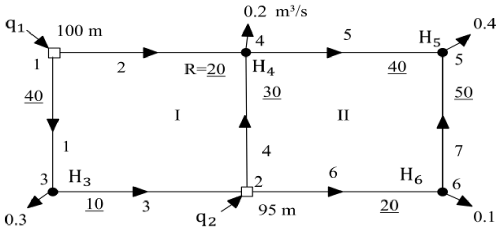


Figure 2: WDN of Example 1 [49].

Example 2. Figure 3 shows a WDN having two source nodes and four consumption nodes. This network consists of four loops. The system of equations related to this network consists of four linear equations and five nonlinear equations. Therefore, in this example, the number of nonlinear equations is more than linear equations. The resistance constant of the pipes of this network is reported in Table 3, and the amount of harvest in the consumption nodes is reported in Table 4.

Table 3: Constant resistance of network pipes of Figure 3

Pipe	R	Pipe	R
1	20	6	10.7365
2	30	7	30
3	40	8	200
4	100	9	200
5	23.53		

Table 4: Water withdrawal from network consumption nodes of Figure 3

Consumption node	2	3	4	5
Harvest rate	0	0.2	1.1590	0.7059

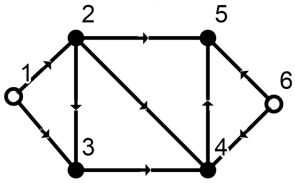


Figure 3: WDN of Example 2.

Example 3. [49] The WDN of Figure 4 consists of a pump and two spring nodes. The flow equations of this network have eleven variables. These equations consist of seven linear equations and four nonlinear equations. The pipe resistance constant of this network is given in Table 5.

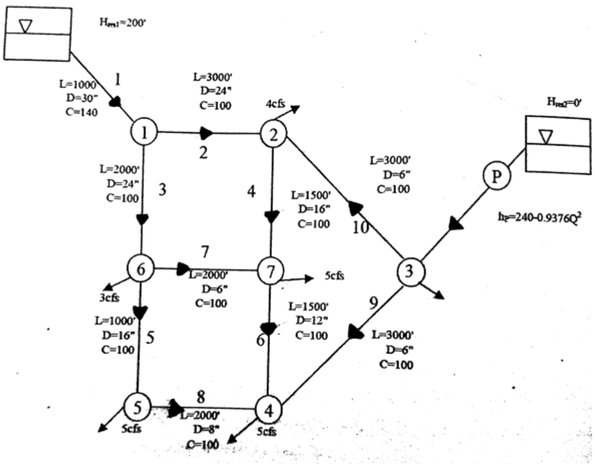


Figure 4: WDN of Example 3 [49].

Table 5: Constant resistance of network pipes of Figure 3

Pipe	R	Pipe	R
1	0.0072	7	68.5175
2	0.1202	8	16.8791
3	0.0801	9	102.7762
4	0.4329	10	102.7762
5	0.2886	11	0.3055
6	1.7573		

Example 4. The WDN of Figure 5 consists of seventeen nodes, twenty pipes, and four loops. The flow equations for this example have twenty variables. The amount of discharge from the nodes of this network, as well as the resistance constant of its pipes, is given in Table 6. The values of nodes 1 and 13 are 300 and 250, respectively.

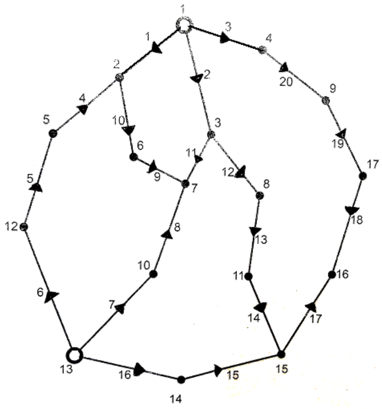


Figure 5: WDN of Example 4.

Table 6: Flow rate picked up and constant resistance of network pipes of Figure 5

Row	Constant pipe resistance R	flow rate picked up
1	40	*
2	10	0.7956
3	60	0.2089
4	60	0.3536
5	10	0.7716
6	10	0.5142
7	80	2.0484
8	120	0.1157
9	15	0.1726
10	12	0.0511
11	240	0.0931
12	80	0.2092
13	120	*
14	120	0.3559
15	10	0.8339
16	10	1.2614
17	20	0.1186
18	120	*
19	120	*
20	120	*

Example 5. The WDN of Figure 6 consists of sixty-three nodes, 110 pipes, and 48 loops. The flow equations for this example have 110 variables. The resistance constant of the pipes related to this network is written on the pipes of Figure 6. The amount of discharge from the nodes of this network is given in Table 7. The head of nodes 1 and 55 are 200 and 100, respectively.

Table 7: Flow rate picked up and constant resistance of network pipes of Figure 6

Row	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
flow rate picked up	-	0.3	0.2	0.1	0.3	0.5	0.3	0.7	0.2	0.5	0.3	0.2	0.1	0.4	0.1	0.3	0.5	0.2	0.5	0.1	0.2
Row	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42
flow rate picked up	0.3	0.6	0.5	0.3	0.2	0.3	0.2	0.5	0.2	0.3	0.4	0.2	0.3	0.2	0.3	0.2	0.3	0.1	0.3	0.3	0.2
Row	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
flow rate picked up	0.1	0.3	0.5	0.3	0.2	0.3	0.2	0.5	0.2	0.7	-	0.3	0.2	0.5	0.4	0.5	0.3	0.5	0.1	0.5	0.6

4.3 Examining the Trust-Region Method

In the following, we compare the trust-region method with the gradient method in terms of convergence speed and accuracy. Table 8 shows the results of the trust-region and gradient methods for Examples 1 to 5. In Table 4 the stopping criterion for both methods is considered $|f(Q_k) - f(Q_{k-1})| < \epsilon$.

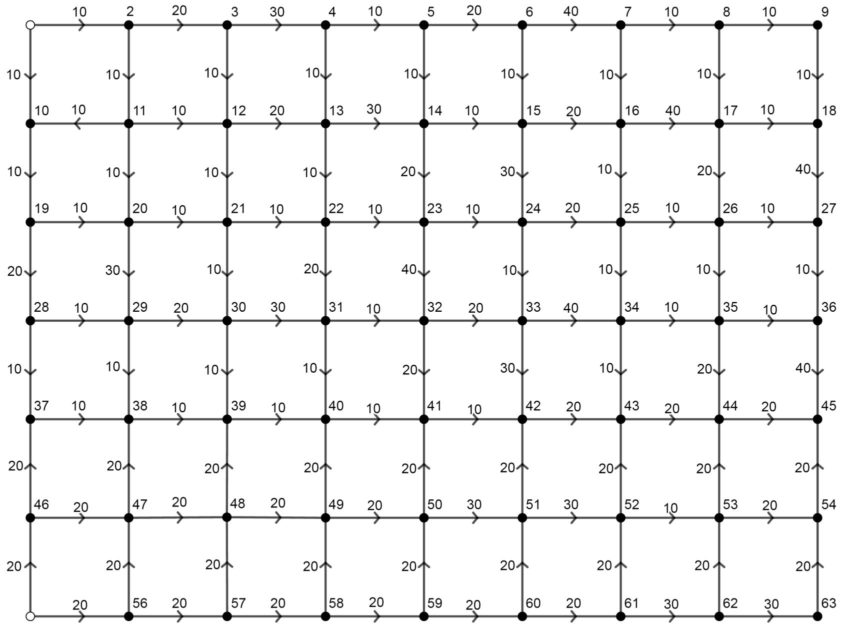


Figure 6: WDN of Example 5.

Table 8: Comparison of trust-region and gradient methods

	Convergence	Example 1	Example 2	Example 3	Example 4	Example 5
Dimension		7	9	11	20	110
Gradient	Time	0.018	0.011	0.020	0.023	-
	Iteration	5	5	5	4	-
	Accuracy	0.0054	$2.52e-05$	60.2242	0.0961	Not converge
Trust-Region	Time	0.052	0.032	0.043	0.045	1.49
	Iteration	8	8	31	6	26
	Accuracy	$2.88e-31$	$1.20e-29$	0.0224	$5.87e-28$	$9.85e-27$

By using the value of the objective function of problem (10), we can conclude that if the value of the objective function of problem (10) is low then the accuracy of the obtained solution is high. Hence table 8 compares the accuracy of the trust-region method and the gradient method based on the objective function value of problem (10). As can be seen, in Examples 1 to 4, the accuracy of the trust-region method is much better than the gradient method. Example 5 is related to a relatively large water distribution network. The gradient method does not achieve convergence in solving the hydraulic equations of this network, but the trust-region method solves the equations of this network with reasonable accuracy and implementation time.

EPANET software is a common software in the hydraulic analysis of WDNs. This software uses the gradient method to solve network equations. For a more applied comparison, the following network (Figure 7) was implemented in EPANET software. Also, the hydraulic equations governing this network were solved by the trust-region method.

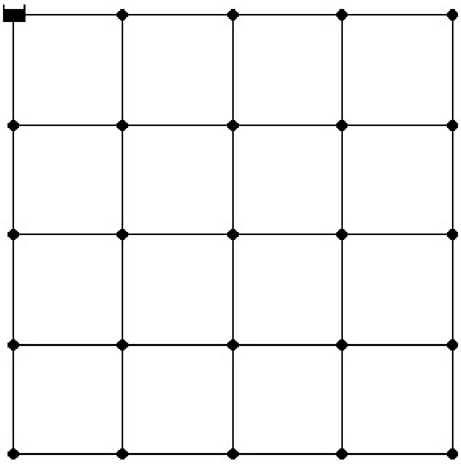


Figure 7: Implement a WDN in the EPANET software.

Table 9 compares the accuracy of the trust-region method with the accuracy of solving equations by EPANET.

Table 9: Comparison of trust region and EPANET software

Method	Solution accuracy	Dimension
EPANET (Gradient)	0.0080	40
Trust-region	0.0010	40

As can be seen, solving the network equations using the trust-region method is more accurate than the solution obtained from the EPANET software.

In general, more convergence and better accuracy are the advantages of the trust-region method compared to the gradient method. Hence, using the trust-region method compared to the gradient method can provide a better hydraulic analysis of a water distribution network.

The gradient method has performed somewhat better in terms of convergence speed. Therefore, changes in the method of trust-region to increase the speed of convergence can be considered for future research.

5 Conclusion

Here, for solving nonlinear hydraulic equations, we proposed a trust-region method. We solved some randomly generated test examples and made a comparative study to show the effectiveness of our proposed algorithm with the gradient method. The results showed that the trust-region method is more accurate than the gradient method, and also, the results show that the gradient method can not be converged when the dimensions of the problem become high, while the trust-region method solved these equations with suitable accuracy. Therefore, using the trust-region method can provide a better hydraulic analysis of a WDN.

Acknowledgments

The first and second authors thank the Research Council of Ferdowsi University of Mashhad and Optimization Laboratory of Ferdowsi University of Mashhad and the third author thanks the Mosaheb Institute of Mathematics, Kharazmi University for supporting this work.

Data Availability Statement

Some or all data, models, or codes that support the findings of this study are available from the corresponding author upon reasonable request.

References

[1] Amat S., Busquier S., Gutiérrez J.M. (2003). “Geometric constructions of iterative functions to solve nonlinear equations”, Journal of Computational and Applied Mathematics, 157, 197-205.

[2] Aragones D.G., Calvoa G.F., Galan A. (2021). “A heuristic algorithm for optimal cost design of gravity-fed water distribution networks. a real case study”, Applied Mathematical Modelling, 95, 379-395.

[3] Ates S. (2017). “Hydraulic modelling of control devices in loop equations of water distribution networks”, Flow Measurement and Instrumentation, 53, 243-260.

[4] Bermúdez J.R., Estrda F.R.L., Besanćon G., Palomo G.V., Torres L., Hernández H.R. (2018). “Modeling and simulation of a hydraulic network for leak diagnosis”, Mathematical and Computational Applications, 23.

[5] Bertsekas D.P. (2014). “Constrained optimization and Lagrange multiplier methods”, Academic Press.

[6] Bhave P.R. (1991). “Analysis of flow in water distribution networks”, Technomic Publishing Co., Inc., Lancaster.

[7] Brkić D. (2011). “Iterative methods for looped network pipeline calculation”, Water Resources Management, 25, 2951-2987.

[8] Brkić D., Hansen P. (2009). “An improvement of hardy cross method applied on looped spatial natural gas distribution networks”, Applied Energy, 86, 1290-1300.

[9] Broyden C.G. (1965). “A class of methods for solving nonlinear simultaneous equations”, Mathematics of Computation, 19, 577-593.

[10] Collins A.G., Johnson R.L. (1975). “Finite-element method for water-distribution networks”, American Water Works Association, 67, 385-389.

[11] Costa C.M., Grapiglia G.N. (2020). “A subspace version of the Wang–Yuan augmented Lagrangian-trust-region method for equality constrained optimization”, Applied Mathematics and Computation, 387.

[12] Cross H. (1936). “Analysis of flow in networks of conduits or conductors”, University of illinois at urbana champaign, College of Engineering.

[13] Djebedjiana B., Abdel-Gawad H. A.A., Ezzeldin R.M. (2021). “Global performance of metaheuristic optimization tools for water distribution networks”, Ain Shams Engineering Journal, 12, 223-239.

[14] Donachie R.P. (1974). “Digital program for water network analysis”, Journal of the Hydraulics Division, 100, 393-403.

[15] Elhay S., Piller O., Deuerlein J., Simpson A. (2016). “A robust, rapidly convergent method that solves the water distribution equations for pressure-dependent models”, Journal of Water Resources Planning and Management, 142(2), 04015047-1.

[16] El-Sobky B. Elnaga Y.A. (2018). “A penalty method with trust-region mechanism for nonlinear bilevel optimization problem”, Journal of Computational and Applied Mathematics, 340, 360-374.

[17] Fan J., Pan J. (2011). “An improved trust-region algorithm for nonlinear equations”, Computational Optimization and Applications, 48, 59-70.

[18] Giustolisi O., Laucelli D. (2011). “Water distribution network pressure-driven analysis using the enhanced global gradient algorithm (EGGA)”, Journal of Water Resources Planning and Management, 137, 498-510.

[19] Grosan C., Abraham A. (2008). “A new approach for solving nonlinear equations systems”, IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, 38, 698-714.

[20] Hamlehdar M., Yousef H., Noorollahi Y., Mohammadi M. (2022). “Energy recovery from water distribution networks using micro hydropower: A case study in Iran”, Energy, 252, 124024.

[21] Huzsvár T., Wéber R., Déllei Á., Hös C. (2021). “Increasing the capacity of water distribution networks using fitness function transformation”, Water Research, 201, 117362.

[22] Isaacs L.T., Mills K.G. (1980). “Linear theory methods for pipe network analysis”, Journal of the Hydraulics Division, 106, 1191-1201.

[23] Jeppson R.W. (1976). “Analysis of flow in pipe networks”, Butterworth-Heinemann.

[24] Jerez D.J., Jensen H.A., Beer M., Broggi M. (2021). “Contaminant source identification in water distribution networks: A Bayesian framework”, Mechanical Systems and Signal Processing, 159.

[25] Kelley C.T. (2003). “Solving nonlinear equations with Newton’s method”, Society for Industrial and Applied Mathematics.

[26] Koşucu M.M., Albay E., Demirel M.C. (2022). “Extending EPANET hydraulic solver capacity with rigid water column global gradient algorithm”, Journal of Hydro-Environment Research, 42, 31-43.

[27] Kumar S.M., Narasimhan S., Bhallamudi S.M. (2010). “Parameter estimation in water distribution networks”, Water Resources Management, 24, 1251-1272.

[28] Lam C.F., Wolla M. (1972). “Computer analysis of water distribution systems: Part II–numerical solution”, Journal of the Hydraulics Division, 98, 447-460.

[29] Lemieux P.F. (1972). “Efficient algorithm for distribution networks”, Journal of the Hydraulics Division, 98, 1911-1920.

[30] Liu K. (1969). “The numerical analysis of water supply networks by digital computers, in: Thirteenth congress of the international association for hydraulic Research”, Applied Soft Computation, 1, 36-43.

[31] López-Ibáñez M., Dubois-Lacoste J., Pérez Cáceres L., Birattari M., Stützle T. (2016). “The irace package: Iterated racing for automatic algorithm configuration”, Operations Research Perspectives, 3, 43-58.

[32] Mabrok M.A., Saadb A., Ahmed T., Alsayab H. (2022). “Modeling and simulations of water network distribution to assess water quality: Kuwait as a case study”, Alexandria Engineering Journal, 61, 11859-11877.

[33] Mankad J., Natarajan B., Srinivasan B. (2022). “Integrated approach for optimal sensor placement and state estimation: A case study on water distribution networks”, ISA Transactions, 123, 272-285.

[34] Martin D., Peters G. (1963). “The application of newton’s method to network analysis by digital computer”, Journal of the Institute of Water Engineering, 17.

[35] Mohades M.M., Kahaei M.H., Mohades H. (2021). “Haplotype assembly using Riemannian trust-region method”, Digital Signal Processing, 112, 102999.

[36] Moosavian N., Jaefarzadeh M. (2014). “Multistage linearization method for hydraulic analysis of water distribution network”, Journal of Computational Methods in Engineering, 32, 173-187.

[37] Nielsen H.B. (1989). “Methods for analyzing pipe networks”, Journal of Hydraulic Engineering, 115, 139-157.

[38] Nocedal J., Stephen S.J. (2006). “Numerical optimization”, Springer.

[39] Nogueira A.C. (1993). “Steady-state fluid network analysis”, Journal of Hydraulic Engineering, 119, 431-436.

[40] Ormsbee L.E. (2008). “The history of water distribution network analysis: The computer age”, 8th Annual Water Distribution Systems Analysis Symposium, 1-6.

[41] Petkovic M., Neta B., Petkovic L., Dzunic J. (2012). “A hybrid method for non-linear equations”, Elsevier,

[42] Powell M.J.D. (1970). “A hybrid method for non-linear equations”, Numerical Methods for Nonlinear Algebraic Equations, 14, 87-114.

[43] Powell M.J.D. (1978). “Algorithms for nonlinear constraints that use Lagrangian functions”, Mathematical Programming, 14, 224-248.

[44] Rao H., Bree D.W. (1977). “Extended period simulation of water systems–Part A”, Journal of the Hydraulics Division, 103, 97-108.

[45] Shamir U.Y., Howard C.D. (1968). “Water distribution systems analysis”, Journal of the Hydraulics Division, 94, 219-234. .

[46] Sheng Z., Luo D. (2020). "A Cauchy point direction trust-region algorithm for nonlinear equations", Mathematical Problems in Engineering, 2020.

[47] Simpson A., Elhay S. (2011). "Jacobian matrix for solving water distribution system equations with the Darcy–Weisbach head-loss model", Journal of Hydraulic Engineering, 137, 696-700.

[48] Swamee P.K., Sharma A.K. (2008). "Design of water supply pipe networks", John Wiley & Sons.

[49] Tabesh M. (1998). "Implications of the pressure dependency of outflows of data management, mathematical modelling and reliability assessment of water distribution systems", PhD Thesis, University of Liverpool.

[50] Tanyimboh T., Tahar B., Templeman A. (2003). "Pressure-driven modelling of water distribution systems", Water Science and Technology: Water Supply, 3, 255-261.

[51] Todini E., Pilati S. (1988). "A gradient algorithm for the analysis of pipe networks", Computer Applications in Water Supply, 1-20.

[52] Walski T. (2018). "Water distribution system analysis before the digital age", WDSA/CCWI Joint Conference Proceedings, 1.

[53] Wood D.J., Charles C.O. (1972). "Hydraulic network analysis using linear theory", Journal of the Hydraulics Division, 98, 1157-1170.

[54] Yang P., Jiang Y.L., Xu K.L. (2019). "A trust-region method for H2 model reduction of bilinear systems on the Stiefel manifold", Journal of the Franklin Institute, 356, 2258-2273.

[55] Yuan Y. (2015). "Recent advances in trust-region algorithms", Mathematical Programming, 151, 249-281.

[56] Zarghamee M.S. (1971). "Mathematical model for water distribution systems", Journal of the Hydraulics Division, 97, 1-14.

How to Cite this Article:

Ahmadnia, M., Ghanbari, R., Ghorbani-Moghadam, Kh. (2022). "Solving nonlinear hydraulic equations of water distribution networks by using a trust-region method", Control and Optimization in Applied Mathematics, 7(2): 53-75. doi: 10.30473/coam.2022.62968.1195

Payame Noor University

Control and Optimization in Applied Mathematics (COAM)
Vol. 7, No. 2, Summer-Autumn 2022 (77-98), ©2016 Payame Noor University, Iran

DOI. 10.30473/coam.2022.62261.1207 (Cited this article)

Research Article

Using the Integral Operational Matrix of B-Spline Functions to Solve Fractional Optimal Control Problems

Yousef Edrisi-Tabriz*

Department of Mathematics, Payame Noor University (PNU), P.O. BOX 19395-4697, Tehran, Iran.

Received: June 11, 2022 ; **Accepted:** November 17, 2022 .

Abstract. In this paper, we present a numerical method for solving the fractional optimal control problems in which fractional integral operational matrices of basic B-spline functions are used. In the proposed method, we use the Riemann-Liouville fractional integral. With the help of the operational matrix of the fractional integral and the collocation method, we transform the fractional optimal control problem into a nonlinear programming problem and then solve it with an appropriate optimization algorithm. Compared to similar numerical techniques, our method has better accuracy and efficiency, and also it is easy to use. To provide a clear view of the applicability and efficiency of our numerical method, several illustrative examples are presented.

Keywords. Fractional optimal control problem, Riemann-Leiville fractional integral, Operational matrix, B-Spline function, Collocation method.

MSC. 49N10; 65D07; 65R10; 65L60.

* Corresponding author
yousef_edrisi@pnu.ac.ir

1 Introduction

Currently, one of the most widely used parts of applied mathematics belongs to fractional calculus. In recent years, this field has become an emerging position for science and engineering researchers with a wide range of applications. The range of applications of fractional calculus is increasing rapidly, including in pharmacokinetics [32], hyperchaotic system [30], radar-guided missile [36], quantum mechanics [20], stochastic programming [7], control theory [23], and image processing [1]. Numerical methods for solving fractional optimal control problems (FOCPs), on the other hand, have received much attention in recent years due to their ease of use and flexibility. Increasing the accuracy of these methods improves the results in practical applications, so the development of more accurate methods is of interest to researchers. Direct and indirect methods are the two main approaches in solving optimal control problems (OCPs) and more recently FOCPs [26]. In the current paper, we use a direct method to solve such problems. To use the direct method, a basic polynomial is needed to discretize FOCP. Various methods are developed with different polynomials such as Legendre [21], Jacobi [9], Bernstein [24], Boubaker [25] and Taylor polynomials [39]. Some other works in solving FOCPs that have been done recently and are of high accuracy are [3, 4, 15, 31, 37, 38]. Here, we use linear B-spline functions as basic polynomials [17]. Spline and B-spline polynomials were first introduced by Schoenberg in 1946 in his landmark paper. In this article, he states the theoretical foundations for this issue [28, 29, 34]. Due to the desirable properties of polynomial splines, they play a significant role in numerical analysis and approximation theory. Lakestani et al. [18] constructed the operational matrix of fractional derivatives using B-spline functions and solved fractional differential equations with the help of this matrix. This matrix was then used to solve various problems, including the problem of OCP in [11].

In numerical methods, sometimes an operational matrix of derivation [10, 11], and sometimes an operational matrix of integration [6, 12], is used. We choose the operational matrix for Riemann-Liouville integration. We represent this matrix with the equation.

$$I^{\alpha}\Phi_M(t) \approx \mathcal{I}^{\alpha}\Phi_M(t),$$

where I^{α} is the Riemann-Liouville integral operator of order α, $\mathcal{I}^{\alpha}$ is the operational matrix of fractional integration and the elements of $\Phi_M(t)$ are B-spline basis functions. We utilize this matrix to transform FOCPs into a nonlinear programming one and then solve it by suitable algorithms. In this paper, the operational matrix of the Riemann-Liouville fractional integral of B-spline functions are rewritten with the help of Laplace transforms, then using this matrix and in the form of a new numerical method, the fractional optimal control problem is solved. The results of the new numerical method are compared with the results of the numerical methods described in [6, 13, 15, 21, 35].

The paper is organized as follows. First of all, in Section 2, some preliminaries of fractional calculus and some necessary definitions of linear B-spline functions are briefly reviewed. Details on the construction of the operational matrix of fractional integration are reported in Section 3. The structure of the fractional optimal control problems is stated in Section 4. The new numerical method is presented in Section 5. In Section 6, the convergence of the proposed method is considered. In Section 7, we apply our numerical method to solve four examples. Finally, Section 8 completes this paper with a brief conclusion.

2 Introductory Definitions

2.1 The Caputo Fractional Derivative and the Riemann-Liouville Integral Operator

Definition 1. The Caputo fractional-order derivative is defined by [8]

$$D^{\alpha}\mathbf{x}(t)=\frac{1}{\Gamma(n-\alpha)}\int_0^t\frac{\mathbf{x}^{(n)}(\tau)}{(t-\tau)^{\alpha+1-n}}\,\mathrm{d}\tau,\ \ n-1<\alpha\leq n,\ \ n\in\mathbb{N}, \tag{1}$$

where $\alpha>0$ is the order of the derivative and n is the smallest integer not less than α.

Definition 2. The Riemann-Liouville fractional integral operator of order α is defined by [8]

$$I^{\alpha}\mathbf{x}(\mathbf{t})=\begin{cases}\dfrac{1}{\Gamma(\alpha)}\displaystyle\int_0^t\frac{x(\tau)}{(t-\tau)^{1-\alpha}}\,\mathrm{d}\tau=\dfrac{1}{\Gamma(\alpha)}t^{q-1}*\mathbf{x}(t), & \alpha>0,\\ \mathbf{x}(t), & \alpha=0,\end{cases} \tag{2}$$

where $*$ indicates the convolution product.

The relationship between the Caputo derivative and Riemann-Liouville integral is given in the following equation [8]

$$I^{\alpha}(D^{\alpha}\mathbf{y}(t))=\mathbf{y}(t)-\sum_{k=0}^{n-1}\frac{t^k}{k!}\mathbf{y}^{(k)}(0), \tag{3}$$

where $n-1<\alpha\leqslant n$ and $\mathbf{y}^{(k)}(0)$ are the k-th order derivative of $\mathbf{y}(t)$ at $t=0$.

2.2 Linear B-Spline Functions

A spline function of order n complies with a piecewise polynomial function of degree $n-1$. In these functions, knots are the junction of the pieces. The B-spline is short for base spline, first introduced by Isaac Jacob Schoenberg. These basic functions are semi-orthogonal and have unique features that distinguish them for use in approximating functions. One of the most important features of B-spline functions is the continuity of themselves and their derivatives. An arbitrary function can be approximated by a linear combination of B-spline functions [14]. Linear B-spline functions (the second order) are as follows

$$\phi_{i,k}(t)=\begin{cases}t_i-k, & k\leq t_i<k+1,\\ 2-(t_i-k), & k+1\leq t_i<k+2, k=0,\dots,2^i-2,\\ 0, & \text{otherwise},\end{cases} \tag{4}$$

with left-hand side boundary functions

$$\phi_{i,k}(t)=\begin{cases}2-(t_i-k), & 0\le t_i<1, k=-1,\\ 0, & \text{otherwise},\end{cases} \tag{5}$$

and with right-hand side boundary functions

$$\phi_{i,k}(t)=\begin{cases}t_i-k, & k\le t_i<k+1, k=2^i-1,\\ 0, & \text{otherwise.}\end{cases} \tag{6}$$

The relation between t and t_i is $t_i=2^it$ [17].

2.3 Approximation by B-Spline Functions

To approximate an arbitrary function $f(t)\in L^2[0,1]$ through the B-spline functions, first let $i=M$ and then assume [18]

$$f(t)\simeq\sum_{k=-1}^{2^M-1}a_k\phi_{M,k}(t)=A^T\Phi_M(t), \tag{7}$$

where

$$\Phi_M=[\phi_{M,-1}(t),\phi_{M,0}(t),\dots,\phi_{M,2^M-1}(t)]^T, \tag{8}$$

is a (2^M+1)-vector of the basis function similar to (4), (5) and (6) as follows

$$\phi_{M,-1}(t)=\begin{cases}2-(2^Mt+1), & 0\le t<\frac{1}{2^M},\\ 0, & \text{otherwise},\end{cases} \tag{9}$$

$$\phi_{M,k}(t)=\begin{cases}2^Mt-k, & \frac{k}{2^M}\le t<\frac{k+1}{2^M},\\ 2-(2^Mt-k), & \frac{k+1}{2^M}\le t<\frac{k+2}{2^M}, \quad k=0,\dots,2^M-2,\\ 0, & \text{otherwise},\end{cases} \tag{10}$$

$$\phi_{M,2^M-1}(t)=\begin{cases}2^Mt-(2^M-1), & \frac{2^M-1}{2^M}<t\le 1,\\ 0, & \text{otherwise},\end{cases} \tag{11}$$

and

$$A=[a_{-1},a_0,\dots,a_{2^M-1}]^T, \tag{12}$$

with

$$a_k=f(t_k),\quad t_k=\frac{k+1}{2^M},\quad k=-1,\dots,2^M-1, \tag{13}$$

where the points t_k are the collocation points [16, 17].

3 The Operational Matrix of Fractional Integration

This operational matrix was first obtained in [27] that we rewrite with some little changes. It is easy to see that the linear B-spline functions (9)-(11) can be written by

$$\phi_{M,-1}(t) = \Big(1-2^M t\Big)\Big(\mu_0(t)-\mu_{\frac{1}{2^M}}(t)\Big), \tag{14}$$

$$\phi_{M,k}(t) = \Big(2^M t-k\Big)\Big(\mu_{\frac{k}{2^M}}(t)-\mu_{\frac{k+1}{2^M}}(t)\Big) \tag{15}$$

$$+\Big(2-2^M t+k\Big)\Big(\mu_{\frac{k+1}{2^M}}(t)-\mu_{\frac{k+2}{2^M}}(t)\Big), \quad k=0,\ldots,2^M-2, \tag{16}$$

$$\phi_{M,2^M-1}(t) = \Big(2^M(t-1)+1\Big)\Big(\mu_{\frac{2^M-1}{2^M}}(t)-\mu_1(t)\Big), \tag{17}$$

where $\mu_a(t)$ is the unit step function defined by

$$\mu_a(t) = \begin{cases} 1, & t \geq a, \\ 0, & t < a. \end{cases}$$

By taking the Laplace transform from Equations (14)-(17) we get

$$\mathscr{L}\{\phi_{M,-1}(t)\} = \frac{1}{s}\left(1+\frac{2^M}{s}\left(e^{-\frac{s}{2^M}-1}\right)\right), \tag{18}$$

$$\mathscr{L}\{\phi_{M,k}(t)\} = \frac{2^M}{s^2}\left(e^{-\frac{ks}{2^M}}-2e^{-\frac{(k+1)s}{2^M}}+e^{-\frac{(k+2)s}{2^M}}\right), \qquad k=0,1,\ldots,2^M-2, \tag{19}$$

$$\mathscr{L}\{\phi_{M,2^M-1}(t)\} = \frac{2^M}{s^2}\left(e^{-\frac{(2^M-1)s}{2^M}}-e^{-s}\right)-\frac{e^{-s}}{s}. \tag{20}$$

According to Equation (2), the fractional integration of linear *B*-spline functions $\phi_{M,k}(t)$ of order α is

$$I^\alpha\phi_{M,k}(t) = \frac{1}{\Gamma(\alpha)}\Big(t^{\alpha-1} * \phi_{M,k}(t)\Big),$$

therefore, we have

$$\mathscr{L}\{I^\alpha\phi_{M,k}(t)\} = \frac{1}{s^\alpha}\mathscr{L}\{\phi_{M,k}(t)\}. \tag{21}$$

From Equations (18)-(20) and Equation (23), we get

$$\mathscr{L}\{I^\alpha\phi_{M,k}(t)\} = \frac{2^M}{s^{\alpha+2}}\begin{cases} \Big(\frac{s}{2^M}-1\Big)+e^{-\frac{s}{2^M}}, & k=-1, \\ e^{-\frac{ks}{2^M}}-2e^{-\frac{(k+1)s}{2^M}}+e^{-\frac{(k+2)s}{2^M}} & k=0,1,\ldots,2^M-2, \\ e^{-\frac{(2^M-1)s}{2^M}}-\Big(\frac{s}{2^M}+1\Big)e^{-s}, & k=2^M-1. \end{cases} \tag{22}$$

Taking the inverse Laplace transform of Equation (22), we get

$$I^\alpha\phi_{M,k}(t) = \frac{2^M}{\Gamma(\alpha+2)}$$

$$
\begin{cases}
\left(t-\frac{1}{2^M}\right)^{\alpha+1}\mu_{\frac{1}{2^M}}(t)-\left(t-\frac{\alpha+1}{2^M}\right)t^{\alpha}, & k=-1,\\[2ex]
\left(t-\frac{k}{2^M}\right)^{\alpha+1}\mu_{\frac{k}{2^M}}(t)-2\left(t-\frac{k+1}{2^M}\right)^{\alpha+1}\mu_{\frac{k+1}{2^M}}(t) & \\
+\left(t-\frac{k+2}{2^M}\right)^{\alpha+1}\mu_{\frac{k+2}{2^M}}(t), & k=0,1,\dots,2^M-2,\\[2ex]
\left(t-\frac{2^M-1}{2^M}\right)^{\alpha+1}\mu_{\frac{2^M-1}{2^M}}(t) & \\
-\left(t-1+\frac{\alpha+1}{2^M}\right)(t-1)^{\alpha}\mu_1(t), & k=2^M-1.
\end{cases}
\tag{23}
$$

According to Equation (7), we expand $I^{\alpha}\Phi_{M,k}(t)$ by the linear *B*-spline functions as

$$
I^{\alpha}\phi_{M,k}(t)\cong\sum_{i=-1}^{2^M-1}s_{ki}\phi_{M,k}(t)=\mathbf{S}_k^T\Phi_M(t),
\tag{24}
$$

where

$$
s_{ki}=I^{\alpha}\phi_{M,k}\left(\frac{i+1}{2^M}\right)\qquad i,k=-1,\dots,2^M-1,
\tag{25}
$$

S_k is a (2^M+1)-vector and Φ_M is the basis vector in Equation (8). Therefore, the operational matrix of fractional integration is obtained as follows

$$
I^{\alpha}\Phi_M(t)\cong\mathcal{I}_{\alpha}\Phi_M(t).
\tag{26}
$$

Using Equations (23)-(25), it is easy to see that $\mathcal{I}_{\alpha}$ is a $(2^M+1)\times(2^M+1)$ matrix given by

$$
\mathcal{I}_{\alpha}=
\begin{bmatrix}
0 & \eta_0 & \eta_1 & \eta_2 & \cdots & \eta_{2^M-1}\\
 & \kappa & \nu_1 & \nu_2 & \cdots & \nu_{2^M-1}\\
 & & \kappa & \nu_1 & \cdots & \nu_{2^M-2}\\
 & & & \ddots & \ddots & \vdots\\
 & & & & \kappa & \nu_1\\
 & & & & & \kappa
\end{bmatrix},
\tag{27}
$$

where $\kappa=\dfrac{1}{2^{M\alpha}\Gamma(\alpha+2)}$,

$$
\eta_i=\kappa\Big[(\alpha-i)(i+1)^{\alpha}+i^{\alpha+1}\Big],\quad i=0,1,\dots,2^M-1,
$$

and

$$
\nu_i=\kappa\Big[(i-1)^{\alpha+1}-2i^{\alpha+1}+(i+1)^{\alpha+1}\Big],\quad i=1,2,\dots,2^M-1.
$$

4 Problem Statement

This work aims to propose a new numerical method for approximating the solution of the following FOCP:

$$\text{Min(Max)}\ J(\mathbf{x},\mathbf{u}) = \int_0^1 \mathbf{L}(\mathbf{x}(t),\mathbf{u}(t),t)\,\mathrm{d}t, \tag{28}$$

$$s.t: \quad D^{\alpha}\mathbf{x}(t) = \mathbf{f}(\mathbf{x}(t),\mathbf{u}(t),t), \tag{29}$$

$$\mathbf{x}^{(k)}(0) = x_k, \qquad k = 0,1,\dots,\lfloor\alpha\rfloor, \tag{30}$$

$$g_j(\mathbf{x}(t), D^{\alpha}\mathbf{x}(t),\mathbf{u}(t),t) \le 0, \quad j = 1,2,\dots,w, \tag{31}$$

where $D^{\alpha} = [D^{\alpha_1}, D^{\alpha_2},\dots,D^{\alpha_l}]$ is the fractional derivative operator with

$$n_i - 1 < \alpha_i \le n_i, \quad n_i \in \mathbb{N}, \quad i = 1,2,\dots,l,$$

and

$$\begin{aligned}\mathbf{x}(t) &= [x_1(t), x_2(t),\dots,x_l(t)]^T,\\ \mathbf{u}(t) &= \left[u_1(t),u_2(t),\dots,u_q(t)\right]^T,\\ \mathbf{f} &= [f_1,f_2,\dots,f_l].\end{aligned}$$

Also, $\mathbf{L}$, f_i, and g_j, $i = 1,2,\dots,l$, $j = 1,2,\dots,w$ are linear or nonlinear functions. In addition, it should be noted that the elements of Equation (29) can be written as

$$D^{\alpha_i}x_i(t) = f_i(\mathbf{x}(t),\mathbf{u}(t),t), \qquad i = 1,2,\dots,l. \tag{32}$$

5 The Proposed Numerical Method

In this section, we use the linear B-spline functions to solve FOCP as given in Equations (28)-(31). We expand $D^{\alpha_i}x_i(t)$ in Equation (32) by the linear B-spline functions as

$$D^{\alpha_i}x_i(t) \simeq \mathbf{Y}_i^T\Phi_M(t). \tag{33}$$

By using Equations (3), (26) and (33), we have

$$x_i(t) \simeq \mathbf{Y}_i^T\mathcal{I}_{\alpha_i}\Phi_M(t) + \sum_{k=0}^{n_i-1}\frac{t^k}{k!}x_i^{(k)}(0), \tag{34}$$

where $n_i - 1 < \alpha_i \leqslant n_i$. The expansion of the second term on the right-hand side of Equation (34) by the linear B-spline functions yields

$$x_i(t) \simeq \mathbf{Y}_i^T\mathcal{I}_{\alpha_i}\Phi_M(t) + \mathbf{A}_i^T\Phi_M(t) = \left(\mathbf{Y}_i^T\mathcal{I}_{\alpha_i} + \mathbf{A}_i^T\right)\Phi_M(t), \tag{35}$$

and by setting $\mathbf{X}_i^T = \mathbf{Y}_i^T\mathcal{I}_{\alpha_i} + \mathbf{A}_i^T$, we get

$$x_i(t) = \mathbf{X}_i^T\Phi_M(t). \tag{36}$$

For the control variables, we obtain

$$u_j(t) \simeq \mathbf{U}_j^T \Phi_M(t). \tag{37}$$

Let

$$\mathcal{I}_\alpha = \left[\mathcal{I}_{\alpha_1}, \mathcal{I}_{\alpha_2}, \ldots, \mathcal{I}_{\alpha_l}\right],$$

and

$$\widehat{\Phi}_{M,l}(t) = I_l \otimes \Phi_M(t), \tag{38}$$

$$\widehat{\mathcal{I}}_\alpha = I_l \otimes \mathcal{I}_\alpha \tag{39}$$

$$\widehat{\Phi}_{M,q}(t) = I_q \otimes \Phi_M(t), \tag{40}$$

where I_l and I_q are identity matrices of order l and q respectively and $\otimes$ is the Kronecker product [19]. Now, by using Equations (38) and (40), we have

$$\mathbf{x}(t) \simeq \mathbf{X}^T \hat{\Phi}_{M,l}(t), \tag{41}$$

$$D^\alpha \mathbf{x}(t) \simeq \mathbf{Y}^T \hat{\Phi}_{M,l}(t), \tag{42}$$

$$\mathbf{u}(t) \simeq \mathbf{U}^T \hat{\Phi}_{M,q}(t), \tag{43}$$

where $\mathbf{X}$, $\mathbf{Y}$ and $\mathbf{A}$ are vectors of order $l(2^M+1)\times 1$, and $\mathbf{U}$ is a vector of order $q(2^M+1)\times 1$, given by

$$\mathbf{X} = \left[\mathbf{X}_1^T, \mathbf{X}_2^T, \ldots, \mathbf{X}_l^T\right]^T,$$
$$\mathbf{Y} = \left[\mathbf{Y}_1^T, \mathbf{Y}_2^T, \ldots, \mathbf{Y}_l^T\right]^T,$$
$$\mathbf{A} = \left[\mathbf{A}_1^T, \mathbf{A}_2^T, \ldots, \mathbf{A}_l^T\right]^T,$$
$$\mathbf{U} = \left[\mathbf{U}_1^T, \mathbf{U}_2^T, \ldots, \mathbf{U}_q^T\right]^T.$$

Moreover, by using Equation (39), we obtain $\mathbf{X} = \mathbf{Y}\widehat{\mathcal{I}}_\alpha + \mathbf{A}$. To approximate the objective function, we have two approaches, one related to when $\mathbf{L}(\mathbf{x}(t), \mathbf{u}(t), t)$ in (28) is quadratic as

$$\mathbf{L}(\mathbf{x}(t), \mathbf{u}(t), t) = \xi^T(t)\mathbf{Q}\xi(t) + \mathbf{u}^T(t)\mathbf{R}\mathbf{u}(t)$$

and we have

$$J(\mathbf{x}, \mathbf{u}) = \int_0^1 \left(\xi^T(t)\mathbf{Q}\xi(t) + \mathbf{u}^T(t)\mathbf{R}\mathbf{u}(t)\right) \mathrm{d}t, \tag{44}$$

then by substituting Equations (41) and (43) in Equation (44) we get

$$\begin{aligned} J(\mathbf{x}, \mathbf{u}) =& \mathbf{X}^T \left(\int_0^1 \widehat{\Phi}_{M,l}(t)\mathbf{Q}[\widehat{\Phi}_{M,l}(t)]^T \,\mathrm{d}t\right)\mathbf{X} \\ &+ \mathbf{U}^T \left(\int_0^1 \widehat{\Phi}_{M,q}(t)\mathbf{R}[\widehat{\Phi}_{M,q}(t)]^T \,\mathrm{d}t\right)\mathbf{U}. \end{aligned} \tag{45}$$

Equation (45) can be computed more efficiently by writing J as

$$J(\mathbf{x}, \mathbf{u}) = \mathbf{X}^T \left(\int_0^1 \mathbf{Q} \otimes \Phi_M(t)[\Phi_M(t)]^T \,\mathrm{d}t\right)\mathbf{X}$$

$$+\,\mathbf{U}^T\left(\int_0^1 \mathbf{R}\otimes\Phi_M(t)[\Phi_M(t)]^T\,\mathrm{d}t\right)\mathbf{U}. \tag{46}$$

Finally, $J(\mathbf{X},\mathbf{U})$ can be rewritten as

$$J(\mathbf{X},\mathbf{U}) = \mathbf{X}^T\left(\mathbf{Q}\otimes\mathbf{P}\right)\mathbf{X} + \mathbf{U}^T\left(\mathbf{R}\otimes\mathbf{P}\right)\mathbf{U}. \tag{47}$$

Otherwise, in the case that $\mathbf{L}(\mathbf{x}(t),\mathbf{u}(t),t)$ in (28) is an arbitrary function, we calculate it by a suitable Newton-Cotes numerical integration method [33] as

$$J(\mathbf{X},\mathbf{U}) = \sum_{i=0}^{n}\omega_i\mathbf{L}([\hat{\Phi}_{M,l}(t_i)]^T\mathbf{X},[\hat{\Phi}_{M,q}(t_i)]^T\mathbf{U},t_i),\quad t_i=\frac{i}{n}, i=1,2,\ldots,n \tag{48}$$

where the weight ω_i is determined by

$$\omega_i = \int_0^1 l_i(t)\,\mathrm{d}t,$$

and each $l_i(t)$ is the Lagrange polynomial

$$l_i(t) = \prod_{\substack{j=0\\ j\neq i}}^{n}\frac{t-\tau_j}{\tau_i-\tau_j}.$$

Finally, we approximate the dynamic system as follows.

Using Equations (41)-(43) the system constraints (29) and (31) become

$$\mathbf{Y}^T\widehat{\Phi}_{M,l}(t) = \mathbf{f}(\mathbf{X}^T\widehat{\Phi}_{M,l}(t),\mathbf{U}^T\widehat{\Phi}_{M,q}(t),t), \tag{49}$$

$$\mathbf{g}_j([\hat{\Phi}_{M,l}(t)]^T\mathbf{X},[\hat{\Phi}_{M,q}(t)]^T\mathbf{U},t)\leqslant 0,\qquad j=1,2,\ldots,w. \tag{50}$$

We collocate Equations (49) and (50) at

$$t_k = \frac{k-1}{2^M},\quad k=1,2,\ldots,2^M+1, \tag{51}$$

as

$$\mathbf{Y}^T\widehat{\Phi}_{M,l}(t_k) = \mathbf{f}(\mathbf{X}^T\widehat{\Phi}_{M,l}(t_k),\mathbf{U}^T\widehat{\Phi}_{M,q}(t_k),t_k), \tag{52}$$

$$\mathbf{g}_j([\hat{\Phi}_{M,l}(t_k)]^T\mathbf{X},[\hat{\Phi}_{M,q}(t_k)]^T\mathbf{U},t_k)\leqslant 0,\qquad j=1,2,\ldots,w. \tag{53}$$

In this way, we were able to turn FOCP into a nonlinear programming problem which can be stated as follows. Find $\mathbf{X}$ and $\mathbf{U}$ so that $J(\mathbf{X},\mathbf{U})$ in Equations (47) or (48) is minimized (or maximized) subject to Equations (52) and (53). To solve this nonlinear programming problem, we use the NLPSolve command in Maple software, which uses the sequential quadratic programming (SQP) method to solve NLP.

6 Convergence of the Method

To check the convergence of the proposed numerical method, we first express the existence of the optimal solution in the form of Filippov's existence theorem. Suppose the usual set of augmented velocities defined by

$$(f, L_+)(x, U, t) := \{(f(x, u, t), L(x, u, t) + \gamma) | u \in U, \gamma \geq 0\} \subset \mathbb{R}^{n+1},$$

for all $(x, t) \in \mathbb{R}^n \times [0, 1]$. Moreover let $\mathcal{T} \subset \mathcal{C}$ stand for the set of all trajectories x that can be associated with a control u such that the couple (x, u) satisfies all the constraints of the problem FOCP has given in Equations (28)-(31).

Theorem 1. (Filippov's existence theorem) Assume that U is compact, $\mathcal{T}$ is nonempty and bounded in C, and $(f, L_+)(x, U, t)$ is convex for all $(x, t) \in \mathbb{R}^n \times [0, 1]$. Then problem FOCP given in Equations (28)-(31) has at least one optimal solution.

Proof. Refer to [5]. □

Now we know that FOCP given in Equations (28)-(31) has at least one optimal solution of the form (x^*, u^*). So, to show that the method is convergent, it is sufficient $\|x - x^*\| \to 0$ and $\|u - u^*\| \to 0$ as $M \to \infty$ where x and u are approximate values obtained from the proposed numerical method and M is the parameter of the method related to the collocation points. We consider a linear B-spline space $\mathbb{S}_{M,\boldsymbol{\tau}} = \text{span}\{\phi_{M,-1}, \phi_{M,0}, \ldots, \phi_{M,2^M-1}\}$ where $\phi_{M,k}$, $k = -1, 0, \ldots, 2^M - 1$ are *B*-spline functions defined in Equations (4-6) also $\boldsymbol{\tau} = (\tau_j)_{j=1}^{2^M+1}$ where $\tau_j = \frac{j-1}{2^M}$. Assuming that $h_j = \tau_{j+1} - \tau_j$ and $h = \max_{j=1,\ldots,2^M+1} h_j$, we have $h = \frac{1}{2^M}$. For an arbitrary function f we consider the distance from f to $\mathbb{S}_{2,\tau}$ defined by

$$\text{dist}_{\infty,[0,1]}(f, \mathbb{S}_{M,\boldsymbol{\tau}}) = \inf_{g \in \mathbb{S}_{M,\tau}} \|f - g\|_{\infty,[0,1]}.$$

Theorem 2. Suppose that an arbitrary function $f \in C^3[0, 1]$ is given. Then for the linear B-spline space $\mathbb{S}_{M,\boldsymbol{\tau}}$

$$\text{dist}_{\infty,[0,1]}(f, \mathbb{S}_{2,\boldsymbol{\tau}}) \leq K h^3 \|D^3 f\|_{\infty,[0,1]},$$

where $K = \frac{1}{2^3 3!}$ and $D^3 f$ is the third derivative of the function f.

Proof. Refer to [22]. □

Now, according to $h = \frac{1}{2^M}$, by increasing the value of M sufficiently, we can bring the values of the state and control variables closer to their optimal values.

7 Illustrative Examples

In this section, by solving numerical examples, we will clarify the steps of using the proposed numerical method. We used the Maple 2015 program on a personal computer to perform numerical calculations

Example 1. We consider the following time-invariant FOCP from [2]

$$\min J = \frac{1}{2}\int_0^1 [x^2(t) + u^2(t)]\,\mathrm{d}t, \tag{54}$$

subject to the system dynamics

$$D^\alpha x(t) = -x(t) + u(t), \tag{55}$$

and the initial condition

$$x(0) = 1. \tag{56}$$

The exact solution to this problem in the case $\alpha = 1$ is

$$\begin{aligned} \bar{x}(t) &= \cosh(\sqrt{2}t) + \beta \sinh(\sqrt{2}t), \\ \bar{u}(t) &= (1 + \sqrt{2}\beta)\cosh(\sqrt{2}t) + (\sqrt{2} + \beta)\sinh(\sqrt{2}t), \end{aligned}$$

where

$$\beta = -\frac{\cosh(\sqrt{2}) + \sqrt{2}\sinh(\sqrt{2})}{\sqrt{2}\cosh(\sqrt{2}) + \sinh(\sqrt{2})} \simeq -0.979921727.$$

The optimal value of the performance index with the exact solution is $J = 0.1929093$. Assume that $\widetilde{x}(t)$, $\widetilde{u}(t)$ and $\widetilde{J}(x,u)$ are the approximate values obtained from numerical methods for the state, control, and objective functions respectively. Then the error is given by

$$\begin{aligned} E_x &= \max_i(|\bar{x}(t_i) - \widetilde{x}(t_i)|), \\ E_u &= \max_i(|\bar{u}(t_i) - \widetilde{u}(t_i)|), \\ E_J &= |\bar{J} - \widetilde{J}|, \end{aligned}$$

where $t_i = \frac{i+1}{2^M}$, $i = -1, \dots, 2^M - 1$. Figure 1 demonstrates state and control variables obtained by our numerical method for $M = 8$ and different values of α. Figure 2 shows the logarithmic graphs of MAEs (Maximum Absolute Errors) of $x(t)$, $u(t)$ and J for $\alpha = 1$ and different values of M. Given these figures, the convergence of the method can be deduced. Tables 1 and 2 show the absolute errors of the approximate optimal state $\widetilde{x}(t)$ and the absolute error of the optimal control $\widetilde{u}(t)$ respectively. Table 3 shows the approximate value of the performance index $\widetilde{J}$ and its error with the exact value of $\bar{J}$.

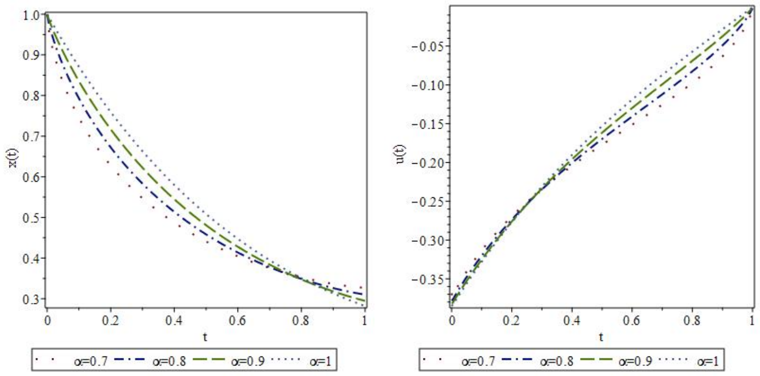


Figure 1: State $x(t)$ and control $u(t)$ functions for Example 1.

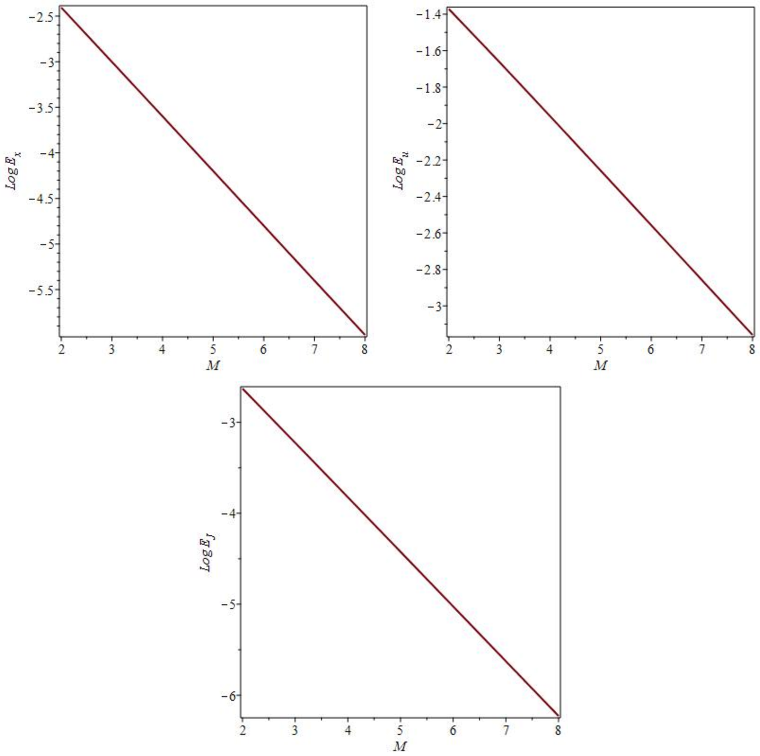


Figure 2: Logarithmic graphs of MAEs for Example 1.

Table 1: The absolute errors of the approximate optimal states for Example 1

t	**Method of [15]** for $N=5$	**Method of [35]** for $N=5$	**Method of [13]** for $M=8$	**Presented method** for $M=8$
0.1	2.11×10^{-5}	6.90×10^{-7}	1.44×10^{-6}	3.52×10^{-6}
0.2	9.71×10^{-6}	3.62×10^{-6}	1.36×10^{-6}	1.86×10^{-6}
0.3	4.08×10^{-7}	1.97×10^{-6}	1.23×10^{-6}	1.40×10^{-6}
0.4	5.76×10^{-7}	2.58×10^{-6}	1.01×10^{-6}	1.74×10^{-6}
0.5	5.66×10^{-6}	4.46×10^{-6}	2.92×10^{-7}	4.89×10^{-7}
0.6	9.25×10^{-6}	1.65×10^{-6}	7.79×10^{-7}	1.08×10^{-6}
0.7	8.35×10^{-6}	2.80×10^{-6}	7.85×10^{-7}	3.67×10^{-7}
0.8	4.36×10^{-6}	3.49×10^{-6}	6.71×10^{-7}	2.44×10^{-7}
0.9	2.59×10^{-6}	1.22×10^{-6}	5.32×10^{-7}	5.24×10^{-7}

Table 2: The absolute errors of the approximate optimal controls for Example 1

t	**Method of [35]** for $N=5$	**Method of [15]** for $N=5$	**Method of [13]** for $M=9$	**Presented method** for $M=9$
0.1	1.90×10^{-5}	6.74×10^{-6}	1.26×10^{-6}	1.56×10^{-6}
0.2	5.01×10^{-6}	3.17×10^{-6}	4.68×10^{-6}	7.33×10^{-7}
0.3	1.46×10^{-5}	5.92×10^{-7}	4.49×10^{-6}	4.37×10^{-7}
0.4	1.47×10^{-5}	7.10×10^{-7}	1.23×10^{-6}	4.51×10^{-7}
0.5	1.25×10^{-6}	2.01×10^{-6}	7.31×10^{-6}	3.20×10^{-7}
0.6	1.07×10^{-5}	2.71×10^{-6}	1.19×10^{-6}	7.20×10^{-8}
0.7	1.27×10^{-5}	2.11×10^{-6}	3.83×10^{-6}	1.57×10^{-7}
0.8	7.62×10^{-6}	8.59×10^{-7}	3.68×10^{-6}	1.74×10^{-7}
0.9	1.74×10^{-5}	8.93×10^{-8}	1.15×10^{-6}	5.36×10^{-8}

Table 3: The approximate values and their errors with exact values of $\bar{J}$ for Example 1

Method of [35]			**Presented method**		
N	$\bar{J}$	$E_J=\lvert\bar{J}-\widetilde{J}\rvert$	M	$\bar{J}$	$E_J=\lvert\bar{J}-\widetilde{J}\rvert$
2	0.1926605504081	2.48×10^{-4}	5	0.192909340640602	4.25×10^{-8}
3	0.1929127052722	3.41×10^{-6}	6	0.192909300753628	2.66×10^{-9}
4	0.1929092715551	2.65×10^{-8}	7	0.192909298259509	1.66×10^{-10}
5	0.1929092982262	1.33×10^{-10}	8	0.192909298103643	1.04×10^{-11}
6	0.1929092980936	6.15×10^{-13}	9	0.192909298093859	6.59×10^{-13}

Example 2. Consider the following fractional optimal control problem that was introduced in [21] and also was studied in [6]

$$\text{Min } J=\int_0^1\left[\left(x(t)-t^2\right)^2+\left(u(t)+t^4-\frac{20t^{\frac{9}{10}}}{9\Gamma(\frac{9}{10})}\right)^2\right]dt,$$

subject to the dynamic constraints

$$
\begin{aligned}
&D^{1.1}x(t) = t^2 x(t) + u(t),\\
&x(0) = \dot{x}(0) = 0.
\end{aligned}
$$

The exact solution to this problem is given by

$$
\begin{aligned}
&\bar{x}(t) = t^2,\\
&\bar{u}(t) = \frac{20t^{\frac{9}{10}}}{9\Gamma(\frac{9}{10})} - t^4,\\
&\bar{J} = 0.
\end{aligned}
$$

The exact and approximate values of the state and control variables are illustrated in Figure 3, and their errors are plotted in Figure 4. The logarithmic graphs of MAEs of state and control variables and performance index are shown in Figure 5. In Table 4, the approximate values of the performance index J for different values of M, are presented. Also, these values are compared with similar methods in [6, 21]. According to Table 4, the presented method is more accurate than the existing methods.

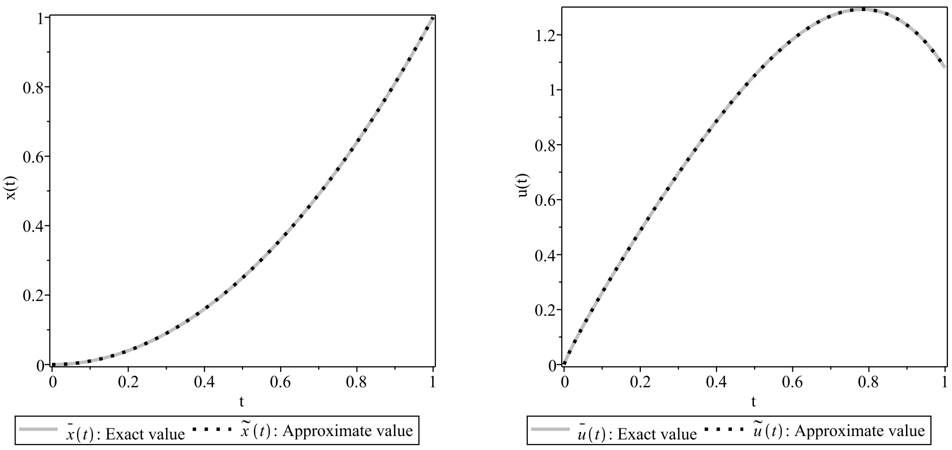


Figure 3: The values of $x(t)$ and $u(t)$ obtained by $M = 8$ for Example 2.

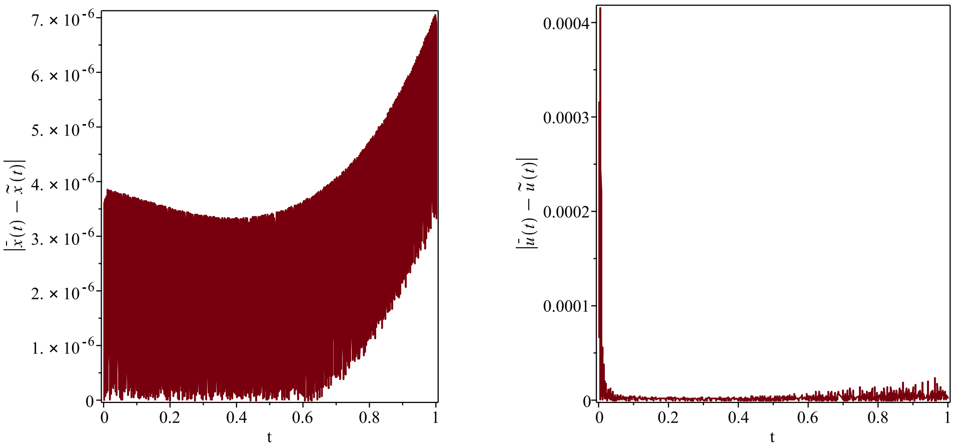


Figure 4: The values of errors of $x(t)$ and $u(t)$ obtained by $M = 8$ for Example 2.

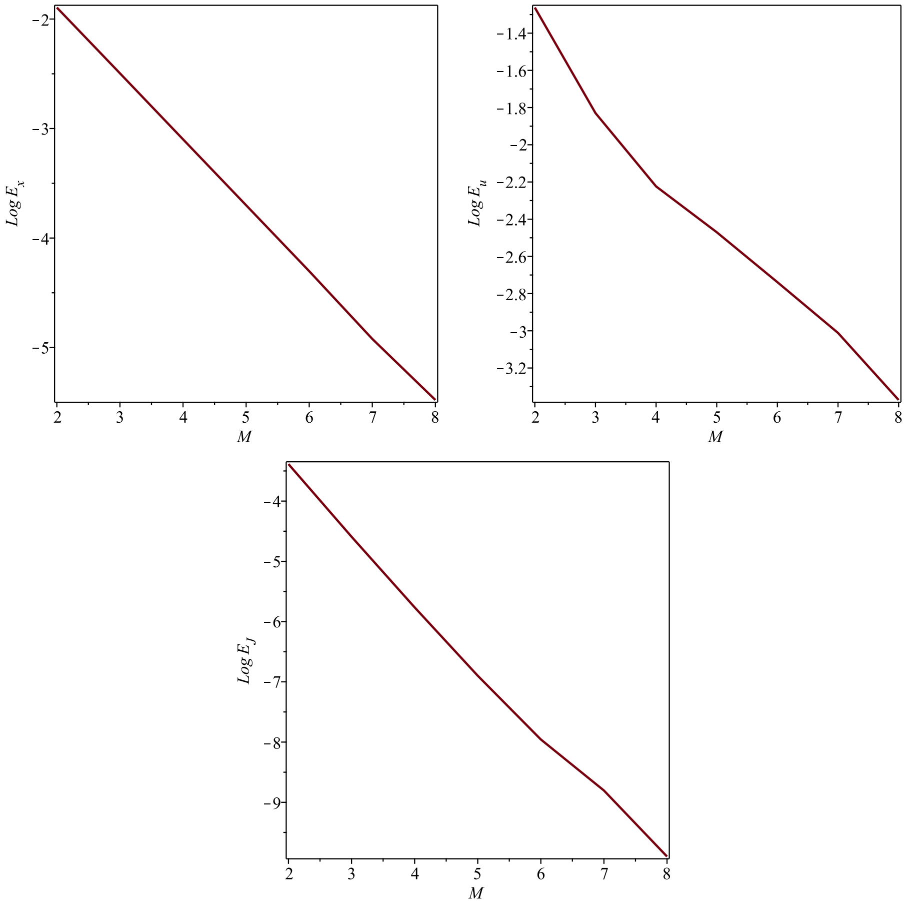


Figure 5: Logarithmic graphs of MAEs for Example 2.

Table 4: Approximate values of J for Example 2

Methods	Parameters of method	$J(x,u)$
Method of [21]	$(m=3, n=4)$	6.0753×10^{-6}
	$(m=4, n=5)$	1.67255×10^{-6}
	$(m=5, n=6)$	5.91532×10^{-7}
	$(m=7, n=8)$	1.21966×10^{-7}
	$(m=8, n=9)$	7.03371×10^{-8}
Method of [6]	$N=4$	4.76932×10^{-6}
	$N=5$	1.47243×10^{-6}
	$N=6$	5.37825×10^{-7}
	$N=8$	1.06099×10^{-7}
	$N=9$	5.44304×10^{-8}
The present method	$M=4$	$1.72145571012670767 \times 10^{-6}$
	$M=5$	$1.26295831601775329 \times 10^{-7}$
	$M=6$	$1.10567794682908139 \times 10^{-8}$
	$M=7$	$1.57143831079185382 \times 10^{-9}$
	$M=8$	$1.26073358425454064 \times 10^{-10}$

Example 3. Consider the following FOCP [21]

$$\text{Min } J = \int_0^1 \Bigg[\exp(t)\left(x(t) - t^4 + t - 1\right)^2 + \left(1+t^2\right)\left(u(t) + 1 - t + t^4 - \frac{8000 t^{\frac{21}{10}}}{77\Gamma\left(\frac{1}{10}\right)}\right)^2 \Bigg] \mathrm{d}t,$$

subject to the dynamic system

$$D^{1.9}x(t) = x(t) + u(t), \qquad t \in [0,1],$$

and the boundary conditions

$$x(0) = 1, \quad \dot{x}(0) = -1.$$

The exact solution is given by

$$\bar{x} = 1 - t + t^4,$$
$$\bar{J} = 0.$$

In Figure 6, the exact and approximate values of the state variable and approximate value of the control variable with $M = 8$ are illustrated. Moreover, we plotted the error value of $x(t)$ in Figure 7. The MAEs of state vector $x(t)$ and performance index J are plotted in Figure 8.

Example 4. In this example, we present a problem involving a two-dimensional state variable and an inequality constraint

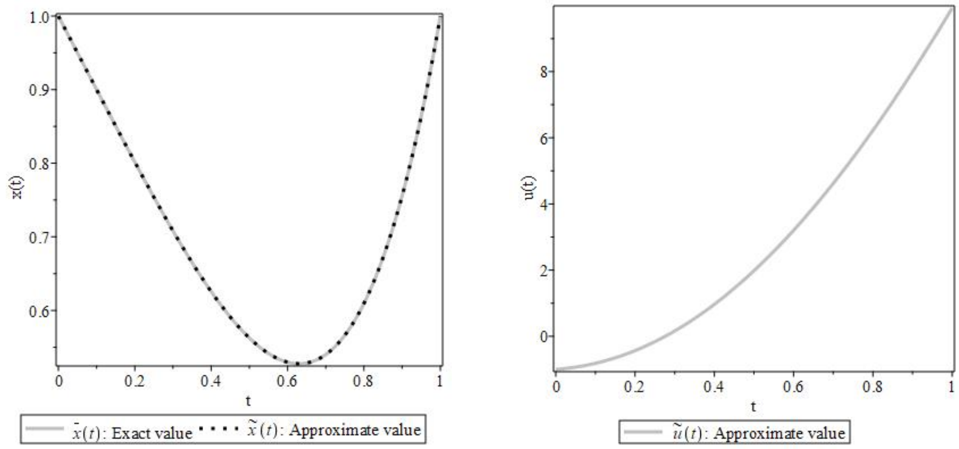


Figure 6: The values of $x(t)$ and $u(t)$ obtained by $M = 8$ for Example 3.

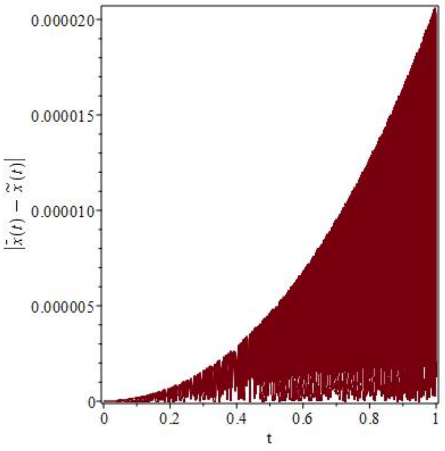


Figure 7: The values of errors of $x(t)$ obtained by $M = 8$ for Example 3.

$$\min J = \frac{1}{2}\int_0^1 u^2(t)\,\mathrm{d}\,t,$$

subject to

$$\begin{aligned}
&\mathrm{D}^{\alpha}\, x_1(t) = x_2(t),\\
&\mathrm{D}^{\alpha}\, x_2(t) = u(t),\\
&x_1(t) \leq 0.1,\\
&x_1(0) = x_1(1) = 0,\\
&x_2(0) = -x_2(1) = 1.
\end{aligned}$$

The exact values of the control variable for $\alpha = 1$ are

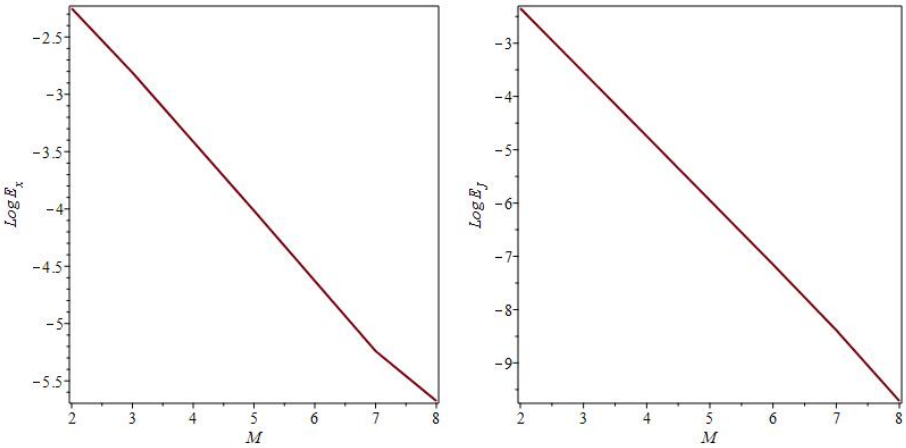


Figure 8: Logarithmic graphs of MAEs for Example 3.

Table 5: Approximate values of J for Example 3

Methods	Parameters of the method	$J(x,u)$
The method of [21]	$m=n=3$	8.93768×10^{-6}
	$m=n=4$	5.42028×10^{-7}
	$m=n=5$	6.77757×10^{-8}
	$m=n=7$	2.84624×10^{-9}
	$m=n=8$	8.22283×10^{-10}
The present method	$M=4$	$1.80165706993258757\times10^{-5}$
	$M=5$	$1.12585635458861543\times10^{-6}$
	$M=6$	$7.02177422426594986\times10^{-8}$
	$M=7$	$4.12444733804727959\times10^{-9}$
	$M=8$	$1.92637108047034916\times10^{-10}$

$$u^*(t)=\begin{cases}\frac{200}{9}t-\frac{20}{3}, & t\in[0,0.3],\\ 0, & t\in[0.3,0.7],\\ -\frac{200}{9}t+\frac{140}{9} & t\in[0.7,1].\end{cases}$$

Figure 9 shows the exact and approximate states and control variables obtained by the proposed method for $M=8$ and $\alpha=1$.

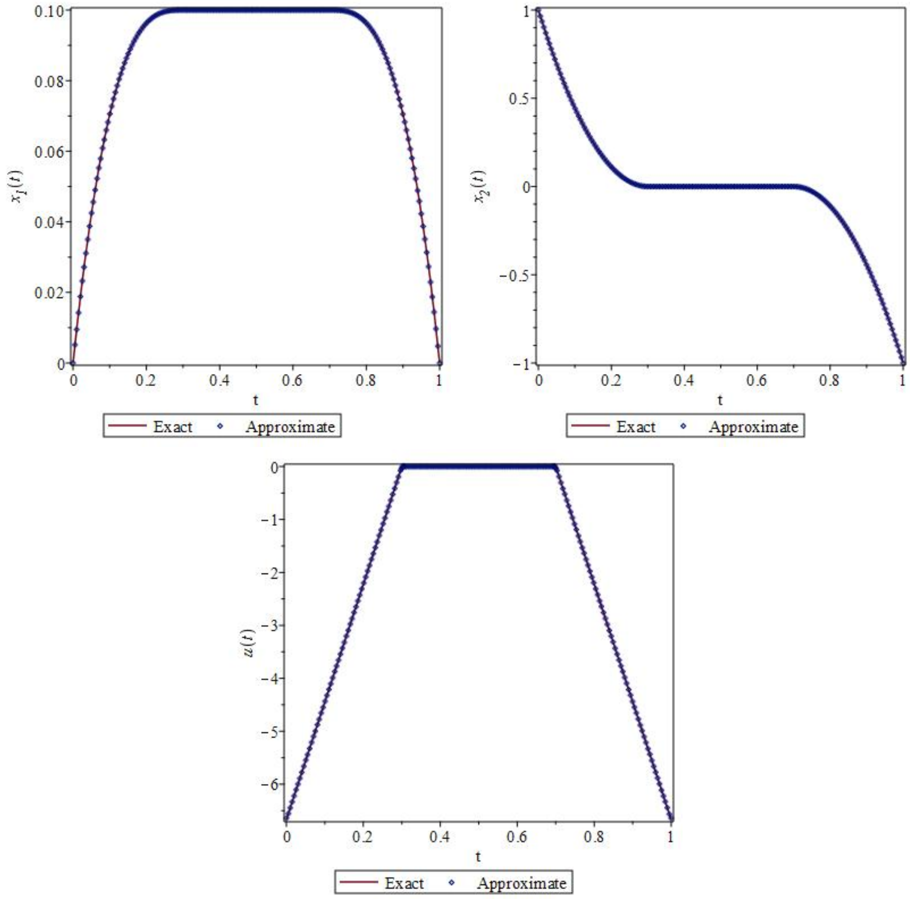


Figure 9: State functions $x_1(t)$ and $x_2(t)$ and control function $u(t)$ for Example 4.

8 Conclusion

In this paper, we presented a numerical method with an emphasis on better accuracy than similar tasks. In this method, we used B-spline functions, and the distinguishing feature of this work is the use of a fractional integral operational matrix in solving the FOCPs. We managed to turn FOCP into NLP with the help of this matrix. Using several numerical examples, we were able to show the high efficiency, and accuracy of the proposed method. In addition, by increasing the value of M, the accuracy of the method increases, and in cases where there is an exact solution, the approximate value converges to the exact solution, and also the error is reduced. For future research, more accurate approximations can be achieved by extending the basic functions for approximation.

References

[1] Adams M. P. (2018). "Numerical schemes for the fractional calculus and their application to image feature detection", (Master's thesis, Graduate Studies).

[2] Agrawal O. P. (1989). "General formulation for the numerical solution of optimal control problems", International Journal of Control, 50, 627-638.

[3] Baghani O. (2021). "Second Chebyshev wavelets (SCWs) method for solving finite-time fractional linear quadratic optimal control problems", Mathematics and Computers in Simulation, 190, 343-361.

[4] Baghani O. (2022). "SCW-iterative-computational method for solving a wide class of nonlinear fractional optimal control problems with Caputo derivatives", Mathematics and Computers in Simulation, 202, 540-558.

[5] Bergounioux M., Bourdin L. (2020). "Pontryagin maximum principle for general Caputo fractional optimal control problems with Bolza cost and terminal constraints", ESAIM: Control, Optimisation and Calculus of Variations, 26, 35.

[6] Bhrawy A. H., Doha E. H., Baleanu D., Ezz-eldien S. S., Abdelkawy M. A. (2015). "An accurate numerical technique for solving fractional optimal control problems", Differential Equations, 15, 23.

[7] Darehmiraki M. (2018). "An efficient solution for stochastic fractional partial differential equations with additive noise by a meshless method", International Journal of Applied and Computational Mathematics, 4, 14.

[8] Das S. (2011). "Functional fractional calculus", Springer-Verlag Berlin Heidelberg.

[9] Doha E. H., Bhrawy A. H., Baleanu D., Ezz-Eldien S. S., Hafez R. M. (2015). "An efficient numerical scheme based on the shifted orthonormal Jacobi polynomials for solving fractional optimal control problems", Advances in Difference Equations, 15.

[10] Edrisi Tabriz Y., Heydari A. (2014). "Generalized B-spline functions method for solving optimal control problems", Computational Methods for Differential Equations, 2, 243-255.

[11] Edrisi Tabriz Y., Lakestani M. (2015) "Direct solution of nonlinear constrained quadratic optimal control problems using B-spline functions", Kybernetika, 51, 81-98.

[12] Edrisi Tabriz Y., Lakestani M., Heydari A. (2016). "Two numerical methods for nonlinear constrained quadratic optimal control problems using linear B-spline functions", Iranian Journal of Numerical Analysis and Optimization, 6, 17-38.

[13] Edrisi-Tabriz Y., Lakestani M., Razzaghi M. (2021). "Study of B-spline collocation method for solving fractional optimal control problems", Transactions of the Institute of Measurement and Control, 43(11), 2425-37.

[14] Goswami J. C., Chan A. K. (2011). "Fundamentals of wavelets: theory, algorithms, and applications", John Wiley & Sons.

[15] Habibli M., Noori Skandari M. H. (2019). "Fractional Chebyshev pseudo spectral method for fractional optimal control problems", Optimal Control Applications and Methods, 40(3), 558-572.

[16] Lakestani M., Razzaghi M., Dehghan M. (2006). "Semiorthogonal spline wavelets approximation for Fredholm integro-differential equations", Mathematical Problems in Engineering.

[17] Lakestani M., Razzaghi M., Dehghan M. (2005). “Solution of nonlinear Fredholm-Hammerstein integral equations by using semiorthogonal spline wavelets”, Mathematical Problems in Engineering, 113-121.

[18] Lakestani M., Dehghan M., Irandoust-Pakchin S. (2012). “The construction of operational matrix of fractional derivatives using B-spline functions”, Communications in Nonlinear Science and Numerical Simulation, 17, 1149-1162.

[19] Lancaster P., Tismenetsky M. (1985). “The theory of matrices: with applications”, Elsevier.

[20] Laskin N. (2018). “Fractional quantum mechanics”, World Scientific.

[21] Lotfi A., Yousefi S. A., Dehghan M. (2013). “Numerical solution of a class of fractional optimal control problems via the Legendre orthonormal basis combined with the operational matrix and the Gauss quadrature rule”, Journal of Computational and Applied Mathematics, 250, 143-160.

[22] Lyche T., Morken K. (2008). “Spline methods draft”, Department of Informatics, Center of Mathematics for Applications, University of Oslo, Oslo, 3-8.

[23] Magin R., Vinagre B., Podlubny I. (2018). “Can cybernetics and fractional calculus be partners?: Searching for new ways to solve complex problems”, IEEE Systems, Man, and Cybernetics Magazine, 4(3), 23-28.

[24] Nemati A., Yousefi S., Soltanian F., Ardabili J. S. (2016). “An efficient numerical solution of fractional optimal control problems by using the Ritz method and Bernstein operational matrix”, Asian Journal of Control, 18, 2272-2282.

[25] Rabiei K., Ordokhani Y., Babolian E. (2016). “The Boubaker polynomials and their application to solve fractional optimal control problems”, Nonlinear Dynamic, 88, 1013-1026.

[26] Rao A. V. (2009). “A survey of numerical methods for optimal control”, Advances in the Astronautical Sciences, 135, 497-528.

[27] Saeedi H. (2017). “The linear b-spline scaling function operational matrix of fractional integration and its applications in solving fractional-order differential equations”, Iranian Journal of Science and Technology, Transaction A, 41, 723-733.

[28] Schoenberg I. J. (1973). “Cardinal spline interpolation”, Society for Industrial and Applied Mathematics.

[29] Schoenberg I. J. (1946). “Contributions to the problem of approximation of equidistant data by analytic functions Part B. On the problem of osculatory interpolation. A second class of analytic approximation formulae”, Quarterly of Applied Mathematics, 4(2), 112-141.

[30] Shukla M. K., Sharma B. B., Azar A. T. (2018). “Control and synchronization of a fractional order hyperchaotic system via backstepping and active backstepping approach”, In Mathematical Techniques of Fractional Order Systems, Elsevier, 559-595.

[31] Skandari M. H. N., Habibli M., Nazemi A. (2020). “A direct method based on the Clenshaw-Curtis formula for fractional optimal control problems”, Mathematical Control & Related Fields, 10(1), 171.

[32] Sopasakis P., Sarimveis H., Macheras P., Dokoumetzidis A. (2018). “Fractional calculus in pharmacokinetics”, Journal of pharmacokinetics and pharmacodynamics, 45(1), 107-125.

[33] Stoer J., Bulirsch R. (2013). “Introduction to numerical analysis”, Springer Science & Business Media.

[34] Unser M., Blu T. (2000). “Fractional Splines and Wavelets”, SIAM Review, 42, 43-67.

[35] Xiaobing P., Yang X., Skandari M. H. N., Tohidi E., Shateyi S. (2022). “”A new high accurate approximate approach to solve optimal control problems of fractional order via efficient basis functions”, Alexandria Engineering Journal, 61(8), 5805-5818.

[36] Yaghi M., Efe M. Ö. (2018). “Fractional order PID control of a radar guided missile under disturbances”, In: Information and Communication Systems (ICICS), 2018 9th International Conference, 238-242.

[37] Yang Y., Noori Skandari M. H. (2020). “Pseudo spectral method for fractional infinite horizon optimal control problems”, Optimal Control Applications and Methods, 41(6), 2201-2212.

[38] Yang X., Yang Y., Skandari M. N., Tohidi E., Shateyi S. (2022). “A new local non-integer derivative and its application to optimal control problems”, AIMS Mathematics, 7(9), 16692-16705.

[39] Yonthanthum W., Rattana A., Razzaghi M. (2018). “An approximate method for solving fractional optimal control problems by the hybrid of block-pulse functions and Taylor polynomials”, Optimal Control Applications and Methods, 39, 873-887.

How to Cite this Article:

Edrisi-Tabriz, Y. (2022). “Using the Integral Operational Matrix of *B*-Spline Functions to Solve Fractional Optimal Control Problems”, Control and Optimization in Applied Mathematics, 7(2): 77-98. doi: 10.30473/coam.2022.62261.1207

Control and Optimization in Applied Mathematics (COAM)
Vol. 7, No. 2, Summer-Autumn 2022 (99-114), ©2016 Payame Noor University, Iran

DOI. 10.30473/coam.2022.63182.1197 (Cited this article)

Research Article

Application of the Mixed-Integer Programming Method in Fishery Supply Chain Network Management: A Case Study of Shrimp in Golestan Province

Javad Mahdavi Varaki[1], **Iraj Mahdavi**[1], **Shahrzad Mirkarimi**[2,*]

[1]Department of Industrial Engineering, University of Science and Technology,
P.O. Box. 85635-47166, Mazandaran, Babol, Iran.
[2]Department of Agricultural Engineering, University of Agricultural Sciences and Natural Resources,
P.O. Box. 48181-66996, Mazandaran, Sari, Iran.

Received: February 26, 2022; **Accepted:** September 18, 2022.

Abstract. Social, economic, and environmental issues such as population growth, reduction of natural resources, climate change, market fluctuations, and changing consumer behavior have attracted the attention of politicians to the supply chain of agricultural products. Designing an effective supply chain for each product can lead to optimal management of the agricultural sector and create coordination and links between activities. In this article, the design of a two-echelon supply chain network of shrimp in Golestan province is investigated. The objective is to minimize the total cost associated with fixed opening and operating costs of shrimp farming companies and to determine the target market for these producers. Also, this study involves deciding on the amount of inputs purchased by each company and determining the best mode of transport. To characterize and solve this problem, we developed a mixed-integer programming (MIP) model that solves with GAMS software. The results show that with the implementation of the MIP model, the total costs of the chain are reduced by nearly 20 percent compared to the current situation. In addition, without increasing production, it is possible to supply 0.053 percent of global market demand, which is 76 percent more than before.

Keywords. Programming optimization, Supply Chain, Golestan province, Farmed shrimp.

MSC. 00A06; 03C98.

[*] Corresponding author

Javadmahdavivaraki@gmail.com, Irajmahdavi@gmail.com, shahrzadmirkarimi@yahoo.com

1 Introduction

One of the most fundamental and essential challenges of the present and future is the issue of food and food security. Nearly 800 million people in the world's population are currently hungry or suffering from severe malnutrition. The growing world population and the growing need for protein require optimal solutions in providing food resources. Of the four billion tons of food consumed by humans, 97 percent comes from 3 to 5 percent of the land level, which can be cultivated, but from 71 percent of the land, which is the sea, only 3 percent of human food is supplied. Given the global constraints of agriculture and livestock, humans must increase aquaculture [1]. In recent years, uncontrolled fishing in the Caspian Sea, the Persian Gulf, and the Oman Sea has resulted in drastic decreases in the reserves of these water resources. The expansion of aquaculture farms not only has contributed to the development of a sustainable source of food for the country but also has been highly effective in the preservation of species that are endangered for whatever reason. Today, seafood and related products have been known for job creation and earning foreign exchange [23]. The high nutritional value of shrimp, on the one hand, and the demand of global markets, on the other hand, has caused shrimp production to play a special role in aquatics. Shrimp as a well-known, rich, and sought-after seafood, is generally obtained from either marine environments or aquaculture [16]. According to the acceptable efficiency of the shrimp industry in the world and its hidden talents in Iran, actions have been taken to utilize shrimp farms in the country [7]. In 2018, about 10 percent of the country's aquatic production has been allocated to shrimp products [10]. Shrimp farming, in addition to currency, in border areas is effective in eliminating smuggling, increasing the security of areas, protecting border residents, creating jobs, and preventing migration of villagers [9]. Therefore, the development of the shrimp industry and related industries has favorable economic consequences [5]. In addition to the reproduction of this product, its distribution, domestic sales, and exports are very important. In 2018, 47.9 thousand tons of shrimp were produced in the country, of which 31.8 tons, equivalent to 66 percent, were exported. Also, the total export revenue of fishery products (caviar, shrimp, and types of fish) is reported to be 528.3 million dollars, of which shrimp exports account for 159 million dollars, about 30 percent of the total export revenue of total aquatic products [10].

Today, due to the size of global markets and the existence of some major social and economic differences between countries and consumer groups, the use of a principled method to identify or so-called determine and prioritize export target markets is one of the requirements to achieve an export leap [12]. Consequently, designing and optimizing the shrimp supply chain network can help governments, investors, and active parties to satisfy market demands, and to overcome obstacles in the supply chain, and in general, can boost the performance of the whole chain [16]. Traditional supply chain practices may be under revision due to issues related to food security [2]. To keep up with the changes occurring in agricultural supply chains, all parties involved must be considered. So, planning models will become of increasing importance to suppliers, farmers, intermediaries, and final distributors of agricultural commodities. Planning tools for each of these people must become increasingly refined to drive extra costs out of the value chain [22]. The above highlights reveal the need to pay attention to the shrimp supply chain. The supply chain is a network of facilities and distribution options for the procurement of materials; the transformation of materials into intermediate and finished products; and the distribution of these finished products to customers. The Agri-food supply chain comprises all the stages that food products go through, from production to consumption. These activities include inputs, production, conversion, processing, packaging, warehousing, transportation, cross-docking, distribution, marketing, and consumption [11]. In other words, the Agri-food supply chain network is typically a multi-echelon supply chain network with multiple products, including four stages: primary production, production of semi-products by plants, production of finished products, and distribution. Decisions on determining the optimal number, location, and capacity of the production companies, product type produced by each company, selecting transportation mode and the corresponding amount of items shipping from supplier to the company, between two companies, and from companies to distribution centers are made given the market demand, such that the total costs are minimized [26]. The existence of diverse activities causes supply chain planning issues to exhibit a multilevel decision-making network structure. Therefore, it is necessary to optimize the economic flow from input suppliers to manufacturing companies and then to consumers [11].

In this paper, the shrimp supply chain is presented, which reveals the innovation of the present study. It should be noted that shrimp, unlike fish, can be grown only in special conditions and places, and therefore its proper location to supply consumer demand, can play a significant role in the reduction of costs. In Section 2, the literature review on the supply chain is reviewed. In Section 3, the problem model is presented with emphasis on the mixed integer programming (MIP) method. Due to the NP-hardness of supply chain problems and the large-sized data in the real world, metaheuristics such as genetic algorithm and particle swarm optimization have been widely used. However, as well as in other optimization problems, the solution technique of MIP models can be employed in supply chain problems. In Section 4, we show the computational results, and conclusions are provided in Section 5.

2 Literature Review

According to a review of the research literature mentioned below, the use of mathematical models and operations research tools for agricultural planning is not a new concept. Instead, optimization models for applications in crop planning can be found since the early 1950s, even in a tenuous way. This solution technique became more widespread during the decade of 1980, with growing interest in the 1990s [2]. Ayoughi et al. (2022) in [4] presented a stable multi-objective model of location, inventory, and supply chain routing under conditions of uncertainty and using a passive defense approach. Parameters such as demand, cost of setting up the facility and cost of maintaining inventory were considered uncertain and in the form of triangular fuzzy numbers. The results of validation showed that the proposed model was valid and feasible, and the proposed algorithm was also valid and converged to the optimal solution. Mosallanezhad et al. (2021) in [16] considered Shrimp Supply Chain (SSC) as a set of distribution centers, wholesalers, shrimp processing factories, markets, shrimp waste powder factories, and shrimp waste powder market. In this paper, a mathematical model was proposed for the SSC, whose aim was to minimize the total cost through the supply chain. The SSC model was NP-hard and was not able to solve large-size problems. Therefore, three well-known meta-heuristics accompanied by two-hybrid ones were exerted. Moreover, a real-world application with 15 test problems was established to validate the model. Finally, the results confirmed that the SSC model and the solution methods were effective and useful to achieve cost savings. Salehi and Jabarpour (2020) in [20] discussed a multi-objective model for multi-period location-distribution-routing problems considering the evacuation of casualties and homeless people and fuzzy paths in relief logistics. Some parameters were considered uncertain, including demand, the capacity of vehicles and time. What distinguishes humanitarian logistics from ordinary logistics is that under critical conditions, the relief supply chain must act at high speed and aim to preserve human lives. While under ordinary circumstances, the supply chain operates at the lowest cost according to the schedule. In this paper, the small sample size problem solved by the GAMS software was solved by these algorithms, which showed the high efficiency of the algorithms in obtaining efficient responses. Tabrizi et al. (2017) in [23] presented a bi-level optimization modeling for the perishable food supply chain. They designed a warm-water-farmed fish supply chain in Iran. This study aimed to maximize the profitability of farms based on the meta-heuristic particle swarm optimization method. The results showed the efficiency of the proposed model in solving the real problems of the perishable food supply chain.

A review of previous studies indicates that supply chain network design has been considered in various issues. Some authors have addressed the issue of location-allocation ([4], [15], [19]-[24]). Some studies have developed these models by considering multiple echelons [17] and multiple products [21]. Also, considering uncertain parameters in the optimization problems (such as demand) is another popular extension of the classical supply chain network design [24]. Closed-loop supply chain design ([3], [8]-[21]) and sustainable supply chain design [25] are two other popular areas in supply chain literature. In addition, considering specific products such as dangerous and perishable products [18], pricing decisions [3], designing resilience supply chains [19], competitive networks [6], selecting transport modes and integrating them with other decisions chains [22] and considering quantity discounts to reduce costs [11] are the topics that have become very popular in the literature of supply chain network design in recent years. This article designs a mathematical modeling and optimization structure that focuses on a two-echelon shrimp supply chain network. To the best of our knowledge, no prior study involving mathematical modeling for the shrimp supply chain network design considers transportation issues and quantity discounts simultaneously. The main goal of this model is to minimize the total cost.

3 Problem Definition and Modeling

Golestan province, with its potential areas for aquaculture, is considered as one of the important centers of the fisheries industry in the country. In 2018, this province, after Hormozgan and Bushehr provinces with 5.4 percent had the highest production of saltwater-farmed shrimp. Also, 3.7 percent of the number of farms and about 11 percent of the area of shrimp farms in the country are active in this province [10]. Gomishan Shrimp Farming Complex is the only potential complex in this province that, according to the climatic conditions of the region, is considered as one of the most important sectors of development in this province to produce protein materials and create employment. In 2011, the Gomishan Complex started operating with 70 hectares of useful area and a production of 140 tons of shrimp. At present, the first phase of the shrimp farming complex with 50 20-hectare farms is operating under the supervision of 13 companies [13]. Shrimp caught from these farms are supplied to domestic and foreign markets after processing. Before the outbreak of coronavirus, the countries like China, Vietnam, Emirates, Hong Kong, Oman, and Spain were the main export destinations for shrimp from Golestan province. Figure 1 shows the network structure of a two-echelon supply chain of farmed shrimp, where level (1) is between the input suppliers and the shrimp breeders, and level (2) is between the breeders and the consumers.

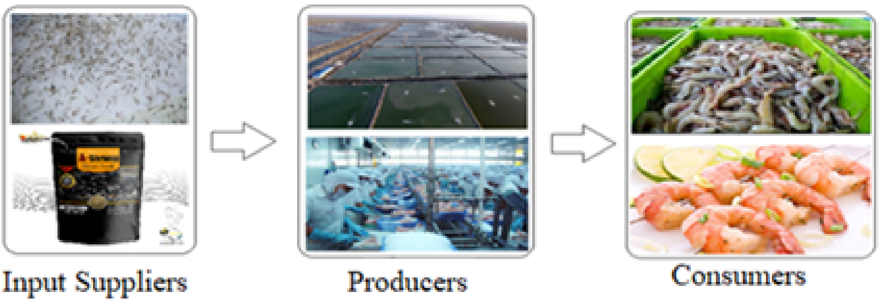


Figure 1: Network structure of a two-echelon supply chain.

In this study, it is assumed that (1). The inputs required for shrimp farming are provided by two groups of domestic and foreign suppliers. Domestic inputs such as water, labor, and larvae are provided by the first group and imported inputs such as fertilizers, vitamins, supplements, disinfectants, and food are provided by the second group ($s = 1, 2$). It should be noted that among the inputs, feed and released larvae account for 65 and 20 percent, respectively, and play an important role in creating the total cost of this product [10]; so the capacity of suppliers has been assessed on this basis. (2). The processing operation is done inside the complex and by the producers. (3). Consumers of this product are wholesalers who can be divided into three groups; within-province (α), within-country (β) and abroad (γ) ($i = 1, 2, 3$). The position of consumers and suppliers of inputs is clear. (4). All the facilities considered in this network have a certain capacity and demand. (5). Each shrimp farming company serves only one group of consumers (single-sourcing strategy). (6). Each company can refer to any of the suppliers to buy the required inputs (multiple-sourcing strategy). (7). Both transportation modes cannot be used simultaneously for products and raw material transmissions between levels. In other words, only one of the available modes can be selected for transporting goods from suppliers to companies or from companies to consumers ($m = 1, 2$). The first mode of transportation cannot be used when the total quantity of carried raw material from supplier to companies is less than CV^1 or the total quantity of carried products from a farm to a consumer is less than CV^2. (8). The costs of purchasing and transporting inputs are reserved for suppliers. Input suppliers usually offer price discounts to encourage companies to buy larger quantities of inputs. Therefore, the existence of a discount creates an incentive for producers to reduce the cost of each unit of production by buying more. The amount of the discount affects the amount of the order and the amount of the order also affects the mode of transportation.

In this study, suppliers sell their inputs based on an All-Unit discount strategy. In this strategy, a price reduction is applied to all purchased units. Different breakdown points are also introduced by suppliers. We use a piecewise linear purchasing and transportation cost between suppliers and farms. Let Q_{sk} denote the amount of materials carried from supplier s to farm k. We assume that there are | l_s | purchasing price segments (price breakdown points) for carrying products and BP_{sls} is the quantity of l_s^{th} breakdown point in each supplier. Therefore, we have | l_s | -1 price intervals in total. If we consider P_{sls} as the unit purchasing and transportation price of raw materials in $(l_s - 1)^{th}$ interval of supplier s, then the total purchasing and transportation cost associated with the purchase between supplier s and company k is as follows:

$$c_{sk} = \begin{cases} 0, & c_{sk} = 0, \\ P_{s2} * Q_{sk}, & 0 < Q_{sk} \leq BP_{s1}, \\ P_{s3} * Q_{sk}, & BP_{s1} < Q_{sk} \leq BP_{s2}, \\ \vdots & \vdots \\ P_{sl_s} * Q_{sk}, & BP_{sl_s - 1} < Q_{sk} \leq BP_{sl_s}, \\ P_{s|l_s|} * Q_{sk}, & BP_{s|l_s|-1} < Q_{sk} \leq BP_{s|l_s|}, \end{cases} \tag{1}$$

where $P_{s1} > P_{s2} > \ldots > P_s \mid l_s \mid$. Figure 2 demonstrates an example of total cost according to all-unit discount and piecewise linear cost function.

The following questions are asked to formulate the problem.

1. Where should shrimp companies be established among the potential places?
2. Which of the modes of transportation should be used to transfer input from suppliers to companies and from companies to consumers?
3. How much input should be purchased from suppliers?

To answer the above questions, the mixed integer programming method is used, which shows the general form of the MIP model in equation (2):

$$\begin{array}{ll} \min & c_x^t + d_y^t \\ \text{s.t.} & Ax + By \leq b \\ & x \geq 0, \quad x \in X \subset \Re^n, \quad y \in \{0,1\}^q \end{array} \tag{2}$$

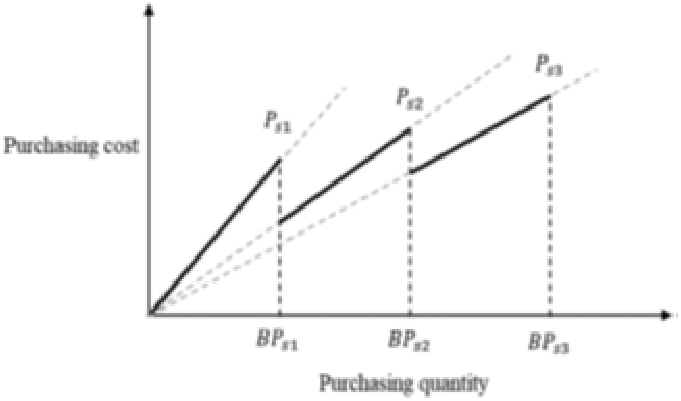


Figure 2: All-unit discount cost function.

where x is a vector of n continuous variables, y is a vector of q variables of 0 and 1, c and d are parameter vectors, A and B are coefficient matrices with proportional dimensions and b is the source vector. The objective function of the present problem is to minimize the total costs of the shrimp supply chain, including fixed and variable costs of constructing ponds and startup companies, fixed and variable costs of transportation modes, and the cost of purchasing inputs. Also, there are 20 limitations in the model that guarantee the assumptions of the problem. We note that we represent parameters in the model in upper case and decision variables in lower case. The following notation is used for our proposed model:

- **Sets:**

  I: The set of consumers indexed by $i = 1, 2, \ldots, |I|$,

  K: The set of potential companies indexed by $k = 1, 2, \ldots, |K|$,

  S: The set of potential suppliers indexed by $s = 1, 2, \ldots, |S|$,

  l_s: The set of price break-down points for each supplier s indexed by $l_s = 1, 2, \ldots, |L_s|$,

  TM: The set of transportation modes indexed by $m = 1, 2$.

- **Parameters:**

  D_i: Product demand from consumer i,

  C_k^2: Maximum capacity of company k,

  C_s^1: Maximum capacity of supplier s for raw material,

  $N\max$: Maximum number of companies which can be established,

  U: Utilization rate of raw material per unit of the product,

  F_k: Annual fixed cost for opening and operating in company k,

CV^2: Upper limit for changing transportation mode in the second echelon (company-consumer),

CV^1: Upper limit for changing transportation mode in the first echelon (supplier-company),

C^2_{km}: Fixed cost of providing transportation mode m for each company k,

C^1_{sm}: Fixed cost of providing transportation mode m in each supplier s,

G_{kim}: Variable cost of products transportation from company k to consumer i via transportation mode m,

P_{sls}: Unit purchasing and transportation cost of raw material in supplier s and l_s^{th} price interval,

BP_{sls}: Quantity of break-down point ls in supplier s,

M: A big real positive number.

- **Decision variables:**

$z_k = 1$ if company k is established, and 0 otherwise.

$r_{ki} = 1$ if company k serves consumer i, and 0 otherwise.

$V^1_{skm} = 1$ if transportation mode m is used for transporting raw material from supplier s to company k, and 0 otherwise.

$V^2_{kim} = 1$ if transportation mode m is used for transporting products from company k to consumer i, and 0 otherwise.

$y_{skls} = 1$ if purchasing and transportation price of raw materials from supplier s to company k is in the l_s^{th} price interval of supplier s except the $(|L_s|)^{th}$ interval due to the problem's assumptions, and 0 otherwise.

We have $|l_s|$ breakdown points. Then, the total number of intervals would be $|l_s| - 1$ and we should assume $y_{sk|l_s|} = 0$ because of this fact.

$X_{skls} \in [0, 1]$ continuous variable between 0 and 1 for determining purchased raw material from supplier s to company k in the l_s^{th} price interval of supplier s.

$tc_{skls} \geq 0$ continuous variable for determining the total purchasing and transportation cost of raw material shipped from supplier s to company k whose quantity falls within the l_s^{th} and $(l_s - 1)^{th}$ breakdown points of supplier s. $l_s \in L_S$ shows price breakdown points of supplier s; therefore, the number of related price intervals is l_s -1 and the quantity of the first break-down point and its related costs is zero.

We present our MIP model for the two-echelon supply chain network design with transportation mode selection and all-unit quantity discount as follows:

$$\begin{aligned} \min \quad & \sum_{k} F_k Z_k + \sum_{k}\sum_{i}\sum_{m} (A^2_{km} + G_{kim} D_i) V^2_{kim} \\ & + \sum_{s}\sum_{k}\sum_{m} A^1_{sm} V^1_{sm} + \sum_{s}\sum_{k}\sum_{l_s} tc_{skl_s} \end{aligned} \tag{3}$$

$$\text{s.t.} \quad \sum_{k} r_{ki} = 1 \quad \forall i = 1 \tag{4}$$

$$U \sum_{I} D_i r_{ki} \leq C^2_k Z_k \quad \forall k \in K \tag{5}$$

$$U \sum_{I} D_i r_{ki} \leq \sum_{S}\sum_{l_s} BP_{sls} X_{skls} \quad \forall k \in K \tag{6}$$

$$\sum_{k}\sum_{l_s} BP_{sls} X_{skls} \leq C^1_s \quad \forall s \in S \tag{7}$$

$$\sum_{k} Z_k \leq N_{\max} \tag{8}$$

$$\begin{aligned} & tc_{sls} \geq (BP_{sls} X_{skls} + BP_{s,ls-1} X_{skls-1}) P_{sls} \\ & \quad - M(1 - y_{skls-1}) \forall s \in S, k \in K, l_s \in L_S | l_s > 1 \end{aligned} \tag{9}$$

$$X_{skls} \leq y_{skls} \quad \forall s \in S, k \in K, l_s \in L_S \quad | l_s = 1 \tag{10}$$

$$X_{skls} \leq y_{skls} + y_{skls-1} \quad \forall s \in S, k \in K, l_s \in L_S 1 < l_s < |l_s| \tag{11}$$

$$X_{skls} \leq y_{skls-1} \quad \forall s \in S, k \in K, l_s \in L_S \quad | l_s = |l_s| \tag{12}$$

$$\sum_{l_s=1}^{|l_s|-1} y_{skls-1} \quad \forall s \in S, k \in K \tag{13}$$

$$\sum_{l_s} X_{skls} = 1 \quad \forall s \in S, k \in K \tag{14}$$

$$\sum_{l_s} BP_{sls} X_{skls} \leq M \sum_{m=1}^{2} V^1_{skm} \quad \forall s \in S, k \in K \tag{15}$$

$$\sum_{l_s} BP_{sls} X_{skls} - CV^1 \leq M V^1_{skm} \quad \forall s \in S, k \in K, m = 2 \tag{16}$$

$$\sum_{m} V^1_{skm} \leq 1 \quad \forall s \in S, k \in K \tag{17}$$

$$D_i r_{ki} \leq M \sum_{m=1}^{2} V^2_{kim} \quad \forall k \in K, i \in I \tag{18}$$

$$CV^2 - D_i r_{ki} \leq M(1 - V^2_{kim}) \quad \forall k \in K, i \in I, m = 2 \tag{19}$$

$$\sum_{m} V^2_{kim} \leq 1 \quad \forall k \in K, i \in I \tag{20}$$

$$\sum_{l_s} tc_{skl_s} \geq 0 \quad \forall s \in S, k \in K, l_s \in L_S \tag{21}$$

$$X_{skl_s} \in [0,1] \quad \forall s \in S, k \in K, l_s \in L_S \tag{22}$$

$$Z_K, r_{ki}, V^2_{kim}, V^1_{skm}, y_{skls} \in 0,1 \forall k \in K, i \in I, m \in TM, l_s \in L_S \tag{23}$$

Function (3) shows the objective function of the model whereby minimizes the total costs of companies including fixed opening and operating, fixed and variable costs of transportation, and purchasing costs of raw materials. Constraint (4) indicates that each consumer must be assigned to one of the companies. Constraint (5) ensures that demand allocation does not exceed the capacity limits. Constraint (6) ensures that the total output of each company is less than its total input. Constraint (7) ensures that buying raw materials does not exceed the capacity limits. Constraint (8) ensures that the number of open facilities does not exceed $N_{\max}$. Constraint (9) computes the total purchasing and transportation cost of raw material shipped from supplier s to company k whose quantity has been located between l_s^{th} and $(l_s - 1)^{th}$ breakdown points. $tc_{sk1} = 0$ because the first break-down point is always zero. Constraint sets (10)-(14) determine the price interval of the purchased raw materials. The way price intervals are calculated is by the piecewise linear function and the linear composition method. More specifically, constraints (10)–(12) determine the portion of each breakdown point in the quantity of the shipped raw materials and constraint. Constraints (10)–(12) refer to the first and last breakdown points where there aren't any other defined price intervals before and after them, respectively (l_s starts from 1 which refers to the first breakdown point in X_{skls} or the first interval in y_{skls}. So, we have | l_s | -1 intervals in total). Constraint (11) covers the intervals between the first and last break-down points. Constraint (13) ensures that this quantity can only belong to one interval. Constraint (14) ensures that the sum of the portions related to the beginning and the end points of each interval must be 1 according to the linear composition method. Constraint (15) ensures that only one of the transportation modes can be utilized to connect the company to the consumer. Constraint (16) ensures that the first mode of transportation cannot be used when the total quantity of shipped raw material from a supplier to a company is less than CV^1. Constraint (17) ensures that only one of the transportation modes can be utilized to connect a company to a consumer. Constraint (18) ensures that only one of the transportation modes can be utilized to connect a supplier to a company. Constraint (19) ensures that the first mode of transportation cannot be used when the total quantity of shipped products from a company to a consumer is less than CV^2. Constraint (20) ensures that only one of the transportation modes can be utilized to connect the supplier to the company. Finally, constraint sets (21) to (23) place restrictions on the nature of our variables. The complexity of our MIP model in the presence of binary variables for the transportation model selection and order quantity is in the order of $O(\max[KIM, SKLs])$. The exclusion of these two factors from our MIP model leads to a model with the size complexity of $O(\max[KI, SK])$. A higher number of binary variables makes a MIP model more difficult to solve due to the number of branching operations required to ensure their integrality. Thus, our problem, in the presence of these two new practical features, is a much more difficult optimization problem than when we do not consider them.

4 Computational Results

As mentioned before, MIP can solve only small instances of the problem in some cases. We need substantially higher computational times to solve the MIP model on the medium and large instances of the problem because the size of the MIP grows quickly as the number of binary and continuous variables in the model increases (i.e., up to 2n branching operations are needed for n binary variables). Since the 1970s, researchers have realized that the complexity of many hard optimization problems can be mitigated if a few complicating constraints are removed and their satisfaction is separately ensured by other heuristic or optimization techniques. The removed set of complicating constraints is therefore dualized, producing a Lagrangian problem that is mostly easy to solve. Therefore, in our study, we used CPLEX solver and GAMS software to solve the MIP model (3 consumers (i), 13 companies (k), and 2 suppliers (s)), first. The computational time is 0.08 seconds. The result showed high-quality solutions that can be found within fractions of the time needed when commercial optimization software was applied directly to the MIP model. In the following, consumers' demand and capacities of all facilities are given in Table 1.

Table 1: Consumer demands and capacities of all facilities

i	D_i (Ton)	k	F_k (Million Rial)	k	C_k^2 (Ton)	S	C_s^1 (Ton)
1	5	1	1106	1	948	1	196
2	35	2	1250	2	812	2	248
3	102	3	1103	3	474		
		4	1117	4	550		
		5	2150	5	761		
		6	1198	6	842		
		7	1377	7	931		
		8	1450	8	522		
		9	1108	9	500		
		10	1118	10	550		
		11	1123	11	775		
		12	2118	12	560		
		13	1103	13	769		

We consider an upper bound of 3 for the number of companies that can be established and assume that one final product consists of 2 units of raw materials. Each supplier offers 4 breakdown points for its purchasing and transportation costs. Two types of transportation modes can be used for shipping raw materials and final products but the second type of transportation mode which contains discounts is forbidden for them if the quantity is lower than 400 and 200 units respectively. The cost of providing these transportation modes is given in Table 2.

Finally, unit purchasing and transportation cost of raw materials and variable cost of product transportation are given in Tables 3 and 4.

The results of the model show that to achieve the minimum cost, companies with numbers 1, 2, 7, 9, 11, and 12 must serve the consumers of the group (α) and (β) and

Table 2: Fixed cost of providing each transportation mode in companies and suppliers (Million Rials)

Transportation mode (m)		1	2
Company (k)	1	130	150
	2	125	148
	3	114	140
	4	112.5	132.5
	5	50	73
	6	125	151
	7	62.5	82.5
	8	62.5	85.5
	9	100	126
	10	125	145
	11	100	123
	12	125	151
	13	100	120
Supplier (s)	1	1256	1600
	2	1400	1100

Table 3: Breakdown points in each supplier and unit purchasing and transportation cost

Breakdown Points (l_s)		1	2	3	4
Supplier 1	BP_{sl_s}	0	230	460	751
	P_{sl_s}	0	17	11	4
Supplier 2	BP_{sl_s}	0	430	517	920
	P_{sl_s}	0	13	5	2

Table 4: Variable cost of transportation from company k to consumer i via mode m (Million Rials)

company (k)	Consumer 1 (α)		Consumer 2 (β)		Consumer 3 (γ)	
	$m = 1$	$m = 2$	$m = 1$	$m = 2$	$m = 1$	$m = 2$
1	0.8	1.2	1.5	2.1	15.6	17.8
2	0.6	0.9	1	1.8	11.8	14.3
3	0.6	0.9	1	1.8	11.8	14.3
4	0.4	0.7	0.8	1.3	9.8	13.8
5	0.5	0.8	0.9	1.9	10	14
6	0.8	1.2	1.5	2.1	15.6	17.8
7	0.8	1.2	1.5	2.1	15.6	17.8
8	0.7	0.9	1.2	2	11.1	17
9	0.8	1.2	1.5	2.1	15.6	17.8
10	0.6	0.9	1	1.8	11.8	14.3
11	0.5	0.8	0.9	1.9	10	14
12	0.8	1.2	1.5	2.1	15.6	17.8
13	0.4	0.7	0.8	1.3	9.8	13.8

other companies serve the consumers of the group (γ). In addition, all products should be carried from companies to consumers by transport mode 1 except when carrying products from companies with numbers 3, 5, and 10 to consumers of the group (γ).

Also, all companies receive their inputs through transport mode 1, and only companies 5 and 10 receive their required inputs from supplier 2 and through transport mode 2. Table 5 summarizes the results of the MIP model.

Table 5: The amount of supplied demand by shrimp farming companies in Golestan province

company (k)	The demand of target market (percentage)		
	α	β	γ
1	0	0.9	0
2	54	0	0
3	0	0	0.002
4	0	0	0.002
5	0	0	0.002
6	0	0	0.007
7	0	0.8	0
8	0	0	0.007
9	23	0	0
10	0	0	0.023
11	23	0	0
12	0	0.3	0
13	0	0	0.01
Total	100	2	0.053

As Table 5 shows, shrimp farming companies in Golestan province will be able to supply 100 percent of the needs in the province by implementing the MIP model. In addition, 2 percent of the products will be sent to the domestic market to supply the demand of other provinces. Also, 85 percent of the province's products are exported, which supplies 0.053 percent of global market demand. It should be noted that currently the companies under study supply 100, 1.48, and 0.03 percent of the needs of target markets, respectively. Also, with the implementation of this model, the total costs of the supply chain will reach 125874.100 million Rials, which is nearly 20 percent less than the current cost.

5 Conclusions and Future Works

In recent years, due to the increase in production of the fisheries sector, international markets, and changes in customers' desires, the seafood business has been astoundingly developed. In many countries, seafood constitutes the most critical part of people's daily diet. Shrimp products are desirable seafood among many populations, and represent a significant amount of food intake in different societies. Shrimp products are either caught in a marine environment like seas and rivers or farmed in aquaculture systems. So, designing a proper supply chain network for shrimp production can offer many benefits for decision-makers, organizations, factories, or even markets to improve the functionality of the supply chain. Thus, this paper introduced a mathematical model for the shrimp supply chain to retrieve the desirable goals of optimizing the total cost of the whole network. To solve this problem, we developed a mixed-integer program, which was able to solve our small problem to optimality. To evaluate this model, two performance measures were considered: the optimality gap (GAP) and the relative percentage deviation (RPD). In this paper, the mean of GAP and PRD were obtained as zero, which indicated the proper performance of the MIP model in finding the feasible solution to the problem. In this paper, we considered a two-echelon supply chain, where level (1) was between the suppliers and the producers and level (2) was between the producers and the consumers; so we prefer that future researches consider other sectors like distribution centers, wholesalers, retailers and so on. Also, future research may need to cover social, environmental, and economical aspects and include them in terms of constraints in the model. Moreover, in real-world settings uncertainty and ambiguity are common for different aspects of the supply chain network especially the demand of markets. For future considerations, the model can be formulated as a stochastic model under the uncertain conditions of demands and other important parameters. Finally, the performance of other exact techniques, including the logic-based Benders decomposition algorithm can be examined for the current work.

References

[1] Adeli, A. (2019). “Strategies for Iran’s fisheries economy ”, Utilization and Cultivation of Aquatics, 8(3), 21-30.

[2] Ahumada, O., Villalobos J. R. (2009). “Applications of planning models in the agri-food supply chain: a review ”, European Journal of Operational Research, 195, 1-20.

[3] Alamdar, S.F., Rabbani, M., Heydari, J. (2018). “Pricing, collection, and effort decisions with coordination contracts in a fuzzy, three-level closed-loop supply chain”, Expert Systems with Applications, 104, 261-276.

[4] Ayoughi, H., Dehghani Poudeh, H., Raad, A., Talebi, D. (2022). “A hybrid heuristic algorithm to provide a multi-objective fuzzy supply chain model with a passive defense approach”, Control and Optimization in Applied Mathematics (COAM), online, 10/30473.

[5] Daneshvar Ameri, J., Salami, H. (2005). “Productivity in shrimp farms: A case study of Bushehr province”, Iranian Journal of Agricultural Sciences, 11(2), 3-13.

[6] Fahimi, K., Seyedhosseini, S.M., Makui, A. (2017). “Simultaneous competitive supply chain network design with continuous attractiveness variables”, Computers and Industrial Engineering, 107, 235-250.

[7] Ghasem Zade, A. (2013). “Evaluation of technical performance of shrimp farms in Boushehr province”, M.A Thesis on Public law, Islamic Azad University.

[8] Govindan, K., Soleimani, H. (2017). “A review of reverse logistics and closed-loop supply chains: A Journal of Cleaner Production focus”, Cleaner Production, 142, 371-384.

[9] Hekmat Shoar, M., Banaderakhshan, R., Asghari, A., Asgari, S., Madani, V. (2010). “Areas of aquaculture investment”, Iran Fisheries Organization, Deputy of Aquaculture, 1-28.

[10] Iran Fisheries Organization. (2018). “Statistical yearbook of Iran fisheries”, Deputy of Planning and Management Development, Planning and Budget Office, page 65.

[11] Kheirabadi, M., Naderi, B., Arshadikhamesh, A., Roshanaei, V. (2019). “A mixed-integer program and a Lagrangian-based decomposition algorithm for the supply chain network design with quantity discount and transportation modes”, Export Systems with Applications, 137, 504-516.

[12] Khorsandifar, S., Feghhi Farahman, N. (2013). “Use of multiple criteria decision-making to study and determine the most attractive target market for exporting agriculture products case study of walnut products”, Industrial Strategic Management, 9(28), 39-56.

[13] Maghsoodi Barmi, M. (2021). “Evaluating technical, allocative and economic efficiency of shrimp production and its effective factors in Golestan province”, M.A. Thesis, Department of Natural Resources and Environmental Economics, Gorgan University of Agricultural Sciences and Natural Resources, page 125.

[14] Mason, N. Flores, H. Villalobos, J. R., Ahumada, O. (2015). “Planning the planting, harvest, and distribution of fresh horticultural products”, In: Plà-Aragonés, L.M.: Handbook of Operations Research in Agriculture and the Agri-Food Industry, Springer Science, New York, Chapter 3, 19-54.

[15] Melo, M., Nickel, S., Saldanhada Gama, F. (2009). “Facility location and supply chain management: A review”, Operational Research, 196, 401-412.

[16] Mosallanezhad, B., Hajiaghaei-Keshteli, M., Triki, Ch. (2021). “Shrimp closed-loop supply chain network design”, Soft Computing, 25, 7399-7422.

[17] Pan, F., Nagi, R. (2013). “Multi-echelon supply chain network design in agile manufacturing”, Omega, 4, 969-983.

[18] Ramezanian, R., Behboodi, Z. (2017). “Blood supply chain network design under uncertainties in supply and demand considering social aspects”, Transportation Research Part E: Logistics and Transportation Review, 104, 69-82.

[19] Rezapour, S., Farahani Zanjirani, R., Pourakbar, M. (2017). “Resilient supply chain network design under competition: A case study”, Operational Research, 259, 1017-1035.

[20] Salehi, M., Jabarpour, E. (2020). “Modeling and solving a multi-objective location-routing problem considering the evacuation of casualties and homeless people and fuzzy paths in relief logistics”, Control and Optimization in Applied Mathematics (COAM), 5, 41-65.

[21] Sampat, A.M., Martin, E., Martin, M., Zavala, V.M. (2017). “Optimization formulations for multi-product supply chain networks”, Computers and Chemical Engineering, 104, 296-310.

[22] Steadieseifi M., Dellaert N. P., Nuijten W., Van Woensel T., Raoufi R. (2014). “Multimodal freight transportation planning: A literature review”, Operational Research, 233, 1-15.

[23] Tabrizi, S., Ghodsypour, S. H., Ahmadi, A. (2017). “A bi-level optimization modeling for perishable food supply chain: The case of a warm-water farmed fish supply chain in Iran”, Trade Studies (IJTS), 21, 169-204.

[24] Tosarkani, B.M., Amin, S.H. (2018). “A possibility solution to configure a battery closed-loop supply chain: Multi-objective approach”, Expert Systems with Applications, 92, 12-26.

[25] Varsei, M., Polyakovskiy, S. (2017). “Sustainable supply chain network design: A case of the wine industry in Australia”, Omega, 66, 236-247.

[26] Zhao, X., Lv, Q. (2011). “Optimal design of agri-food chain network: an improved particle swarm optimization approach”, International Conference on Management and Service Science, 10/1109.

How to Cite this Article:
Mahdavi Varaki, J., Mahdavi, I., Mirkarimi Sh. (2022).“Application of the Mixed-Integer Programming Method in Fishery Supply Chain Network Management: A Case Study of Shrimp in Golestan Province”, Control and Optimization in Applied Mathematics, 7(2): 99-114. doi: 10.30473/coam.2022.63182.1197

Payame Noor University

Control and Optimization in Applied Mathematics (COAM)
Vol. 7, No. 2, Summer-Autumn 2022 (115-130), ©2016 Payame Noor University, Iran

DOI. 10.30473/coam.2022.62157.1186 (Cited this article)

Research Article

A Cramer Method for Solving Fully Fuzzy Linear Systems Based on Transmission Average

Fatemeh Babakordi[1,*], **Tofigh Allahviranloo**[2]

[1]Department of Mathematics and Statistics, Gonbad Kavous University, Gonbad Kavous, Iran.
[2]Faculty of Humanities and Social Sciences, Istinye University, Istanbul, Turkey.

Received: November 17, 2022; **Accepted:** December 17, 2022.

Abstract. Solving fuzzy linear systems has been widely studied during the last decades. However, there are still many challenges to solving fuzzy linear equations, as most of the studies have used the principle of extension, which suffers from shortcomings such as the lack of solution, achieving solutions under very strong conditions, large support of the obtained solutions, inaccurate or even incorrect solutions due to not utilizing all the available information, complicated process and high computational load. These problems motivated us to present a fuzzy Cramer method for solving fuzzy linear equations, which uses arithmetic operations based on the Transmission Average (TA). In this study, fully fuzzy linear systems in the form of $\tilde{A}\tilde{X} = \tilde{B}$, and dual fuzzy linear systems in the form of $\tilde{A}\tilde{X} + \tilde{B} = \tilde{C}\tilde{X} + \tilde{D}$ are solved using the proposed fuzzy Cramer method, and numerical examples are provided to confirm the effectiveness and applicability of the proposed method.

Keywords. Transmission average, Fuzzy arithmetic, Fuzzy approximation, Fuzzy Cramer method.

MSC. 65F05; 15B15; 94D05;03B52.

[*] Corresponding author
babakordif@yahoo.com, Allahviranloo@yahoo.com

1 Introduction

Fuzzy numbers are often used to represent and calculate parameter uncertainties in the procedure of mathematical modeling. Therefore, the analysis and calculation of linear systems with fuzzy numbers are principal in fuzzy mathematics. In recent decades, extensive research has been done in the field of fuzzy mathematics and its utilizations [5, 6, 10, 13, 15, 20, 24, 27, 32, 33, 34].

Friedman et al. have presented an embedding approach to solving general fuzzy linear systems [22]. Asady et al. have investigated solving general fuzzy linear systems and have developed a method to solve an $m \times n$ fuzzy linear system [16]. Iterative methods have been proposed to solve fuzzy systems in [2, 3, 7, 8].

The method of Buckley and Qu has been extended to fuzzy systems in the form of $A_1X + b_1 = A_2X + b_2$, where A_1, A_2, b_1 and b_2 are fuzzy matrices with fuzzy numbers. The classical solution seeks a fuzzy vector X that fulfills the system equation providing the exact equality between the fuzzy vectors X and b. In general, the solution to the system $A_1X + b_1 = A_2X + b_2$ is not identical with that of the system $AX = b$. However, in the case where the matrix $A = A_1 - A_2$ is non-singular, their solutions are identical. Consequently, the system $A_1X + b_1 = A_2X + b_2$ has been transformed into the fully fuzzy linear system $AX = b$, where $A = A_1 - A_2$ and $b = b_2 - b_1$, and has been solved using a new algorithm [19, 29]. In addition, a nonlinear programming method has been utilized to solve fuzzy linear systems [30].

In 2012, the algebraic solution of fuzzy linear systems has been investigated based on interval theory [11]. In [14, 17, 18], fuzzy systems have been solved using linear programming problems, and in [21, 25, 28], fuzzy system-solving methods with the input of complex numbers have been proposed.
Recently, Abbasi et al. have defined new arithmetic operations for fuzzy numbers [4]. Then, fuzzy equations have been solved using these defined arithmetic operations [1, 12].

Solving fuzzy linear systems has been extensively studied in recent decades, and many researchers have utilized the conventional extension principle. This principle defines the standard fuzzy arithmetic, which can lead to inaccurate solutions since it does not consider all the accessible information. Despite reasonable solutions for these methods, they are sometimes complicated with numerous and long techniques and considerable computation. These challenges and problems in solving fuzzy linear systems motivated us to propose a more efficient method. For this purpose, we have studied solving fully fuzzy and dual fuzzy linear systems using Transmission-Average (TA)-based fuzzy operations proposed in [4] and have proposed an analytical Cramer method to solve these systems, which is a more effective method compared to common methods and requires less computation.

The structure of the paper is as follows. In Section 2, the required preliminaries are presented. In Section 3, the fuzzy Cramer method is used to solve systems of the form $\tilde{A}\tilde{X} = \tilde{B}$ and $\tilde{A}\tilde{X} + \tilde{B} = \tilde{C}\tilde{X} + \tilde{D}$. In Section 4, numerical examples are presented to show the effectiveness of the proposed method. Finally, the conclusion ends the paper in Section 5.

2 Basic Definitions

Definition 1. [23] Let A be a fuzzy set in $R\,(A = \{(x, \mu_A(x))\,|x \in R\})$. Then,

i) A is called normal if there exists an $x \in R$ such that $\mu_{\tilde{A}}(x) = 1$. Otherwise, A is subnormal,

ii) The support of A, denoted by supp(A), is the subset of R whose elements all have non-zero membership grades in A. In other words, $\mathrm{supp}(A)\{x \in R|\mu_A(x) > 0\}$,

iii) An α-level set (or α-cut) of a fuzzy set A in R is a non-fuzzy set denoted by A_α and defined by

$$A_\alpha = \begin{cases} \{x \in R \mid \mu_{\tilde{A}}(x) > 0\}, & \alpha > 0, \\ \mathrm{cl}\,(\mathrm{supp}(A)), & \alpha = 0, \end{cases} \tag{1}$$

where $\mathrm{cl}\,(\mathrm{supp}(A))$ denotes the closure of the support of A.

Definition 2. Let $\tilde{A}$ be a Normal, Convex, and Continuous (NCC) fuzzy set on the universal set U. Then, it can be defined from [26]:

$$ac\left(\tilde{A}\right) = \frac{1}{2}\left(\min core\left(\tilde{A}\right) + \max core\left(\tilde{A}\right)\right).$$

Definition 3. [26] A fuzzy number $\tilde{A}$ is called a pseudo-triangular fuzzy number if its membership function $\mu_{\tilde{A}}(x)$ is given by

$$\mu_{\tilde{A}}(x) = \begin{cases} l_{\tilde{A}}(x), & \underline{a} \leqslant x \leqslant a, \\ r_{\tilde{A}}(x), & a \leqslant x \leqslant \overline{a}, \\ 0, & otherwise, \end{cases}$$

where $l_{\tilde{A}}(x)$ and $r_{\tilde{A}}(x)$ are non-decreasing and non-increasing functions respectively. The pseudo-triangular fuzzy number $\tilde{A}$ is denoted by the quintuplet $\tilde{A} = (\underline{a}, a, \overline{a}, l_{\tilde{A}}(x), r_{\tilde{A}}(x))$, and the triangular fuzzy number by the senary $(\underline{a}, a, \overline{a}, -, -)$.

Definition 4. [26] A fuzzy number $\tilde{A}$ is called a pseudo-trapezoidal fuzzy number if its membership function $\mu_{\tilde{A}}(x)$ is given by

$$\mu_{\tilde{A}}(x) = \begin{cases} l_{\tilde{A}}(x), & \underline{a} \leqslant x \leqslant a_1, \\ 1, & a_1 \leqslant x \leqslant a_2, \\ r_{\tilde{A}}(x), & a_2 \leqslant x \leqslant \overline{a}, \\ 0, & otherwise, \end{cases}$$

where $l_{\tilde{A}}(x)$ and $r_{\tilde{A}}(x)$ are non-decreasing and non-increasing functions, respectively. The pseudo-trapezoidal fuzzy number $\tilde{A}$ is denoted by the senary $\tilde{A} = (\underline{a}, a_1, a_2, \overline{a}, l_{\tilde{A}}(x), r_{\tilde{A}}(x))$, and the trapezoidal fuzzy number $\tilde{A}$ is indicated by the senary $\tilde{A} = (\underline{a}, a_1, a_2, \overline{a}, -, -)$.

Definition 5. [12] Consider two pseudo-triangular fuzzy numbers:

$$\tilde{A} = (\underline{a}, a, \overline{a}, l_{\tilde{A}}(x), r_{\tilde{A}}(x)), \quad \tilde{B} = \left(\underline{b}, b, \overline{b}, l_{\tilde{B}}(x), r_{\tilde{B}}(x)\right),$$

with the following α-cut forms:

$$\tilde{A} = \bigcup_{\alpha} A_\alpha, \quad A_\alpha = \left[\underline{A}_\alpha, \overline{A}_\alpha\right], \quad \tilde{B} = \bigcup_{\alpha} B_\alpha, \quad B_\alpha = \left[\underline{B}_\alpha, \overline{B}_\alpha\right].$$

In what follows, fuzzy arithmetic operations are defined based on TA:

$$\begin{gathered}\tilde{A} + \tilde{B} = \bigcup_{\alpha} \left(\tilde{A} + \tilde{B}\right)_\alpha, \\ \left(\tilde{A} + \tilde{B}\right)_\alpha = \left[\frac{a+b}{2} + \left(\frac{\underline{A}_\alpha + \underline{B}_\alpha}{2}\right), \frac{a+b}{2} + \left(\frac{\overline{A}_\alpha + \overline{B}_\alpha}{2}\right)\right],\end{gathered} \tag{2}$$

$$\begin{gathered}\tilde{A} - \tilde{B} = \bigcup_{\alpha} \left(\tilde{A} - \tilde{B}\right)_\alpha, \\ \left(\tilde{A} - \tilde{B}\right)_\alpha = \left[\frac{a-3b}{2} + \left(\frac{\underline{A}_\alpha + \underline{B}_\alpha}{2}\right), \frac{a-3b}{2} + \left(\frac{\overline{A}_\alpha + \overline{B}_\alpha}{2}\right)\right],\end{gathered} \tag{3}$$

$$\begin{gathered}\tilde{A}.\tilde{B} = \bigcup_{\alpha} \left(\tilde{A}.\tilde{B}\right)_\alpha, \\ \left(\tilde{A}.\tilde{B}\right)_\alpha = \begin{cases} \left[\left(\frac{b}{2}\right)\underline{A}_\alpha + \left(\frac{a}{2}\right)\underline{B}_\alpha + \left(\frac{b}{2}\right)\overline{A}_\alpha + \left(\frac{a}{2}\right)\overline{B}_\alpha\right], & a > 0, b > 0, \\ \left[\left(\frac{b}{2}\right)\overline{A}_\alpha + \left(\frac{a}{2}\right)\underline{B}_\alpha + \left(\frac{b}{2}\right)\underline{A}_\alpha + \left(\frac{a}{2}\right)\overline{B}_\alpha\right], & a > 0, b < 0, \\ \left[\left(\frac{b}{2}\right)\overline{A}_\alpha + \left(\frac{a}{2}\right)\overline{B}_\alpha + \left(\frac{b}{2}\right)\underline{A}_\alpha + \left(\frac{a}{2}\right)\underline{B}_\alpha\right], & a < 0, b < 0, \\ \left[\left(\frac{b}{2}\right)\underline{A}_\alpha + \left(\frac{a}{2}\right)\overline{B}_\alpha + \left(\frac{b}{2}\right)\overline{A}_\alpha + \left(\frac{a}{2}\right)\underline{B}_\alpha\right], & a < 0, b > 0, \end{cases}\end{gathered} \tag{4}$$

$$\tilde{A}^{-1} = \bigcup_{\alpha} \left(\tilde{A}^{-1}\right)_\alpha, \left(\tilde{A}^{-1}\right)_\alpha = \left[\frac{1}{a^2}\underline{A}_\alpha, \frac{1}{a^2}\overline{A}_\alpha\right], \tag{5}$$

$$\begin{gathered}\tilde{A}.\tilde{B}^{-1} = \bigcup_{\alpha} \left(\tilde{A}.\tilde{B}^{-1}\right)_\alpha, \\ \left(\tilde{A}.\tilde{B}^{-1}\right)_\alpha = \begin{cases} \left[\left(\frac{1}{2b}\right)\underline{A}_\alpha + \left(\frac{a}{2b^2}\right)\underline{B}_\alpha + \left(\frac{1}{2b}\right)\overline{A}_\alpha + \left(\frac{a}{2b^2}\right)\overline{B}_\alpha\right], & a > 0, b > 0, \\ \left[\left(\frac{1}{2b}\right)\overline{A}_\alpha + \left(\frac{a}{2b^2}\right)\underline{B}_\alpha + \left(\frac{1}{2b}\right)\underline{A}_\alpha + \left(\frac{a}{2b^2}\right)\overline{B}_\alpha\right], & a > 0, b < 0, \\ \left[\left(\frac{1}{2b}\right)\overline{A}_\alpha + \left(\frac{a}{2b^2}\right)\overline{B}_\alpha + \left(\frac{1}{2b}\right)\underline{A}_\alpha + \left(\frac{a}{2b^2}\right)\underline{B}_\alpha\right], & a < 0, b < 0, \\ \left[\left(\frac{1}{2b}\right)\underline{A}_\alpha + \left(\frac{a}{2b^2}\right)\overline{B}_\alpha + \left(\frac{1}{2b}\right)\overline{A}_\alpha + \left(\frac{a}{2b^2}\right)\underline{B}_\alpha\right], & a < 0, b > 0. \end{cases}\end{gathered} \tag{6}$$

Definition 6. [4] Consider two pseudo-trapezoidal fuzzy numbers:

$$\tilde{A} = (\underline{a}, a_1, a_2, \overline{a}, l_{\tilde{A}}(x), r_{\tilde{A}}(x)), \quad \tilde{B} = \left(\underline{b}, b_1, b_2, \overline{b}, l_{\tilde{B}}(x), r_{\tilde{B}}(x)\right),$$

with the following α-cut forms:

$$\begin{aligned} &\tilde{A} = \bigcup_{\alpha} A_{\alpha}, \quad A_{\alpha} = \left[\underline{A}_{\alpha}, \overline{A}_{\alpha}\right], \quad 0 \leqslant \alpha \leqslant 1, \\ &\tilde{B} = \bigcup_{\alpha} B_{\alpha}, \quad B_{\alpha} = \left[\underline{B}_{\alpha}, \overline{B}_{\alpha}\right], \quad 0 \leqslant \alpha \leqslant 1, \\ &B_1 = [b_1, b_2] \quad A_1 = [a_1, a_2]. \end{aligned}$$

Let

$$\varphi = \frac{a_1 + a_2}{2}, \quad \phi = \frac{b_1 + b_2}{2}.$$

In what follows, fuzzy arithmetic operations are defined based on TA:

$$\begin{aligned} &\tilde{A} + \tilde{B} = \bigcup_{\alpha} \left(\tilde{A} + \tilde{B}\right)_{\alpha}, \\ &\left(\tilde{A} + \tilde{B}\right)_{\alpha} = \left[\frac{\phi + \varphi}{2} + \left(\frac{\underline{A}_{\alpha} + \underline{B}_{\alpha}}{2}\right), \frac{\phi + \varphi}{2} + \left(\frac{\overline{A}_{\alpha} + \overline{B}_{\alpha}}{2}\right)\right], \end{aligned} \tag{7}$$

$$\begin{aligned} &\tilde{A} - \tilde{B} = \bigcup_{\alpha} \left(\tilde{A} - \tilde{B}\right)_{\alpha}, \\ &\left(\tilde{A} - \tilde{B}\right)_{\alpha} = \left[\frac{\phi - 3\varphi}{2} + \left(\frac{\underline{A}_{\alpha} + \underline{B}_{\alpha}}{2}\right), \frac{\phi - 3\varphi}{2} + \left(\frac{\overline{A}_{\alpha} + \overline{B}_{\alpha}}{2}\right)\right], \end{aligned} \tag{8}$$

$$\begin{aligned} &\tilde{A}.\tilde{B} = \bigcup_{\alpha} \left(\tilde{A}.\tilde{B}\right)_{\alpha}, \\ &\left(\tilde{A}.\tilde{B}\right)_{\alpha} = \begin{cases} \left[\left(\frac{\varphi}{2}\right)\underline{A}_{\alpha} + \left(\frac{\phi}{2}\right)\underline{B}_{\alpha} + \left(\frac{\varphi}{2}\right)\overline{A}_{\alpha} + \left(\frac{\phi}{2}\right)\overline{B}_{\alpha}\right], & \phi > 0, \varphi > 0, \\ \left[\left(\frac{\varphi}{2}\right)\overline{A}_{\alpha} + \left(\frac{\phi}{2}\right)\underline{B}_{\alpha} + \left(\frac{\varphi}{2}\right)\underline{A}_{\alpha} + \left(\frac{\phi}{2}\right)\overline{B}_{\alpha}\right], & \phi > 0, \varphi < 0, \\ \left[\left(\frac{\varphi}{2}\right)\overline{A}_{\alpha} + \left(\frac{\phi}{2}\right)\overline{B}_{\alpha} + \left(\frac{\varphi}{2}\right)\underline{A}_{\alpha} + \left(\frac{\phi}{2}\right)\underline{B}_{\alpha}\right], & \phi < 0, \varphi < 0, \\ \left[\left(\frac{\varphi}{2}\right)\underline{A}_{\alpha} + \left(\frac{\phi}{2}\right)\overline{B}_{\alpha} + \left(\frac{\varphi}{2}\right)\overline{A}_{\alpha} + \left(\frac{\phi}{2}\right)\underline{B}_{\alpha}\right], & \phi < 0, \varphi > 0, \end{cases} \end{aligned} \tag{9}$$

$$\tilde{A}^{-1} = \bigcup_{\alpha} \left(\tilde{A}^{-1}\right)_{\alpha}, \left(\tilde{A}^{-1}\right)_{\alpha} = \left[\frac{1}{\phi^2}\underline{A}_{\alpha}, \frac{1}{\phi^2}\overline{A}_{\alpha}\right], \tag{10}$$

$$\tilde{A}.\tilde{B}^{-1} = \bigcup_{\alpha} \left(\tilde{A}.\tilde{B}^{-1}\right)_{\alpha},$$

$$\left(\tilde{A}.\tilde{B}^{-1}\right)_{\alpha} = \begin{cases} \left[\left(\frac{1}{2\varphi}\right)\underline{A}_{\alpha} + \left(\frac{\phi}{2\varphi^2}\right)\underline{B}_{\alpha} + \left(\frac{1}{2\varphi}\right)\overline{A}_{\alpha} + \left(\frac{\phi}{2\varphi^2}\right)\overline{B}_{\alpha}\right], & \phi > 0, \varphi > 0, \\ \left[\left(\frac{1}{2\varphi}\right)\overline{A}_{\alpha} + \left(\frac{\phi}{2\varphi^2}\right)\underline{B}_{\alpha} + \left(\frac{1}{2\varphi}\right)\underline{A}_{\alpha} + \left(\frac{\phi}{2\varphi^2}\right)\overline{B}_{\alpha}\right], & \phi > 0, \varphi < 0, \\ \left[\left(\frac{1}{2\varphi}\right)\overline{A}_{\alpha} + \left(\frac{\phi}{2\varphi^2}\right)\overline{B}_{\alpha} + \left(\frac{1}{2\varphi}\right)\underline{A}_{\alpha} + \left(\frac{\phi}{2\varphi^2}\right)\underline{B}_{\alpha}\right], & \phi < 0, \varphi < 0, \\ \left[\left(\frac{1}{2\varphi}\right)\underline{A}_{\alpha} + \left(\frac{\phi}{2\varphi^2}\right)\overline{B}_{\alpha} + \left(\frac{1}{2\varphi}\right)\overline{A}_{\alpha} + \left(\frac{\phi}{2\varphi^2}\right)\underline{B}_{\alpha}\right], & \phi < 0, \varphi > 0. \end{cases} \tag{11}$$

Definition 7. [4] Let $F_C(R)$ be a set of pseudo-geometric fuzzy numbers defined on a set of real numbers. Then, for each $\tilde{A}$ there exists $0_{\tilde{A}}$ such that

$$\tilde{A} + 0_{\tilde{A}} = 0_{\tilde{A}} + \tilde{A} = \tilde{A}, \quad \tilde{A} - \tilde{A} = 0_{\tilde{A}},$$

for $\tilde{A} = (\underline{a}, a, \overline{a}, l_{\tilde{A}}(x), r_{\tilde{A}}(x))$, we have:

$$0_{\tilde{A}} = (\underline{a} - a, 0, \overline{a} - a, l_{\tilde{A}}(x+a), r_{\tilde{A}}(x+a)), \tag{12}$$

and for $\tilde{A} = (\underline{a}, a_1, a_2, \overline{a}, l_{\tilde{A}}(x), r_{\tilde{A}}(x))$, we have:

$$0_{\tilde{A}} = \left(\underline{a} - \phi, \frac{a_1 - a_2}{2}, \frac{a_2 - a_1}{2}, \overline{a} - \phi, l_{\tilde{A}}(x+\phi), r_{\tilde{A}}(x+\phi)\right). \tag{13}$$

Definition 8. [4] Let $\tilde{A}$ and $\tilde{B}$ be two NCC fuzzy sets. Then,

$$\tilde{A} \cong \tilde{B} \quad \textit{if and only if} \quad ac\left(\tilde{A}\right) = ac\left(\tilde{B}\right).$$

3 The Proposed Method

Definition 9. Let $\tilde{A} = \left[\tilde{a}_{ij}\right]$ and $\tilde{B} = \left[\tilde{b}_{ij}\right]$, $1 \leqslant j,\ j \leqslant n$ be fuzzy matrices. It is said that $\tilde{A} \cong \tilde{B}$, if:

$$\forall\ 1 \leqslant j,\ j \leqslant n,\ ac\left(\tilde{a}_{ij}\right) = ac\left(\tilde{b}_{ij}\right).$$

Definition 10. Let $\tilde{A} = \left[\tilde{a}_{ij}\right]$, $1 \leqslant j,\ j \leqslant n$ be a fuzzy matrix. The corresponding zero matrix is shown by $O_{\tilde{A}}$ and can be defined as follows:

$$O_{\tilde{A}} = \begin{pmatrix} 0_{\tilde{a}_{11}} & 0_{\tilde{a}_{12}} & \cdots & 0_{\tilde{a}_{1n}} \\ 0_{\tilde{a}_{21}} & 0_{\tilde{a}_{22}} & \cdots & 0_{\tilde{a}_{2n}} \\ \vdots & \vdots & \vdots & \vdots \\ 0_{\tilde{a}_{n1}} & 0_{\tilde{a}_{n2}} & \cdots & 0_{\tilde{a}_{nn}} \end{pmatrix},$$

where $0_{\tilde{a}_{ij}}$, $1 \leqslant i,\ j \leqslant n$ is determined based on (12) and (13).

Lemma 1. If $\tilde{A}$, $\tilde{B}$ and $\tilde{C}$ are fuzzy matrices, then:

i. $\tilde{A} - \tilde{A} \cong O_{\tilde{A}}$,

ii. $\tilde{A} + O_{\tilde{A}} \cong \tilde{A}$,

iii. $\tilde{A}.\tilde{B} = \tilde{B}.\tilde{A}$,

iv. $\tilde{A}.\tilde{B} + \tilde{A} + \tilde{C} \cong \tilde{A}.\left(\tilde{B} + \tilde{C}\right)$.

Definition 11. The determinant of a 2×2 fuzzy matrix $\tilde{A} = \begin{pmatrix} \tilde{a}_{11} & \tilde{a}_{12} \\ \tilde{a}_{21} & \tilde{a}_{22} \end{pmatrix}$ is shown by $|\tilde{A}|$ and is defined by

$$|\tilde{A}| = (\tilde{a}_{11}.\tilde{a}_{22}) - (\tilde{a}_{12}.\tilde{a}_{21}). \tag{14}$$

Definition 12. Let $\tilde{A} = \left[\tilde{a}_{ij}\right]_{n \times n}$. The (i,j)-minor of $\tilde{A}$, which is the determinant of the matrix of $\tilde{A}$ formed by deleting the i-th row and j-th column of $\tilde{A}$, is denoted by $\tilde{M}_{ij}$.

Example 1. Consider the following fuzzy matrix:

$$\tilde{A} = \begin{pmatrix} (-4,1,2,-,-) & (5,6,7,-,-) & \left(1,2,4,\left(1-(x-2)^2\right)^{\frac{1}{2}},\left(1-\frac{1}{4}(x-2)^2\right)^{\frac{1}{2}}\right) \\ (1,2,3,-,-) & (6,6,7,-,-) & (1,2,3,-,-) \\ (-4,1,2,-,-) & (4,5,7,-,-) & (-7,2,3,-,-) \end{pmatrix}.$$

The $(3,1)$-minor of $\tilde{A}$ is obtained as:

$$\tilde{M}_{31} = \begin{pmatrix} (5,6,7,-,-) & \left(1,2,4,\left(1,(x-2)^2\right)^{\frac{1}{2}},\left(1-\frac{1}{4}(x-2)^2\right)^{\frac{1}{2}}\right) \\ (6,6,7,-,-) & (1,2,3,-,-) \end{pmatrix}.$$

Definition 13. Consider the fuzzy matrix

$$\tilde{A} = \begin{pmatrix} \tilde{a}_{11} & \tilde{a}_{12} & \cdots & \tilde{a}_{1n} \\ \tilde{a}_{21} & \tilde{a}_{22} & \cdots & \tilde{a}_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ \tilde{a}_{n1} & \tilde{a}_{n2} & \cdots & \tilde{a}_{nn} \end{pmatrix}.$$

The (i,j) element of the cofactor matrix of $\tilde{A}$ is shown by $\tilde{A}_{ij}$ and is defined as follows:

$$\tilde{A}_{ij} = (-1)^{i+j}|\tilde{M}_{ij}|.$$

Example 2. Consider the matrix $\tilde{A}$ of Example 1. Using Definitions 11 to 13 and the TA-based arithmetic operations (2) to (6), it is obtained:

$$\tilde{A}_{13} = \left(-\frac{21}{4},4,8,-,-\right).$$

Definition 14. (Expansion method for calculating the determinant of an $n \times n$ fuzzy matrix). Consider the fuzzy matrix

$$\tilde{A} = \begin{pmatrix} \tilde{a}_{11} & \tilde{a}_{12} & \cdots & \tilde{a}_{1n} \\ \tilde{a}_{21} & \tilde{a}_{22} & \cdots & \tilde{a}_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ \tilde{a}_{n1} & \tilde{a}_{n2} & \cdots & \tilde{a}_{nn} \end{pmatrix}.$$

The determinant of the matrix can be evaluated by expanding each row or column of the matrix. For example, by expanding on the first row, we have:

$$|\tilde{A}| = \tilde{a}_{11}.\tilde{A}_{11} + \tilde{a}_{12}.\tilde{A}_{12} + \cdots + \tilde{a}_{1n}.\tilde{A}_{1n}. \tag{15}$$

Example 3. Consider the matrix $\tilde{A}$ of Example 1. From (15), and (2) to (6), it can be achieved that:

$$|\tilde{A}| = \bigcup_{\alpha} \left[-\frac{60}{16} - \frac{1}{2}\sqrt{1-\alpha^2} + \frac{28}{16}\alpha, \frac{29}{16} - \frac{61}{16}\alpha + \frac{17}{32}\sqrt{1-\alpha^2} \right].$$

Definition 15. The fuzzy matrix $\tilde{A}$ is called singular, if $|\tilde{A}| \cong 0_{|\tilde{A}|}$ and is called non-singular, if $ac\left(|\tilde{A}|\right) \neq 0$.

Definition 16. The following system

$$\begin{pmatrix} \tilde{a}_{11} & \tilde{a}_{12} & \cdots & \tilde{a}_{1n} \\ \tilde{a}_{21} & \tilde{a}_{22} & \cdots & \tilde{a}_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ \tilde{a}_{n1} & \tilde{a}_{n2} & \cdots & \tilde{a}_{nn} \end{pmatrix} \begin{pmatrix} \tilde{x}_1 \\ \tilde{x}_2 \\ \vdots \\ \tilde{x}_n \end{pmatrix} = \begin{pmatrix} \tilde{b}_1 \\ \tilde{b}_2 \\ \vdots \\ \tilde{b}_n \end{pmatrix}, \tag{16}$$

is called a fully fuzzy vector system and is denoted by $\tilde{A}\tilde{X} = \tilde{B}$, where $\tilde{A} = \left[\tilde{a}_{ij}\right]$, $1 \leqslant i$, $j \leqslant n$ is a known $n \times n$ fuzzy matrix, $\tilde{B} = \left[\tilde{b}_i\right]$ is a known $n \times 1$ fuzzy vector, and $\tilde{X} = [\tilde{x}_i]$ is an unknown $n \times 1$ fuzzy vector.

Properties of the fuzzy determinant

- If two rows or two columns of a fuzzy matrix $\tilde{A}$ are equal, then $|\tilde{A}| \cong 0_{|\tilde{A}|}$.
- In a fuzzy matrix $\tilde{A}$, if for ith row and jth column, $j = 1, 2, \ldots, n$, there is $\tilde{a}_{ij} \cong 0_{\tilde{a}_{ij}}$, then $|\tilde{A}| \cong 0_{|\tilde{A}|}$.
- For any fuzzy square matrix $\tilde{A}$, we have $|\tilde{A}| \cong |\tilde{A}^T|$.
- If two rows or two columns of a fuzzy matrix $\tilde{A}$ are switched and the obtained (or resulting) matrix is called $\tilde{B}$, then $|\tilde{B}| \cong |\tilde{A}|$.

- In a fuzzy matrix $\tilde{A} = \left[\tilde{a}_{ij}\right]$, if we have $\tilde{a}_{ij} \cong 0_{\tilde{a}_{ij}}$ for each $i, j = 1, 2, \ldots, n$, then $|\tilde{A}| = [0, 0]$.

Example 4. Consider the fuzzy matrix

$$\tilde{A} = \begin{pmatrix} \left(0,4,6,-,1-\frac{1}{4}(x-4)^2\right) & \left(0,4,6,-,1-\frac{1}{4}(x-4)^2\right) \\ (-3,-2,-1,-,-) & (-3,-2,-1,-,-) \end{pmatrix},$$

in which the first and second columns are equal. From (14), it can be found that

$$|A| = \bigcup_{\alpha} \left[-2 + 2\alpha - 2\sqrt{1-\alpha}, 6 - 6\alpha\right],$$

and as a result, $|\tilde{A}| \cong 0_{|\tilde{A}|}$.

The Fuzzy Cramer method: The solution to the fuzzy system (16) obtained using the fuzzy Cramer method is achieved as follows:

$$\tilde{x}_j = \frac{|\tilde{A}_j|}{|A|}, \qquad j = 1, 2, \ldots, n, \tag{17}$$

in which $\tilde{A}_j$ is determined by substituting $\tilde{B}$ in the jth column of $\tilde{A}$.

Theorem 1. If a fuzzy matrix $\tilde{A}$ is non-singular, then the Cramer method always has a fuzzy solution for the fuzzy system (16).

Definition 17. The following system

$$\begin{pmatrix} \tilde{a}_{11} & \tilde{a}_{12} & \cdots & \tilde{a}_{1n} \\ \tilde{a}_{21} & \tilde{a}_{22} & \cdots & \tilde{a}_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ \tilde{a}_{n1} & \tilde{a}_{n2} & \cdots & \tilde{a}_{nn} \end{pmatrix} \begin{pmatrix} \tilde{x}_1 \\ \tilde{x}_2 \\ \vdots \\ \tilde{x}_n \end{pmatrix} + \begin{pmatrix} \tilde{b}_1 \\ \tilde{b}_2 \\ \vdots \\ \tilde{b}_n \end{pmatrix} = \begin{pmatrix} \tilde{c}_{11} & \tilde{c}_{12} & \cdots & \tilde{c}_{1n} \\ \tilde{c}_{21} & \tilde{c}_{22} & \cdots & \tilde{c}_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ \tilde{c}_{n1} & \tilde{c}_{n2} & \cdots & \tilde{c}_{nn} \end{pmatrix} \begin{pmatrix} \tilde{x}_{11} & \tilde{x}_{12} & \cdots & \tilde{x}_{1n} \\ \tilde{x}_{21} & \tilde{x}_{22} & \cdots & \tilde{x}_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ \tilde{x}_{n1} & \tilde{x}_{n2} & \cdots & \tilde{x}_{nn} \end{pmatrix} + \begin{pmatrix} \tilde{d}_1 \\ \tilde{d}_2 \\ \vdots \\ \tilde{d}_n \end{pmatrix}, \tag{18}$$

is called the dual fully fuzzy system and can be shown as $\tilde{A}\tilde{X}+\tilde{B} = \tilde{C}\tilde{X}+\tilde{D}$ by considering $\tilde{A} = \left[\tilde{a}_{ij}\right]_{n\times n}$, $\tilde{B} = \left[\tilde{b}_i\right]_{n\times 1}$, $\tilde{C} = \left[\tilde{c}_{ij}\right]_{n\times n}$ and $\tilde{D} = \left[\tilde{d}_i\right]_{n\times 1}$. It is assumed that the matrix $A - C = \left[a_{ij}\right] - \left[c_{ij}\right]$ is non-singular.

Now, the goal is to solve $\tilde{A}\tilde{X} + \tilde{B} = \tilde{C}\tilde{X} + \tilde{D}$. Therefore, we have:

$$\tilde{A}\tilde{X} + \tilde{B} - \tilde{B} = \tilde{C}\tilde{X} + \tilde{D} - \tilde{B}.$$

Using Lemma 1, (i) and (ii), we have:

$$\tilde{A}\tilde{X} \cong \tilde{C}\tilde{X} + \tilde{D} - \tilde{B}.$$

Adding $-\tilde{C}\tilde{X}$ to both sides of the above equation and using Lemma 1, (i) and (ii), it is obtained:

$$\tilde{A}\tilde{X} + \left(-\tilde{C}\right)\tilde{X} \cong \tilde{D} - \tilde{B}.$$

Moreover, using Lemma 1,(iv), we achieve:

$$\left(\tilde{A}-\tilde{C}\right)\tilde{X}\cong\tilde{D}-\tilde{B}. \tag{19}$$

Finally, the solution for the fuzzy system (19) is obtained as follows using the fuzzy Cramer method:

$$\tilde{x}_j=\frac{\left|\left(\tilde{A}-\tilde{C}\right)_j\right|}{\left|\left(\tilde{A}-\tilde{C}\right)\right|}\qquad \tilde{x}_j=\bigcup_{\alpha}\left(\tilde{x}_j\right)_\alpha \tag{20}$$

in which $\left(\tilde{A}-\tilde{C}\right)_j$ is determined by substituting the elements of $\tilde{D}-\tilde{B}$ in the j-th column of $\tilde{A}-\tilde{C}$.

Theorem 2. If the matrix $\left(\tilde{A}-\tilde{C}\right)$ is non-singular, the dual fully fuzzy system (20) has a fuzzy solution.

4 Numerical Examples

Example 5. Consider the following system of linear equations.

$$\begin{pmatrix}(-4,1,2,-,-) & (4,7,8,-,-)\\ (2,4,6,-,-) & (6,6,7,-,-)\end{pmatrix}\begin{pmatrix}\tilde{x}\\ \tilde{y}\end{pmatrix}=\begin{pmatrix}(1,5,7,-,-)\\ (1,2,3,-,-)\end{pmatrix}. \tag{21}$$

Using the arithmetic operations (2) to (6), the fuzzy Cramer method (17) and determinant definition (14), we obtain:

$$\tilde{x}=\frac{\left|\begin{pmatrix}(1,5,7,-,-) & (4,7,8,-,-)\\ (1,2,3,-,-) & (6,6,7,-,-)\end{pmatrix}\right|}{\left|\begin{pmatrix}(-4,1,2,-,-) & (4,7,8,-,-)\\ (2,4,6,-,-) & (6,6,7,-,-)\end{pmatrix}\right|}=\left(-\frac{2142}{1936},-\frac{16}{22},-\frac{801}{1936},-,-\right),$$

$$\tilde{y}=\frac{\left|\begin{pmatrix}(-4,1,2,-,-) & (1,5,7,-,-)\\ (2,4,6,-,-) & (1,2,3,-,-)\end{pmatrix}\right|}{\left|\begin{pmatrix}(-4,1,2,-,-) & (4,7,8,-,-)\\ (2,4,6,-,-) & (6,6,7,-,-)\end{pmatrix}\right|}=\left(\frac{1128}{1936},\frac{18}{22},\frac{27445}{21296},-,-\right),$$

which is shown in Figure 1.

Example 6. Consider the following dual fuzzy system:

$$\begin{aligned}&\begin{pmatrix}1 & 2 & -1\\ 3 & 0 & 5\\ -2 & 4 & 1\end{pmatrix}\begin{pmatrix}\tilde{x}_1\\ \tilde{x}_2\\ \tilde{x}_3\end{pmatrix}+\begin{pmatrix}(-2,0,1,1,-,-)\\ \left(1,2,4,6,x-1,\left(1-\frac{1}{4}(x,-4)^2\right)^{\frac{1}{2}}\right)\\ (-2,0,2,4,-,-)\end{pmatrix}=\\ &\begin{pmatrix}2 & 0 & -3\\ 1 & -2 & 0\\ 6 & 1 & -1\end{pmatrix}\begin{pmatrix}\tilde{x}_1\\ \tilde{x}_2\\ \tilde{x}_3\end{pmatrix}+\begin{pmatrix}\left(1,2,7,9,\left(1-(x-2)^2\right)^{\frac{1}{2}},\left(1-\frac{1}{4}(x-7)^2\right)^{\frac{1}{2}}\right)\\ (-3,-2,1,3,-,-)\\ (-2,0,1,1,-,-)\end{pmatrix}.\end{aligned} \tag{22}$$

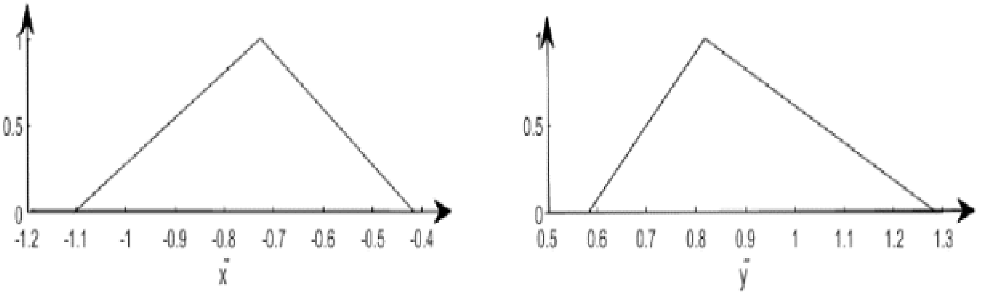


Figure 1: The fuzzy solution to Example 5.

Using (19) and the difference (8), we obtain:

$$\begin{pmatrix} -1 & 2 & 2 \\ 2 & 2 & 5 \\ -8 & 3 & 2 \end{pmatrix} \begin{pmatrix} \tilde{x}_1 \\ \tilde{x}_2 \\ \tilde{3}_3 \end{pmatrix} \cong \begin{pmatrix} \left(\frac{5}{2}, 3, 7, 8, \left(1-(6-2x)^2\right)^{\frac{1}{2}}\right) \\ \left(-5, -\frac{9}{1}, -\frac{3}{2}, -\frac{1}{2}, 2x+10, \left(1-\left(x+\frac{3}{2}\right)^2\right)^{\frac{1}{2}}\right) \\ \left(-\frac{13}{4}, -\frac{5}{4}, \frac{1}{4}, \frac{5}{4}, \frac{x}{2}+\frac{13}{8}, \frac{5}{4}-x\right) \end{pmatrix}.$$

Finally, using the fuzzy Cramer method (17), fuzzy determinant (15), and arithmetic operations (7) to (11), the solution to the dual fuzzy system (22) is obtained as follows:

$$\tilde{x}_1 = \frac{\begin{vmatrix} \left(\frac{5}{2}, 3, 7, 8, \left(1-(6-2x)^2\right)^{\frac{1}{2}}, \left(1-(x-7)^2\right)^{\frac{1}{2}}\right) & 2 & 2 \\ \left(-5, -\frac{9}{2}, -\frac{3}{2}, -\frac{1}{2}, 2x+10, \left(1-\left(x+\frac{3}{2}\right)^2\right)^{\frac{1}{2}}\right) & 2 & 5 \\ \left(-\frac{13}{4}, -\frac{5}{4}, \frac{1}{4}, \frac{5}{4}, \frac{x}{2}+\frac{13}{8}, \frac{5}{4}-x\right) & 3 & 2 \end{vmatrix}}{\begin{vmatrix} -1 & 2 & 2 \\ 2 & 2 & 5 \\ -8 & 3 & 2 \end{vmatrix}}$$

$$= \bigcup_{\alpha} \left[\frac{7523}{4224} + \frac{36}{4224}\alpha + \frac{12}{4224}\sqrt{1-\alpha^2}, \frac{8551}{4224} - \frac{88}{4224}\alpha + \frac{88}{4224}\sqrt{1-\alpha^2}\right],$$

$$\tilde{x}_2 = \frac{\begin{vmatrix} -1 & \left(\frac{5}{2}, 3, 7, 8, \left(1-(6-2x)^2\right)^{\frac{1}{2}}, \left(1-(x-7)^2\right)^{\frac{1}{2}}\right) & 2 \\ 2 & \left(-5, -\frac{9}{2}, -\frac{3}{2}, -\frac{1}{2}, 2x+10, \left(1-\left(x+\frac{3}{2}\right)^2\right)^{\frac{1}{2}}\right) & 5 \\ -8 & \left(-\frac{13}{4}, -\frac{5}{4}, \frac{1}{4}, \frac{5}{4}, \frac{x}{2}+\frac{13}{8}, \frac{5}{4}-x\right) & 2 \end{vmatrix}}{\begin{vmatrix} -1 & 2 & 2 \\ 2 & 2 & 5 \\ -8 & 3 & 2 \end{vmatrix}}$$

$$= \bigcup_{\alpha} \left[-\frac{8069}{8448} + \frac{35}{2112}\alpha - \frac{11}{132}\sqrt{1-\alpha^2}, -\frac{4595}{8448} - \frac{21}{2112}\alpha + \frac{61}{1056}\sqrt{1-\alpha^2} \right],$$

$$\tilde{x}_3 = \frac{\begin{vmatrix} -1 & 2 & \left(\frac{5}{2}, 3, 7, 8, \left(1-(6-2x)^2\right)^{\frac{1}{2}}, \left(1-(x-7)^2\right)^{\frac{1}{2}}\right) \\ 2 & 2 & \left(-5, -\frac{9}{2}, -\frac{3}{2}, -\frac{1}{2}, 2x+10, \left(1-\left(x+\frac{3}{2}\right)^2\right)^{\frac{1}{2}}\right) \\ -8 & 3 & \left(-\frac{13}{4}, -\frac{5}{4}, \frac{1}{4}, \frac{5}{4}, \frac{x}{2}+\frac{13}{8}, \frac{5}{4}-x\right) \end{vmatrix}}{\begin{vmatrix} -1 & 2 & 2 \\ 2 & 2 & 5 \\ -8 & 3 & 2 \end{vmatrix}}$$

$$= \bigcup_{\alpha} \left[-\frac{14367}{4224} + \frac{35}{4224}\alpha - \frac{64}{4224}\sqrt{1-\alpha^2}, -\frac{13936}{4224} - \frac{20}{4224}\alpha + \frac{70}{4224}\sqrt{1-\alpha^2} \right],$$

which is shown in Figure 2.

Example 7. Consider the following fuzzy system:

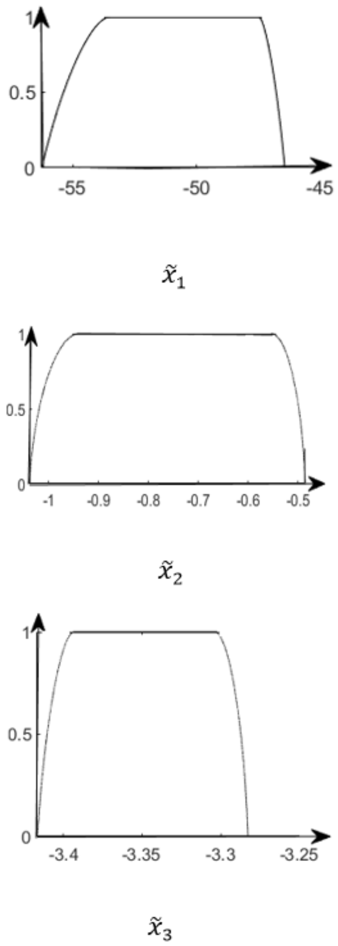
$$\begin{pmatrix} (-1,1,3,-,-) & \left(1,2,4,\left(1-(x-2)^2\right)^{\frac{1}{2}}, \left(1-\frac{1}{4}(x-2)^2\right)^{\frac{1}{2}}\right) & (1,3,6,-,-) \\ (-2,-2,2,-,-) & \left(3,4,5,\left(1-(x-4)^2\right)^{\frac{1}{2}}, 5-x\right) & \left(4,7,9,\frac{1}{3}(x-4), \left(1-\frac{1}{4}(x-7)^2\right)^{\frac{1}{2}}\right) \\ (2,3,5,-,-) & (2,6,8,-,-) & \left(8,9,10,x-8,\left(1-(x-9)^2\right)^{\frac{1}{2}}\right) \end{pmatrix} \begin{pmatrix} \tilde{x}_1 \\ \tilde{x}_2 \\ \tilde{x}_3 \end{pmatrix} \tag{23}$$
$$= \begin{pmatrix} (2,4,7,-,-) \\ \left(1,2,6,\left(1-(x-2)^2\right)^{\frac{1}{2}}, \left(1-\frac{1}{16}(x-2)^2\right)^{\frac{1}{2}}\right) \\ (3,6,8,-,-) \end{pmatrix}.$$

Assuming:

$$\tilde{A} = \begin{pmatrix} (-1,1,3,-,-) & \left(1,2,4,\left(1-(x-2)^2\right)^{\frac{1}{2}}, \left(1-\frac{1}{4}(x-2)^2\right)^{\frac{1}{2}}\right) & (1,3,6,-,-) \\ (-2,-2,2,-,-) & \left(3,4,5,\left(1-(x-4)^2\right)^{\frac{1}{2}}, 5-x\right) & \left(4,7,9,\frac{1}{3}(x-4), \left(1-\frac{1}{4}(x-7)^2\right)\right)^{\frac{1}{2}} \\ (2,3,5,-,-) & (2,6,8,-,-) & \left(8,9,10,x-8,\left(1,(x-9)^2\right)^{\frac{1}{2}}\right) \end{pmatrix},$$

we have $ac\left(|\tilde{A}|\right) = 0$ (because $\begin{vmatrix} 1 & 2 & 3 \\ -2 & 4 & 7 \\ 3 & 6 & 9 \end{vmatrix} = 0$), that is, the coefficient matrix of the system is singular, and the system (23) does not have a fuzzy solution.

Example 8. Suppose you want to calculate the approximate prices of pistachios and almonds in 1390, while you know that one of your colleagues bought about 1 kg ($\tilde{1} = (0,1,2,-,-)$) of pistachios and about 2 kg ($\tilde{2} = (1,2,4,-,-)$) of almonds this year at a price of about 10 tomans ($\tilde{10} = (8,10,11,-,-)$), and your other colleague bought about 3 kg ($\tilde{3} = (2,3,5,-,-)$) of pistachios and about 4 kg ($\tilde{4} = (3,4,7,-,-)$) of almonds



$\tilde{x}_1$

$\tilde{x}_2$

$\tilde{x}_3$

Figure 2: The fuzzy solution to Example 6.

this year at a price of about 24 tomans ($\tilde{24} = (22, 24, 25, -, -)$). (Shopping malls had incentive packages for shoppers.) Now, considering the approximate price of pistachios as $\tilde{x}$ and the approximate price of almonds as $\tilde{y}$, we form the following fuzzy system:

$$\begin{cases} \tilde{1}.\tilde{x} + \tilde{2}.\tilde{y} = \tilde{10}, \\ \tilde{3}.x + \tilde{4}.\tilde{y} = \tilde{24}. \end{cases} \tag{24}$$

Using the fuzzy Cramer method (17), fuzzy determinant (14), and TA-based arithmetic operations (2) to (6), the solution to the fuzzy system (24) is obtained:

$$\tilde{x} = \frac{\begin{vmatrix} (8, 10, 11, -, -) & (1, 2, 4, -, -) \\ (22, 24, 25, -, -) & (3, 4, 7, -, -) \end{vmatrix}}{\begin{vmatrix} (0, 1, 2, -, -) & (1, 2, 4, -, -) \\ (2, 3, 5, -, -) & (3, 4, 7, -, -) \end{vmatrix}} = \left(-\frac{37}{8}, 4, \frac{75}{8}, -, -\right),$$

$$\tilde{y} = \frac{\begin{vmatrix} (0,1,2,-,-) & (8,10,11,-,-) \\ (2,3,5,-,-) & (22,24,25,-,-) \end{vmatrix}}{\begin{vmatrix} (0,1,2,-,-) & (1,2,4,-,-) \\ (2,3,5,-,-) & (3,4,7,-,-) \end{vmatrix}} = \left(-3,3,\frac{15}{2},-,-\right).$$

That is, the price of pistachios was about 4 tomans ($\tilde{4} = \left(-\frac{37}{8},4,\frac{75}{8},-,-\right)$), and the price of almonds was about 3 tomans ($\tilde{3} = \left(-3,3,\frac{15}{2},-,-\right)$).

5 Conclusion

An analytical method for solving a system of fuzzy linear equations is the Cramer method. We can find some limitations in the methods used in the literature. The methods based on arithmetic operations using the extension principle and α-cuts have problems in subtraction and division operations, as well as problems in attaining membership functions for the operators and also the dependence effect in the fuzzy arithmetic operations. Therefore, in this paper, using TA-based fuzzy arithmetic, which is more realistic than other arithmetic operations, we solved a fuzzy system by a Cramer method, which does not have the limitations of the other methods presented by e.g., Allahviranloo et al. or Radhakrishnan et al. In other words, the proposed method was used for all fuzzy systems such as the fully fuzzy and the dual fuzzy systems with all numbers such as quasi-triangular and quasi-trapezoidal numbers as inputs and calculates all the solutions of the fuzzy systems, including non-negative and non-positive solutions. Finally, using the proposed method and assuming that the 1-cut coefficient matrix of the fuzzy system is non-singular, the fuzzy system always contains a fuzzy solution.

References

[1] Abbasai, F., Allahviranloo, T. (2021). "Computational procedure for solving fuzzy equations", Soft Computing, 25, 2701-2703.

[2] Abbasbandy, S., Jafarian, A., Ezzati, R. (2005). "Conjugate gradient method for fuzzy symmetric positive definite system of linear equations", Applied Mathematics and Computation, 171(2), 1184-1191.

[3] Abbasbandy, S., Ezzati, R., Jafarian, A. (2006). "Lu decomposition method for solving fuzzy system of equations", Applied Mathematics and Computation, 172, 633-643.

[4] Abbasi, F., Allahviranloo, T., Abbasbandy, S. (2015). "A new attitude coupled with fuzzy thinking to fuzzy rings and fields", Journal of Intelligent & Fuzzy Systems, 29, 851-861.

[5] Abidin, A.S., Mashadi, M., Sri, G. "Algebraic modification of trapezoidal fuzzy numbers to complete fully fuzzy linear equations system using Gauss-Jacobi method", International Journal of Management and Fuzzy Systems, 5(2), 40-46.

[6] Adabitabar Firozja, M., Babakordi, F., Shahhosseini, M. (2011). “Gauss elimination algorithm for interval matrix”, International Journal of Industrial Mathematics, 3(1), 9-11.

[7] Allahviranloo, T. (2004). “Numerical methods for fuzzy system of linear equations”, Applied Mathematics and Computation, 155, 493-502.

[8] Allahviranloo T. (2005). “Successive over relaxation iterative method for fuzzy system of linear equations”, Applied Mathematics and Computation, 162, 189-196.

[9] Allahviranloo, T., Afshar Kermani, M. (2007). “Cramer’s rule for fuzzy system of equations”, Nonlinear Studies, 14.

[10] Allahviranloo, T., Babakordi, F. (2017). “Algebraic solution of fuzzy linear system as: $\tilde{A}\tilde{X} + \tilde{B}\tilde{X} = \tilde{Y}$”, Soft Computing, 21, 2463-7472.

[11] Allahviranloo, T., Ghanbari, M. (2012). “On the algebraic solution of fuzzy linear systems based on interval theory”, Applied Mathematical Modelling, 36, 5360-5379.

[12] Allahviranloo, T., Perfilieva, I., Abbasi, F. (2018). “A new attitude coupled with fuzzy thinking for solving fuzzy equations”, Soft Computing, 22, 3077-3095.

[13] Allahviranloo, T., Salahshour, S., Khezerloo, M. (2011). “Maximal- and minimal- symmetric solutions of fully fuzzy linear systems”, Journal of Computational and Applied Mathematics, 235, 4652-4662.

[14] Allahviranloo, T., Hosseinzadeh, Lotfi, F., Khorasani Kiasari, M., Khezerloo, M. (2012). “On the fuzzy solution of lr fuzzy linear systems”, Applied Mathematical Modeling, 37(3), 1170-1176.

[15] Araghi, M.A.F., Zarei, E. (2017). “Dynamical control of computations using the iterative methods to solve fully fuzzy linear systems”, Advanced Fuzzy Logic Technologies in Industrial Applications, 641, 55-68.

[16] Asady, B., Abbasbandy, S., Alavi, M. (2005). “Fuzzy general linear systems”, Applied Mathematics and Computation, 169(1), 34-40.

[17] Babakordi, F., Adabitabar, Firozja, M. (2020). “Solving fully fuzzy dual matrix system with optimization problem”, International Journal of Industrial Mathematics, 12(2), 109-119.

[18] Babakordi, F., Allahviranloo, T., Adabitabar Firozja, M. (2015). “An efficient method for solving LR fuzzy dual matrix systems”, Journal of Intelligent and Fuzzy Systems, 30, 575-581.

[19] Buckley, J.J., Qu, Y. (1991). “Solving systems of linear fuzzy equations”, Fuzzy Sets and Systems, 43, 33-43.

[20] Dehghan, M., Hashemi, B., Ghatee, M. (2007). “Solution of the fully fuzzy linear systems using iterative techniques”, Chaos, Solitons & Fractals, 34(2), 316-336.

[21] Farahani, H., Mishmast Nehi, H., Paripour, M. (2016). “Solving fuzzy complex system of linear equations using eigenvalue method”, Journal of Intelligent and Fuzzy Systems, 31(3) 1-11.

[22] Friedman, M., Ming, M., Kandel, A. (1998). “Fuzzy linear systems”, Fuzzy Sets and Systems, 96, 201-209.

[23] Fuller, R. (1998). “Fuzzy reasoning and fuzzy optimization”, On leave from Department of Operations Research, Eötvös Loránd University, Budapest.

[24] Guo, X.B., Shang, D.Q. (2019). “Solving lr fuzzy linear matrix equation”, Iranian Journal of Fuzzy Systems, 16, 33-44.

[25] Jahantigh, M.A., Khezerloo, S., Khezerloo, M. (2010). “Complex fuzzy linear systems”, International Journal of Industrial Mathematics, 2(1), 21-28.

[26] Klir, G.J., Yuan, B. (1995). “Fuzzy sets and fuzzy logic: theory and applications”, Prentice-Hall PTR, Upper Saddlie River.

[27] Landowski, M. (2018). “Method with horizontal fuzzy numbers for solving real fuzzy linear systems”, Soft Computing, 1-13.

[28] Akram, M., Ali, M., Allahviranloo, T. (2022). “A method for solving bipolar fuzzy complex linear systems with real and complex”, Soft computing, 26, 2157–2178.

[29] Muzzioli, S., Reynaerts, H. (2006). “Fuzzy linear system of the form $a_1x + b_1 = a_2x + b_2$”, Fuzzy Sets and Systems, 157, 939-951.

[30] Muzzioli, S., Reynaerts, H. (2007). “A financial application”, European Journal of Operational Research, 177, 1218-1231.

[31] Radhakrishnan, S., Gajivaradhan, P., Govindarajan, R. (2014). “A new and simple method of solving fully fuzzy linearsystem”, Annals of Pure and Applied Mathematics, 8, 1993-1999.

[32] Sankar Prasad, M., Mostafijur, R., Banashree, C., Shariful, A. (2021). “The solution techniques for linear and quadratic equations with coefficients as Cauchy neutrosphic numbers”, Granular Computing, 1-10.

[33] Sevastjanov, P., Dymova, L. (2009). “A new method for solving interval and fuzzy equations: Linear case”, Information Sciences, 179, 925-937.

[34] Zheng, B., Wang, K. (2006). “General fuzzy linear systems”, Applied Mathematics and Computation, 181, 1276-1286.

How to Cite this Article:

Babakordi, F., Allahviranloo, T. (2022). “A cramer method for solving fully fuzzy linear systems based on transmission average”, Control and Optimization in Applied Mathematics, 7(2): 115-130. doi: 10.30473/coam.2022.62157.1186