

Article

An Extreme Learning Machine-Based Method for Solving Linear–Quadratic Nonzero-Sum Differential Games

Changdong Duan and Yuefei Yuan *

School of Automotive Engineering, Chengdu Industry & Trade College, Chengdu 611730, China

* Correspondence: yuanyuefei@cdgmx.edu.cn

Abstract

Multi-agent interaction in linear–quadratic (LQ) differential games gives rise to open-loop Nash equilibria that rarely admit closed-form expressions, motivating the development of reliable numerical solvers. Classical approaches such as shooting and spectral collocation are sensitive to the initial guess on the unknown boundary values and accumulate discretisation error over long horizons, while deep-learning alternatives require iterative gradient-based training with architecture- and convergence-specific overhead. To overcome these limitations, we recast the LQ nonzero-sum game as a linear two-point boundary value problem (TPBVP) via the Pontryagin maximum principle (PMP) and solve it with a single-layer feedforward neural network (SLFN) in which hidden-layer parameters are sampled once and fixed. The state and all player-specific costates are parameterised by random hidden features on a uniform time grid, the boundary conditions are appended as dedicated rows of the linear collocation system, and the output weights follow from a single Moore–Penrose pseudoinverse, entirely bypassing gradient-based iteration. For the scalar LQ optimal-control TPBVP, a residual-to-solution stability theorem converts the continuous equation and boundary residuals into uniform state, costate, control, and cost error bounds. Validation across two-player low- and high-dimensional benchmarks, a heterogeneous three-player game, and paired seed sweeps confirms high accuracy against analytical and matrix-exponential references, while revealing that no single activation function dominates across all problem types: tanh is most accurate in one-dimensional settings, and Gaussian RBF leads in multidimensional cases.

Keywords: linear–quadratic differential game; extreme learning machine; Pontryagin maximum principle; two-point boundary value problem; Nash equilibrium; Moore–Penrose pseudoinverse; numerical collocation method; nonzero-sum game



Academic Editor: Sam Ganzfried

Received: 30 June 2026

Revised: 28 July 2026

Accepted: 31 July 2026

Published: 6 August 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Differential games arise in a wide range of settings that include multi-agent systems (Ge et al., 2026; Ren et al., 2026; Zhang et al., 2025), economics (Clemhout & Wan, 1994; Saltari et al., 2022), and engineering control (Chen et al., 2026; Wang, 2026; Zhuang et al., 2026). Within this class, the linear–quadratic (LQ) nonzero-sum differential game is of particular interest because it models the strategic interaction of several decision makers with quadratic running and terminal costs, and it underlies many formulations in stochastic control and online learning (Wu et al., 2026; Zhang et al., 2024). The solution concept is the open-loop Nash equilibrium (Barghi Jond & Nabiyevev, 2019): for each player i , given that the other players are already at their equilibrium strategies, deviating unilaterally cannot lower player i 's cost. Closed-form expressions for such equilibria are available only in

special cases, so numerical methods are required for almost every problem of practical interest (Nikooeinejad et al., 2016). The Pontryagin maximum principle (PMP) furnishes the necessary conditions for optimality and turns the problem into a two-point boundary value problem (TPBVP) (Engwerda, 2005), yet solving that TPBVP to high accuracy remains non-trivial (Ha & Halme, 1976), and an efficient, reliable numerical scheme for LQ nonzero-sum games is still missing.

Existing numerical approaches fall into two camps, each with a well-known weakness. Deep-learning-based solvers—in particular, physics-informed neural networks and reinforcement-learning policies (Li et al., 2024; Mowlavi & Nabi, 2023; Na & Lee, 2024; Sun et al., 2023; Xu et al., 2022; Zhang et al., 2024a, 2024b)—replace the discretisation by a smooth ansatz and train it with gradient-based optimisation. This eliminates explicit meshing but requires iterative optimisation with method-dependent convergence guarantees and sensitivity to nonconvex loss landscapes. Classical numerical methods such as shooting are fast in low dimensions but highly sensitive to the initial guess on the unknown boundary values (Alomari, 2025; Cole & Sands, 2023; Kashiri et al., 2011) and accumulate discretisation error over long horizons. In contrast, extreme learning machines (ELMs) (Huang et al., 2006) sidestep both difficulties: because the hidden-layer parameters are sampled once and frozen, the output weights follow from a single Moore–Penrose solution without iterative optimisation or an initial guess. ELMs have demonstrated this non-iterative advantage in regression and classification, yet the LQ nonzero-sum TPBVP has, to our knowledge, not been tackled within an ELM framework, and no error bound has been established for such a coupled state–costate approximation.

This paper develops a numerical method for LQ nonzero-sum differential games that preserves the linear PMP structure and avoids gradient-based training. The PMP first converts the game into a coupled linear TPBVP. An SLFN then parameterises the shared state and player-specific costates, while dedicated rows impose the initial and transversality conditions. The output weights follow from one over-determined least-squares solution. For the scalar LQ optimal-control TPBVP, we further prove a residual-to-solution stability estimate: small continuous equation and boundary residuals imply small uniform state, costate, control, and cost errors. This a posteriori result does not impose an unsupported rate in the network width or collocation spacing. The method is evaluated on two-player low- and high-dimensional benchmarks and a heterogeneous three-player benchmark.

We do not claim novelty for the ELM architecture or random-feature least squares. The contribution is their joint application to the shared state and player-specific costates of an LQ nonzero-sum PMP system. We compare the resulting solver with a physics-informed neural network (PINN) (Raissi et al., 2019), contrasting gradient-trained nonlinear residual minimisation with a single linear least-squares solver that preserves the LQ PMP structure.

The main contributions of this paper are as follows:

- **A coupled multi-player random-feature formulation.** The shared state and player-specific costates of the PMP-derived TPBVP are collocated simultaneously, with initial and transversality conditions appended as dedicated rows of the augmented system.
- **Structure-preserving one-step solution of the augmented PMP system.** The constant-coefficient LQ TPBVP, being linear in the output weights, reduces the coupled equations to one over-determined least-squares system, avoiding learning-rate selection and back-propagation while retaining discretisation and conditioning limits of collocation.
- **Scalar residual-stability result.** For the scalar LQ optimal-control TPBVP, an explicit stability estimate bounds the uniform state, costate, and control errors by the continuous equation and boundary residuals; the corresponding cost error has valid first- and second-order terms.

- **Empirical validation across player counts, dimensions, and parameter settings.** The study covers two-player low-/high-dimensional and heterogeneous three-player benchmarks. Repeated-seed experiments against analytical or matrix-exponential references confirm high accuracy, preserving the one-step solution structure; the high-dimensional comparison also includes an FDM reference.

The remainder of this paper is organised as follows. Section 2 presents the nonzero-sum game, the PMP, the ELM model, and the convergence analysis. Section 3 reports the benchmark experiments. Section 4 presents a robustness study and statistical validation, including a sensitivity analysis on the number of hidden neurons and a quantitative check of the boundary-condition residuals. Section 5 concludes and outlines directions for future work.

2. Fundamental Concepts

2.1. Nonzero-Sum Differential Game

Consider a nonzero-sum differential game with $P \geq 2$ players and state dimension d , where the state trajectory is described by

$$\begin{cases} \dot{x}(t) = f(t, x(t), u_i(t), u_{-i}(t)), & t \in (t_0, T]; \\ x(t_0) = x_0. \end{cases} \quad (1)$$

The cost functional of player i over the finite time horizon T is

$$J_i(t, x(t), u_i(t), u_{-i}(t)) = \int_{t_0}^T L_i(t, x(t), u_i(t), u_{-i}(t)) dt + \Phi_i(x(T)), \quad (2)$$

where

$$x : [t_0, T] \rightarrow \mathbb{R}^d, \quad u_i \in U_i,$$

and $u_{-i}(t)$ denotes the stacked column vector of all controls except player i :

$$u_{-i}(t) := \text{col}_{j \neq i} u_j(t) \in \mathbb{R}^{m-i}, \quad m_{-i} := \sum_{j \neq i} m_j, \quad m := \sum_{j=1}^P m_j.$$

Moreover,

$$f : [t_0, T] \times \mathbb{R}^d \times U_i \times U_{-i} \rightarrow \mathbb{R}^d$$

is the rate of change of the state, and x_0 denotes the initial state of $x(t)$. $L_i(\cdot)$ and $\Phi_i(\cdot)$ are the instantaneous and terminal cost functionals, respectively.

Definition 1 (Sadana et al., 2023). Assume all players except i select their optimal strategies $u_{-i}^*(t) = \text{col}_{j \neq i} u_j^*(t)$. For an arbitrary $u_i(t) \in U_i$, the strategy profile $(u_i^*(\cdot), u_{-i}^*(\cdot))$ is a Nash equilibrium if

$$J_i(t, x^*(t), u_i^*(t), u_{-i}^*(t)) \leq J_i(t, x(t), u_i(t), u_{-i}^*(t)), \quad (3)$$

namely, no player i can reduce their cost by unilaterally deviating from the equilibrium strategy. In this case, $(u_i^*(\cdot), u_{-i}^*(\cdot))$ solves System (1).

2.2. Linear–Quadratic Differential Game

We now restrict to the linear–quadratic case, where the dynamics are affine in the state and controls and all cost functionals are quadratic. Specifically, the state dynamics are

$$\begin{aligned}\dot{x}(t) &= Ax(t) + B_i u_i(t) + C_i u_{-i}(t), & t \in (t_0, T]; \\ x(t_0) &= x_0,\end{aligned}\quad (4)$$

with the cost functional of player i :

$$\begin{aligned}J_i(x(t), u_i(t), u_{-i}(t)) &= \frac{1}{2} \int_{t_0}^T \left(x(t)^T R_i x(t) + u_i(t)^T Q_i u_i(t) + u_{-i}(t)^T Q_{-i} u_{-i}(t) \right) dt \\ &\quad + \frac{1}{2} x(T)^T S_i x(T),\end{aligned}\quad (5)$$

where

$$\begin{aligned}x : [t_0, T] &\rightarrow \mathbb{R}^d, \quad u_i \in U_i \subset \mathbb{R}^{m_i}, \quad u_{-i} \in U_{-i} \subset \mathbb{R}^{m_{-i}}, \\ A &\in \mathbb{R}^{d \times d}, \quad B_i \in \mathbb{R}^{d \times m_i}, \quad C_i \in \mathbb{R}^{d \times m_{-i}},\end{aligned}$$

Here, $R_i = R_i^T \succeq 0 \in \mathbb{R}^{d \times d}$, $Q_i = Q_i^T \succ 0 \in \mathbb{R}^{m_i \times m_i}$, and $S_i = S_i^T \succeq 0 \in \mathbb{R}^{d \times d}$ penalise the running state, own control, and terminal state, respectively. Define $Q_{-i}^{(i)}$ as the block-diagonal matrix whose j -th block (for $j \neq i$) is $Q_{ij} = Q_{ij}^T \succeq 0 \in \mathbb{R}^{m_j \times m_j}$, where Q_{ij} weights player j 's control in player i 's cost. Thus, $u_{-i}^T Q_{-i}^{(i)} u_{-i} = \sum_{j \neq i} u_j^T Q_{ij} u_j$; below, Q_{-i} abbreviates $Q_{-i}^{(i)}$. No equality between Q_{-i} and Q_i is assumed unless their dimensions and blocks have been identified explicitly.

2.3. Pontryagin Maximum Principle

To establish the necessary conditions for optimality in a nonzero-sum differential game, the Hamiltonian for player i is defined as

$$H_i(t, x(t), u_i(t), u_{-i}(t), \lambda_i(t)) := L_i(t, x(t), u_i(t), u_{-i}(t)) + \lambda_i^T(t) f(t, x(t), u_i(t), u_{-i}(t)), \quad (6)$$

where $\lambda_i(t)$ is the adjoint variable (also referred to as the Lagrange multiplier) for player i .

Theorem 1 (Engwerda, 2005). *The Pontryagin maximum principle establishes the following necessary conditions for optimality in a nonzero-sum differential game:*

$$\begin{aligned}\dot{x}(t) &= f(t, x(t), u_i(t), u_{-i}(t)), \quad t \in (t_0, T]; \\ x(t_0) &= x_0; \\ \dot{\lambda}_i(t) &= -\frac{\partial H_i}{\partial x}, & t \in [t_0, T]; \\ \lambda_i(T) &= \frac{\partial \Phi_i(x(T))}{\partial x}; \\ \frac{\partial H_i}{\partial u_i} &= 0.\end{aligned}\quad (7)$$

These conditions hold for the optimal control $u_i^*(t)$, the corresponding state trajectory $x^*(t)$, and the adjoint variable $\lambda_i(t)$ for player i .

Corollary 1. *By Theorem 1, the LQ differential game problem can be formulated as the following two-point boundary value problem. We now specialise the Hamiltonian to the LQ setting:*

$$H_i(x(t), \lambda_i(t), u_i(t), u_{-i}(t)) := \lambda_i(t)^T (Ax(t) + B_i u_i(t) + C_i u_{-i}(t)) + \frac{1}{2} x(t)^T R_i x(t) + \frac{1}{2} u_i(t)^T Q_i u_i(t) + \frac{1}{2} u_{-i}(t)^T Q_{-i} u_{-i}(t), \quad (8)$$

where $\lambda_i : [t_0, T] \rightarrow \mathbb{R}^d$. By Theorem 1, we obtain

$$\dot{\lambda}_i(t) = -\frac{\partial H_i}{\partial x} = -R_i x(t) - A^T \lambda_i(t), \quad (9)$$

$$\lambda_i(T) = S_i x(T), \quad (10)$$

$$\frac{\partial H_i}{\partial u_i} = 0 \Rightarrow u_i^*(t) = -Q_i^{-1} B_i^T \lambda_i(t). \quad (11)$$

Here, S_i denotes the terminal cost matrix for player i (reducing to S when all players share the same terminal cost). For the general P -player case, define the stacked opponent input matrix and costate by

$$C_i := [B_1, B_2, \dots, B_{i-1}, B_{i+1}, \dots, B_P] \in \mathbb{R}^{d \times (m-m_i)},$$

$$\lambda_{-i}(t) := (\lambda_1^T(t), \dots, \lambda_{i-1}^T(t), \lambda_{i+1}^T(t), \dots, \lambda_P^T(t))^T \in \mathbb{R}^{d(P-1)}.$$

Here, C_i stacks every opponent's input matrix and λ_{-i} stacks all opponents' costates. Since each opponent satisfies $u_j^*(t) = -Q_j^{-1} B_j^T \lambda_j(t)$,

$$u_{-i}^*(t) = \text{col}_{j \neq i}(-Q_j^{-1} B_j^T \lambda_j(t)), \quad C_i u_{-i}^*(t) = -\sum_{j \neq i} B_j Q_j^{-1} B_j^T \lambda_j(t).$$

Thus, substituting all players' stationary controls into the dynamics gives

$$\dot{x}(t) = Ax(t) - \sum_{j=1}^P B_j Q_j^{-1} B_j^T \lambda_j(t).$$

The corresponding costate equation is unchanged because the u_{-i} terms in H_i are independent of x , so $\partial H_i / \partial x$ contains no u_{-i} contribution. Collecting the state dynamics, initial condition, adjoint equation, and transversality condition yields the closed TPBVP

$$\dot{x}(t) = Ax(t) - \sum_{j=1}^P B_j Q_j^{-1} B_j^T \lambda_j(t), \quad (12)$$

$$x(t_0) = x_0, \quad (13)$$

$$\dot{\lambda}_i(t) = -R_i x(t) - A^T \lambda_i(t), \quad \lambda_i(T) = S_i x(T), \quad i = 1, \dots, P. \quad (14)$$

Even in the symmetric two-player case, the costates remain coupled through the shared state and are therefore solved jointly; no algebraic relation between $\lambda_1 + \lambda_2$ and x is assumed.

2.4. Fundamental Model of ELMs (Huang et al., 2006)

For the time-dependent problems considered here, the SLFN takes the scalar time variable as input. Given samples

$$\{(t_j, y_j)\}_{j=1}^N,$$

where $t_j \in \mathbb{R}$ is the time input and $y_j \in \mathbb{R}$ is the target value, the SLFN output is

$$\hat{y}_j = \sum_{k=1}^L \xi^k \sigma(w^k t_j + b^k), \quad j = 1, 2, \dots, N, \quad (15)$$

where $\hat{y}_j$ is the output of the SLFN; L is the number of neurons in the hidden layer; w^k, b^k denote the weights and biases of the hidden neurons; $\sigma(\cdot)$ is the (scalar-valued) activation function applied element-wise; and $\xi = [\xi^1, \xi^2, \dots, \xi^L]^T \in \mathbb{R}^L$ is the weight vector between the hidden layer and the output layer.

Let the SLFN output and target vectors be denoted, respectively, by

$$\begin{aligned} \hat{Y} &= [\hat{y}_1, \hat{y}_2, \dots, \hat{y}_N]^T \in \mathbb{R}^N, \\ Y &= [y_1, y_2, \dots, y_N]^T \in \mathbb{R}^N. \end{aligned}$$

The hidden-layer output matrix is defined by

$$H = \begin{bmatrix} \sigma(w^1 t_1 + b^1) & \dots & \sigma(w^L t_1 + b^L) \\ \sigma(w^1 t_2 + b^1) & \dots & \sigma(w^L t_2 + b^L) \\ \vdots & \ddots & \vdots \\ \sigma(w^1 t_N + b^1) & \dots & \sigma(w^L t_N + b^L) \end{bmatrix}. \quad (16)$$

Stacking the hidden-layer outputs at the N collocation points gives the compact matrix form

$$\hat{Y} = H\xi. \quad (17)$$

The ELM parameters $\alpha^k = (w^k, b^k)$ are sampled once and remain fixed during fitting. The hidden parameters and sampling points t_j then determine the output matrix H . Thus, fitting the target sequence reduces to least squares, and the following least-squares problem is formulated:

$$\min_{\xi} \|H\xi - Y\|_2. \quad (18)$$

The Moore–Penrose pseudoinverse is employed to obtain the optimal solution ξ^* , which is given by

$$\xi^* = H^\dagger Y, \quad (19)$$

$$\hat{Y} = HH^\dagger Y. \quad (20)$$

Here, $H^\dagger = (H^T H)^{-1} H^T$ requires that H have full column rank (i.e., $H^T H$ is invertible); this rank condition must be checked for the realised hidden parameters and sampling points and is not implied by $N \geq L$ alone.

To illustrate the algorithm, we consider a one-dimensional LQ optimal control problem; multi-dimensional cases are analogous.

$$\begin{aligned} \min_u J(x(t), u(t)) &= \frac{1}{2} \int_{t_0}^T (rx(t)^2 + qu(t)^2) dt + \frac{1}{2} sx(T)^2, \\ \text{s.t.} & \\ \dot{x}(t) &= ax(t) + cu(t), \\ x(t_0) &= x_0. \end{aligned} \quad (21)$$

Applying the PMP to this scalar problem yields the TPBVP form

$$\begin{cases} \dot{x}(t) - ax(t) + \frac{c^2}{q}\lambda(t) = 0, \\ rx(t) + a\lambda(t) + \dot{\lambda}(t) = 0, \\ x(t_0) = x_0, \\ -sx(T) + \lambda(T) = 0. \end{cases} \quad (22)$$

From this point onward, $\sigma(\cdot)$ denotes the scalar activation, whereas $\theta(t) := [\sigma(w^1t + b^1), \dots, \sigma(w^Lt + b^L)] \in \mathbb{R}^{1 \times L}$ denotes its row of hidden-feature evaluations and $\dot{\theta}(t)$ denotes the row derivative. In the ELM framework, the time interval $[t_0, T]$ is randomly partitioned into N segments, yielding the approximations

$$\begin{aligned} x(t_i) &\approx \theta(t_i)\beta, \\ \lambda(t_i) &\approx \theta(t_i)\zeta. \end{aligned}$$

Accordingly, (22) can be written as

$$\begin{bmatrix} \dot{\theta}(t_1) - a\theta(t_1) & \frac{c^2}{q}\theta(t_1) \\ \dot{\theta}(t_2) - a\theta(t_2) & \frac{c^2}{q}\theta(t_2) \\ \vdots & \vdots \\ \dot{\theta}(t_N) - a\theta(t_N) & \frac{c^2}{q}\theta(t_N) \\ r\theta(t_1) & a\theta(t_1) + \dot{\theta}(t_1) \\ r\theta(t_2) & a\theta(t_2) + \dot{\theta}(t_2) \\ \vdots & \vdots \\ r\theta(t_N) & a\theta(t_N) + \dot{\theta}(t_N) \\ \theta(t_0) & 0 \\ -s\theta(T) & \theta(T) \end{bmatrix} \begin{bmatrix} \beta \\ \zeta \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ x_0 \\ 0 \end{bmatrix}. \quad (23)$$

Denote the matrix on the left-hand side of (23) by $\Theta \in \mathbb{R}^{2(N+1) \times 2L}$ and $[\beta, \zeta]^T$ by $\xi \in \mathbb{R}^{2L}$, while the right-hand side is denoted by $Y \in \mathbb{R}^{2(N+1)}$. The first $2N$ rows enforce the differential equations at the N collocation points; the last two rows enforce the initial condition $x(t_0) = x_0$ and the transversality condition $\lambda(T) = sx(T)$, respectively. Because the initial and terminal conditions are algebraic constraints that contain no time derivatives, the corresponding rows of Θ involve only $\theta(\cdot)$ evaluated at t_0 and T , rather than $\dot{\theta}(\cdot)$; the time derivative $\dot{\theta}$ appears only in the interior rows that discretise the ODE residuals. Since $2(N+1) > 2L$ (i.e., the system is over-determined), the solution is defined in the least-squares sense. We then abbreviate (23) as

$$\Theta\xi = Y. \quad (24)$$

Solving the following least-squares problem

$$\min_{\xi} \|\Theta\xi - Y\|_2, \quad (25)$$

yields

$$\xi^* = \Theta^\dagger Y, \quad (26)$$

$$\beta^* = \xi_{1:L}^*, \quad (27)$$

$$\zeta^* = \xi_{L+1:2L}^*. \quad (28)$$

Finally, the optimal state and control are obtained as

$$x^*(t) = \theta(t)\beta^*, \quad (29)$$

$$u^*(t) = -\frac{c}{q}\theta(t)\zeta^*. \quad (30)$$

In the multi-input multi-output (MIMO) case where $x(t) \in \mathbb{R}^d$ and $u(t) \in \mathbb{R}^m$, the same approximations $x(t_i) \approx \theta(t_i)\beta^*$ and $\lambda(t_i) \approx \theta(t_i)\zeta^*$ hold, with $\beta^* \in \mathbb{R}^{L \times d}$ and $\zeta^* \in \mathbb{R}^{L \times d}$. The optimal control is then given in matrix form as $u^*(t) = -Q^{-1}B^T\lambda^*(t) = -Q^{-1}B^T\theta(t)\zeta^*$, where $Q \in \mathbb{R}^{m \times m}$ and $B \in \mathbb{R}^{d \times m}$.

Algorithm 1 summarises the procedure.

Algorithm 1 Solving an LQ optimal control problem using an ELM.

Require: N collocation points $\{t_j\}_{j=1}^N \subset [t_0, T]$, L hidden neurons, scalar coefficients a, c, q, r, s, x_0 , and activation $\sigma(\cdot)$.

Ensure: State $x^*(t) \in \mathbb{R}$ and control $u^*(t) \in \mathbb{R}$.

Draw $w = (w^1, \dots, w^L)^T \in \mathbb{R}^L$ and $b = (b^1, \dots, b^L)^T \in \mathbb{R}^L$.

Initialise $G, G' \in \mathbb{R}^{N \times L}$, $\Theta \in \mathbb{R}^{2(N+1) \times 2L}$, and $Y \in \mathbb{R}^{2(N+1)}$.

for $j = 1 \rightarrow N$ **do**

for $k = 1 \rightarrow L$ **do**

$G_{j,k} \leftarrow \sigma(w^k t_j + b^k) \in \mathbb{R}$

$G'_{j,k} \leftarrow w^k \sigma'(w^k t_j + b^k) \in \mathbb{R}$

end for

end for

Assemble Θ blockwise (each interior block is in $\mathbb{R}^{N \times L}$):

$\Theta_{1:N,1:L} \leftarrow G' - aG$, $\Theta_{1:N,L+1:2L} \leftarrow (c^2/q)G$,

$\Theta_{N+1:2N,1:L} \leftarrow rG$, $\Theta_{N+1:2N,L+1:2L} \leftarrow aG + G'$.

Set the four boundary blocks in $\mathbb{R}^{1 \times L}$:

$\Theta_{2N+1,1:L} \leftarrow \theta(t_0)$, $\Theta_{2N+1,L+1:2L} \leftarrow \mathbf{0}_{1 \times L}$,

$\Theta_{2N+2,1:L} \leftarrow -s\theta(T)$, $\Theta_{2N+2,L+1:2L} \leftarrow \theta(T)$,

where $\theta(t) = [\sigma(w^1 t + b^1), \dots, \sigma(w^L t + b^L)] \in \mathbb{R}^{1 \times L}$.

Set $Y = [\mathbf{0}_{2N}; x_0; 0] \in \mathbb{R}^{2(N+1)}$.

Compute $\xi^* = \Theta^\dagger Y \in \mathbb{R}^{2L}$, with $\Theta^\dagger \in \mathbb{R}^{2L \times 2(N+1)}$.

Extract $\beta^* = \xi_{1:L}^* \in \mathbb{R}^L$ and $\zeta^* = \xi_{L+1:2L}^* \in \mathbb{R}^L$.

Output: $x^*(t) = \theta(t)\beta^* \in \mathbb{R}$ and $u^*(t) = -(c/q)\theta(t)\zeta^* \in \mathbb{R}$.

Remark 1. For the LQ TPBVP in Theorem 1, the residuals are linear in the ELM output weights. The least-squares problem in (18) therefore determines those weights in one step. A nonlinear TPBVP would produce nonlinear residuals and require a different solver, so the one-step construction is limited to LQ problems.

Costate-Related Notation

Table 1 summarises the costate-related notation used in the ELM formulation and convergence analysis.

Table 1. Costate-related symbols used in this paper.

Symbol	Meaning	Dimension
$\lambda_i(t)$	costate (adjoint variable) of player i	$\mathbb{R}^d$
$\lambda_i^*(t)$	optimal costate of player i at the Nash equilibrium	$\mathbb{R}^d$
$\lambda(t)$	costate in the single-player 1D example	$\mathbb{R}$
$\lambda^*(t)$	optimal costate in the single-player example	$\mathbb{R}$
$\lambda_{-i}(t)$	stacked opponent costate $(\lambda_j)_{j \neq i}$	$\mathbb{R}^{d(P-1)}$

Table 1. Cont.

Symbol	Meaning	Dimension
$\lambda_L(t)$	SLFN approximation of $\lambda(t)$	$\mathbb{R}$
$\lambda_L^*(t)$	ELM pseudoinverse output costate	$\mathbb{R}$
ζ^k	output-layer weight for the costate, $k = 1, \dots, L$	$\mathbb{R}$
$\zeta = (\zeta^1, \dots, \zeta^L)$	full costate weight vector	$\mathbb{R}^L$
ζ^*	ELM pseudoinverse costate weights	$\mathbb{R}^L$
$\tilde{\zeta} = (\beta, \zeta)$	full SLFN output-weight vector (state + costate)	$\mathbb{R}^{2L}$
C_i	stacked opponent input matrix $[B_1, \dots, B_{i-1}, B_{i+1}, \dots, B_P]$	$\mathbb{R}^{d \times (m-m_i)}$

2.5. Convergence Analysis

For the scalar LQ optimal-control TPBVP, the following a posteriori estimate bounds the uniform state, costate, control, and cost errors by the differential-equation and boundary residuals of an ELM reconstruction. It gives no a priori rate in network width or collocation spacing. In a general game, the shared state couples the player-specific costates, so the scalar result cannot be applied to each player separately. Remark 2 specifies the scope of the augmented multi-player construction.

2.5.1. Setting and Assumptions

Consider the closed TPBVP on $t \in [t_0, T]$:

$$\begin{cases} \dot{x}(t) = a x(t) - \frac{c^2}{q} \lambda(t), & t \in (t_0, T], \\ \dot{\lambda}(t) = -r x(t) - a \lambda(t), & t \in [t_0, T), \\ x(t_0) = x_0, \\ \lambda(T) = s x(T), \end{cases} \quad (31)$$

with $t_0 < T$, $a, c, q, r, s \in \mathbb{R}$, $q > 0$, $r \geq 0$, $s \geq 0$, and $x_0 \in \mathbb{R}$. The associated optimal control is $u^*(t) = -(c/q)\lambda^*(t)$. The ELM reconstruction has the form

$$\begin{aligned} x_L^*(t) &= \sum_{k=1}^L \beta^{*,k} \sigma(w^k t + b^k), \\ \lambda_L^*(t) &= \sum_{k=1}^L \zeta^{*,k} \sigma(w^k t + b^k), \end{aligned} \quad (32)$$

where the output weights are obtained from the pseudoinverse system in (23).

Assumption 1 (Activation regularity). $\sigma \in C^2(\mathbb{R})$ with $\sup_{y \in \mathbb{R}} |\sigma(y)| \leq M_0$, $\sup_{y \in \mathbb{R}} |\sigma'(y)| \leq M_1$, and $\sup_{y \in \mathbb{R}} |\sigma''(y)| \leq M_2$ for finite constants M_0, M_1, M_2 .

Assumption 2 (Hyperbolic regime). With $b = c^2/q$, let

$$M = \begin{bmatrix} a & -b \\ -r & -a \end{bmatrix}, \quad \kappa = \sqrt{a^2 + br}.$$

We assume $\kappa > 0$. For $q > 0$ and $r \geq 0$, this holds when $a \neq 0$ or when $r > 0$ and $c \neq 0$.

Define the continuous equation residual and the two boundary residuals by

$$\rho_L(t) = \begin{bmatrix} \dot{x}_L^*(t) - ax_L^*(t) + b\lambda_L^*(t) \\ \dot{\lambda}_L^*(t) + rx_L^*(t) + a\lambda_L^*(t) \end{bmatrix}, \quad (33)$$

$$\eta_{0,L} = x_L^*(t_0) - x_0, \quad \eta_{T,L} = \lambda_L^*(T) - s\lambda_L^*(T), \quad (34)$$

$$\varepsilon_{\text{res},L} = \|\rho_L\|_{L^\infty(t_0,T)} + |\eta_{0,L}| + |\eta_{T,L}|. \quad (35)$$

2.5.2. Main Result

Theorem 2 (Residual-to-solution stability). *Under Assumptions 1 and 2, the TPBVP in (31) has a unique solution (x^*, λ^*) . There is a constant $C_{\text{res}} > 0$, depending only on a, c, q, r, s, t_0 , and T , such that the ELM reconstruction satisfies*

$$E_L := \sup_{t \in [t_0, T]} \left\| \begin{bmatrix} x_L^*(t) - x^*(t) \\ \lambda_L^*(t) - \lambda^*(t) \end{bmatrix} \right\|_2 \leq C_{\text{res}} \varepsilon_{\text{res},L}, \quad (36)$$

$$\|u_L^* - u^*\|_{L^\infty(t_0, T)} \leq \frac{|c|}{q} C_{\text{res}} \varepsilon_{\text{res},L}, \quad (37)$$

where $u_L^* = -(c/q)\lambda_L^*$. Moreover, for

$$J(x, u) = \frac{1}{2} \int_{t_0}^T (rx^2 + qu^2) dt + \frac{1}{2} sx(T)^2,$$

there exist constants $C_{J,1}, C_{J,2} > 0$, independent of L and N , such that

$$|J(x_L^*, u_L^*) - J(x^*, u^*)| \leq C_{J,1} \varepsilon_{\text{res},L} + C_{J,2} \varepsilon_{\text{res},L}^2. \quad (38)$$

Consequently, regarding any sequence of ELM reconstructions for which $\varepsilon_{\text{res},L} \rightarrow 0$ converges uniformly in state, costate, and control, its cost converges to the optimal cost.

Proof. Set $H = T - t_0$, $z = (x, \lambda)^T$, $e_1 = (1, 0)^T$, $e_2 = (0, 1)^T$, and $g = (-s, 1)^T$. Direct multiplication gives

$$M^2 = \begin{bmatrix} a & -b \\ -r & -a \end{bmatrix}^2 = \begin{bmatrix} a^2 + br & 0 \\ 0 & a^2 + br \end{bmatrix} = \kappa^2 I_2.$$

Consequently, $M^{2j} = \kappa^{2j} I_2$ and $M^{2j+1} = \kappa^{2j} M$ for every $j \geq 0$. Separating the even and odd terms in the matrix-exponential series therefore yields

$$\begin{aligned} \Phi(\tau) &= e^{M\tau} = \sum_{n=0}^{\infty} \frac{(M\tau)^n}{n!} \\ &= \sum_{j=0}^{\infty} \frac{\kappa^{2j} \tau^{2j}}{(2j)!} I_2 + \sum_{j=0}^{\infty} \frac{\kappa^{2j} \tau^{2j+1}}{(2j+1)!} M \\ &= \cosh(\kappa\tau) I_2 + \frac{\sinh(\kappa\tau)}{\kappa} M. \end{aligned} \quad (39)$$

The terminal condition is $g^T z(T) = 0$. Writing the initial value as $z(t_0) = x_0 e_1 + \lambda(t_0) e_2$ and using $z(T) = \Phi(H)z(t_0)$ gives

$$0 = g^T z(T) = x_0 g^T \Phi(H) e_1 + \lambda(t_0) g^T \Phi(H) e_2.$$

Thus, $g^T \Phi(H)e_2$ is the coefficient of the unknown initial costate. From (39),

$$Me_2 = \begin{bmatrix} -b \\ -a \end{bmatrix}, \quad \Phi(H)e_2 = \begin{bmatrix} -\frac{b}{\kappa} \sinh(\kappa H) \\ \cosh(\kappa H) - \frac{a}{\kappa} \sinh(\kappa H) \end{bmatrix}.$$

Since $g^T = (-s, 1)$, direct multiplication yields

$$\begin{aligned} D &:= g^T \Phi(H)e_2 \\ &= (-s) \left[-\frac{b}{\kappa} \sinh(\kappa H) \right] + \cosh(\kappa H) - \frac{a}{\kappa} \sinh(\kappa H) \\ &= \cosh(\kappa H) + \frac{sb - a}{\kappa} \sinh(\kappa H). \end{aligned} \quad (40)$$

Because $s, b \geq 0$ and $\kappa \geq |a|$, $(sb - a)/\kappa \geq -1$. Therefore,

$$D \geq \cosh(\kappa H) - \sinh(\kappa H) = e^{-\kappa H} > 0. \quad (41)$$

Thus, the terminal condition uniquely determines the initial costate, proving the existence and uniqueness of (x^*, λ^*) .

Let $z_L^* = (x_L^*, \lambda_L^*)^T$, $e = z_L^* - z^*$, and write $\rho_L = \dot{z}_L^* - Mz_L^*$. Then,

$$\dot{e} = Me + \rho_L, \quad e_1^T e(t_0) = \eta_{0,L}, \quad g^T e(T) = \eta_{T,L}. \quad (42)$$

Writing $e(t_0) = \eta_{0,L}e_1 + \nu e_2$, variation of constants gives

$$e(t) = \Phi(t - t_0)(\eta_{0,L}e_1 + \nu e_2) + \int_{t_0}^t \Phi(t - \tau)\rho_L(\tau) d\tau. \quad (43)$$

Applying g^T at $t = T$ and using (40) yields

$$\nu = \frac{1}{D} \left[\eta_{T,L} - \eta_{0,L} g^T \Phi(H)e_1 - \int_{t_0}^T g^T \Phi(T - \tau)\rho_L(\tau) d\tau \right]. \quad (44)$$

Let $K_\Phi = \max_{0 \leq \tau \leq H} \|\Phi(\tau)\|_2$ and $G = \|g\|_2$. From (41) and (44),

$$|\nu| \leq e^{\kappa H} [1 + GK_\Phi(1 + H)] \varepsilon_{\text{res},L}. \quad (45)$$

Substitution into (43) gives

$$\|e\|_{L^\infty(t_0, T)} \leq K_\Phi \left\{ 1 + H + e^{\kappa H} [1 + GK_\Phi(1 + H)] \right\} \varepsilon_{\text{res},L}. \quad (46)$$

This proves (36) with the coefficient on the right-hand side as C_{res} . Equation (37) follows from $u_L^* - u^* = -(c/q)(\lambda_L^* - \lambda^*)$.

It remains to bound the cost. Let $X^* = \|x^*\|_{L^\infty}$ and $U^* = \|u^*\|_{L^\infty}$. Since each component error is bounded by E_L and $|u_L^* - u^*| \leq (|c|/q)E_L$, direct expansion of the quadratic terms gives

$$\begin{aligned} |J(x_L^*, u_L^*) - J(x^*, u^*)| &\leq [H(rX^* + |c|U^*) + sX^*] E_L \\ &\quad + \frac{1}{2} \left[H \left(r + \frac{c^2}{q} \right) + s \right] E_L^2. \end{aligned} \quad (47)$$

Combining (47) with (36) proves (38). The convergence statement follows when $\varepsilon_{\text{res},L} \rightarrow 0$. $\square$

Remark 2 (Scope of the multi-player construction). For a P -player game with state dimension d , set $D_j = B_j Q_j^{-1} B_j^T$ and stack $z = (x^T, \lambda_1^T, \dots, \lambda_P^T)^T \in \mathbb{R}^{(P+1)d}$. The coupled TPBVP is

$$\dot{z} = \mathcal{M}_P z, \quad \mathcal{M}_P = \begin{bmatrix} A & -D_1 & -D_2 & \cdots & -D_P \\ -R_1 & -A^T & 0 & \cdots & 0 \\ -R_2 & 0 & -A^T & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -R_P & 0 & 0 & \cdots & -A^T \end{bmatrix},$$

with $x(t_0) = x_0$ and $\lambda_i(T) - S_i x(T) = 0$ for $i = 1, \dots, P$. Collocating this augmented system and all boundary conditions produces one linear least-squares system for the $(P+1)dL$ output weights. Theorem 2, however, is proven only for the scalar TPBVP in (31); no dimension-uniform or unchanged-rate claim is made for the coupled system.

Remark 3 (Linearity requirement). Theorem 2 uses the exact fundamental matrix of the constant-coefficient scalar TPBVP and the nonsingularity of its two-point boundary map. It is therefore a stability statement for the linear LQ system, not a convergence theorem for nonlinear residual equations. The ELM algorithm likewise retains a single linear least-squares solution only in the LQ setting.

3. Experiments

Example 1 (Nikooeinejad et al., 2016). Consider the following two-player LQ nonzero-sum differential game on $t \in [0, 3]$:

$$\begin{aligned} \min_{u_1} J_1(x(t), u_1(t)) &= \int_0^3 (x(t)^2 + u_1(t)^2) dt, \\ \min_{u_2} J_2(x(t), u_2(t)) &= \int_0^3 (4x(t)^2 + u_2(t)^2) dt + 5x(T)^2, \\ \text{s.t. } \dot{x}(t) &= 2x(t) + u_1(t) + u_2(t), \\ x(0) &= 1. \end{aligned} \quad (48)$$

The analytical solution is

$$\begin{aligned} x^*(t) &= e^{-3t}, \\ u_1^*(t) &= -e^{-3t} + e^{-(2t+3)}, \\ u_2^*(t) &= -4e^{-3t} - e^{-(2t+3)}. \end{aligned} \quad (49)$$

The ELM uses $N = 1000$ uniformly spaced collocation points and an SLFN with $L = 40$ hidden neurons. The hidden weights and biases are independently sampled from $\mathcal{N}(0, 1)$, and the sigmoid activation is

$$\sigma(w_k t_j + b_k) = \frac{1}{1 + \exp[-(w_k t_j + b_k)]}.$$

The sigmoid, Gaussian sampling, and grid satisfy the corresponding regularity, sampling, and discretisation assumptions used in the scalar analysis. Figures 1 and 2 show the resulting state and control trajectories.

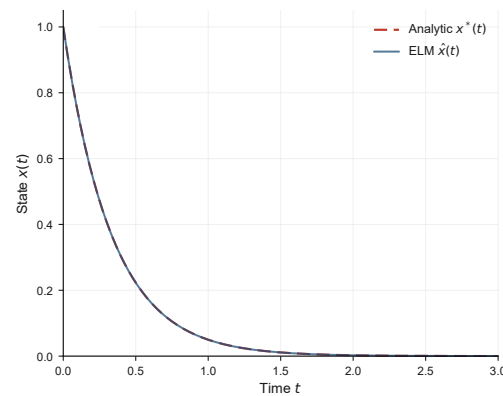


Figure 1. State trajectory $x(t)$ in Example 1, obtained using an ELM with $N = 1000$ collocation points, $L = 40$ hidden neurons, and sigmoid activation.

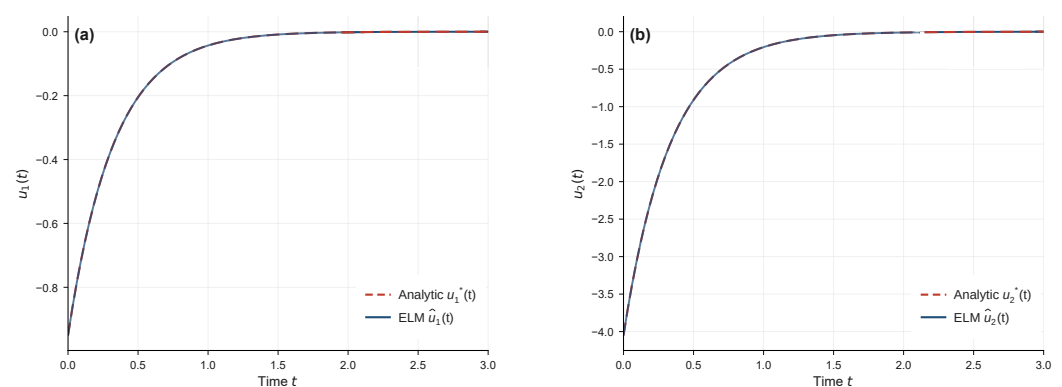


Figure 2. Control trajectories in Example 1: ELM solution (blue solid) with $N = 1000$, $L = 40$, and sigmoid activation, compared with the analytic solution (red dashed). (a) $u_1(t)$. (b) $u_2(t)$.

Section 3.1 assesses activation performance across all three benchmarks; sigmoid is used here only for the trajectory and solver comparisons.

For reproducibility, the shooting baseline integrates $(x, \lambda_1, \lambda_2)$ on $[0, 3]$ using 1000 uniform RK4 steps ($h = 0.003$). Newton iteration starts from $(0, 0)^T$, uses variational sensitivities propagated by the same RK4 scheme, stops when the terminal-residual infinity norm is at most 10^{-10} , and is limited to 10 updates. It converged in one update with a terminal residual of 6.831×10^{-11} .

Errors are evaluated at 1001 grid nodes against Equation (49). MAE is the average componentwise maximum absolute error over (x, u_1, u_2) , and RLE is the relative Frobenius norm of their concatenated trajectories. Table 2 reports the 30-seed ELM means and the deterministic shooting results. Under this protocol, shooting is more accurate. Fifty complete shooting solutions (one warm-up excluded) took 0.02325 ± 0.00100 seconds in Python/NumPy; this timing is reported for transparency but is not compared directly with the MATLAB ELM runs. Specifically, the shooting implementation used Python 3.13.9 with NumPy 2.3.5, whereas the ELM implementation used MATLAB R2024a.

Table 2. Accuracy of the ELM and RK4 shooting solutions in Example 1. ELM values are 30-seed means.

	ELMs	Shooting Method
MAE	2.704×10^{-8}	2.443×10^{-10}
RLE	1.631×10^{-8}	1.636×10^{-10}

Let $d_z = (P + 1)d$. The dense collocation matrix has $M_\Theta = d_z(N + 1)$ rows and $K_\Theta = d_z L$ columns; a QR- or SVD-based solve costs $O(M_\Theta K_\Theta^2 + K_\Theta^3)$ and stores $O(M_\Theta K_\Theta)$ entries. Thus, the reported small-problem results do not establish a high-dimensional efficiency advantage.

Riccati methods can provide reusable feedback solutions when a stabilising solution exists, whereas the present ELM computes one open-loop trajectory.

Matrix-free, sparse, low-rank, and domain-decomposition implementations remain future work.

Example 2. To assess performance in higher dimensions, consider the following two-player LQ nonzero-sum differential game on $t \in [0, 3]$, with $x(t) \in \mathbb{R}^5$ and scalar controls $u_1(t)$ and $u_2(t)$:

$$\begin{aligned} \min_{u_i} J_i &= \frac{1}{2} x(3)^T M_i x(3) + \frac{1}{2} \int_0^3 \left(x(t)^T Q_i x(t) + u_i(t)^T R_i u_i(t) \right) dt, \\ \text{s.t. } \dot{x}(t) &= Ax(t) + B_1 u_1(t) + B_2 u_2(t), \\ x(0) &= [2, -1, 1, -2, 1]^T. \end{aligned} \quad (50)$$

The system matrices are

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ -1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 \end{bmatrix} \in \mathbb{R}^{5 \times 5}, \quad B_1 = B_2 = [0, 1, 0, 1, 0]^T \in \mathbb{R}^{5 \times 1},$$

$$Q_1 = \text{diag}(2, 1, 2, 1, 2) \in \mathbb{R}^{5 \times 5}, \quad Q_2 = \text{diag}(1, 2, 1, 2, 1) \in \mathbb{R}^{5 \times 5},$$

$$M_1 = M_2 = I_5 \in \mathbb{R}^{5 \times 5}, \quad R_1 = R_2 = 1 \in \mathbb{R}.$$

The two 2×2 blocks in A have eigenvalues $(-1 \pm i\sqrt{3})/2$, and the fifth state has eigenvalue -1 . The identical input vectors $B_1 = B_2$ actuate the second and fourth state components. The choices $M_1 = M_2 = I_5$ and $R_1 = R_2 = 1$ impose equal terminal-state and control-effort penalties.

The FDM reference and the ELM use $N = 3000$ grid points. The ELM has $L = 150$ hidden neurons with tanh activation; tanh satisfies Assumption 1.

Figures 3 and 4 compare the state and control trajectories; the ELM results closely follow the FDM reference.

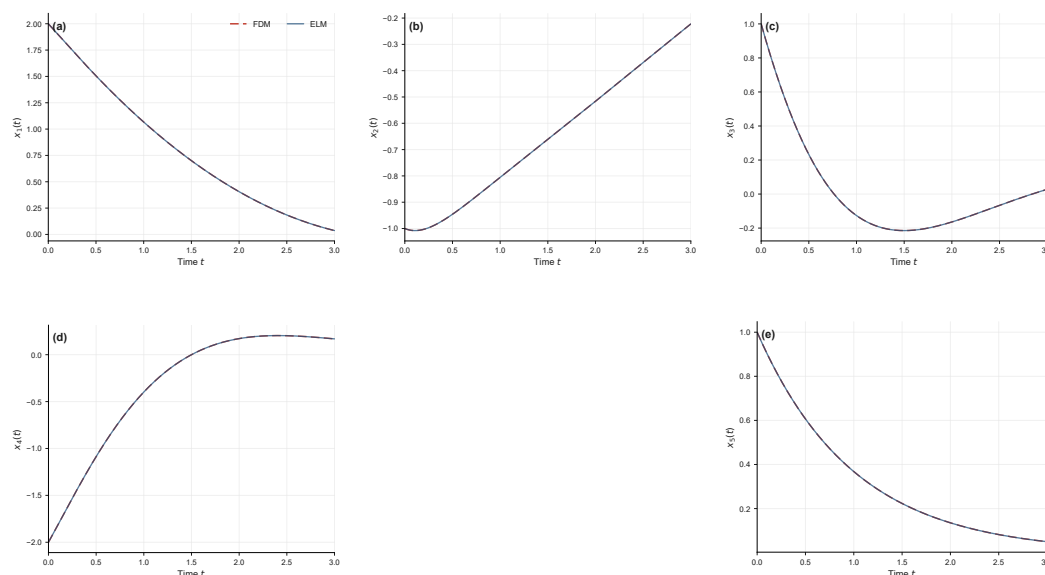


Figure 3. State trajectories (x_1 – x_5) in Example 2: ELM solution (blue solid) with $N = 3000$, $L = 150$, and tanh activation, compared with the FDM reference (red dashed). (a) $x_1(t)$. (b) $x_2(t)$. (c) $x_3(t)$. (d) $x_4(t)$. (e) $x_5(t)$.

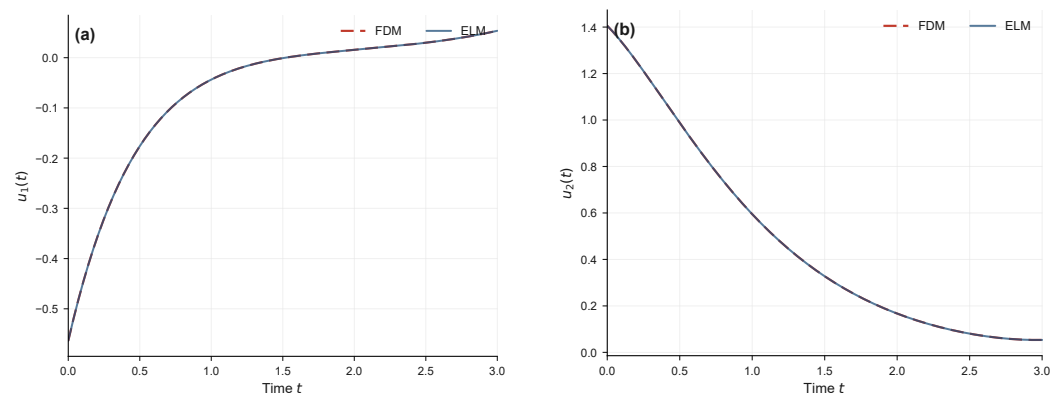


Figure 4. Control trajectories in Example 2: ELM solution (blue solid) with $N = 3000$, $L = 150$, and tanh activation, compared with the FDM reference (red dashed). (a) $u_1(t)$. (b) $u_2(t)$.

Table 3 reports the componentwise ELM–FDM state discrepancies and the retained execution times for the original configuration; control errors are assessed separately below.

Table 3. Componentwise ELM–FDM state discrepancies and retained execution times for Example 2.

	x_1	x_2	x_3	x_4	x_5	Average
ELM–FDM discrepancy	8.21×10^{-4}	6.53×10^{-4}	9.12×10^{-4}	7.84×10^{-4}	5.93×10^{-4}	7.53×10^{-4}
FDM runtime (s)			0.58			
ELM runtime (s)			0.011			

We assessed control errors in a 30-seed experiment with $N = 1000$, $L = 40$, tanh activation, and a matrix-exponential reference. MAE and RLE are computed separately for the state and control groups. Table 4 reports the means and 95% confidence-interval half-widths.

Table 4. Thirty-seed state and control errors for Example 2 ($N = 1000$, $L = 40$, tanh). Values are mean (95% CI half-width).

Component Group	MAE	RLE
State	1.613×10^{-10} (6.546×10^{-11})	1.244×10^{-10} (4.938×10^{-11})
Control	4.339×10^{-10} (1.763×10^{-10})	4.798×10^{-10} (1.894×10^{-10})

Table 3 retains the original fixed-configuration ELM–FDM discrepancies and legacy execution times. Because the timing protocol is not matched to the other experiments, these values are descriptive only and do not establish comparative speed or high-dimensional scaling. The supplementary check independently evaluates state and control accuracy against the matrix-exponential reference.

Example 3 (Heterogeneous three-player 3D benchmark). To assess performance with more players and heterogeneous coefficients, consider the following three-player, three-dimensional game on $t \in [0, 2]$:

$$\min_{u_i} J_i = \frac{1}{2} x(2)^T S_i x(2) + \frac{1}{2} \int_0^2 \left(x(t)^T R_i x(t) + q_i u_i(t)^2 \right) dt, \quad i = 1, 2, 3, \quad (51)$$

$$\dot{x}(t) = Ax(t) + \sum_{i=1}^3 B_i u_i(t), \quad x(0) = [1, -0.5, 0.75]^T.$$

The heterogeneous coefficients are

$$A = \begin{bmatrix} -0.6 & 0.2 & 0 \\ -0.1 & -0.8 & 0.15 \\ 0 & -0.2 & -1.0 \end{bmatrix}, \quad (q_1, q_2, q_3) = (1, 1.5, 0.8),$$

$$B_1 = \begin{bmatrix} 1 \\ 0.2 \\ 0 \end{bmatrix}, \quad B_2 = \begin{bmatrix} 0.1 \\ 1 \\ 0.2 \end{bmatrix}, \quad B_3 = \begin{bmatrix} 0 \\ 0.3 \\ 1 \end{bmatrix},$$

$$R_1 = \text{diag}(1, 0.5, 0.2), \quad R_2 = \text{diag}(0.3, 1.2, 0.4),$$

$$R_3 = \text{diag}(0.4, 0.2, 1.4),$$

$$S_1 = \text{diag}(1, 0.5, 0.3), \quad S_2 = \text{diag}(0.4, 1, 0.6),$$

$$S_3 = \text{diag}(0.6, 0.3, 1.2).$$

The reference solution is obtained from $z(t) = \exp(\mathcal{M}_3 t)z(0)$, with the initial costates determined by $\lambda_i(2) = S_i x(2)$. The ELM uses $N = 400$, $L = 40$, $\tanh$ activation, and 30 random seeds. MAE and RLE are computed separately for the state and control groups. Table 5 reports the mean, standard deviation, and 95% confidence-interval half-width.

Table 5. Thirty-seed results for the heterogeneous three-player 3D benchmark.

Metric	Mean	Standard Deviation	95% CI Half-Width
State MAE	1.159×10^{-12}	1.507×10^{-12}	5.391×10^{-13}
Control MAE	2.146×10^{-12}	2.837×10^{-12}	1.015×10^{-12}
Overall MAE	1.653×10^{-12}	2.169×10^{-12}	7.760×10^{-13}
State RLE	1.670×10^{-12}	2.162×10^{-12}	7.737×10^{-13}
Control RLE	6.583×10^{-12}	8.920×10^{-12}	3.192×10^{-12}
Overall RLE	3.572×10^{-12}	4.777×10^{-12}	1.709×10^{-12}
Maximum boundary residual	2.967×10^{-14}	4.037×10^{-14}	1.445×10^{-14}
ELM runtime (s)	0.1700	0.0210	0.0075

Figure 5 uses the seed whose overall MAE is closest to the median, rather than the best-performing seed. The ELM trajectories closely follow the reference.

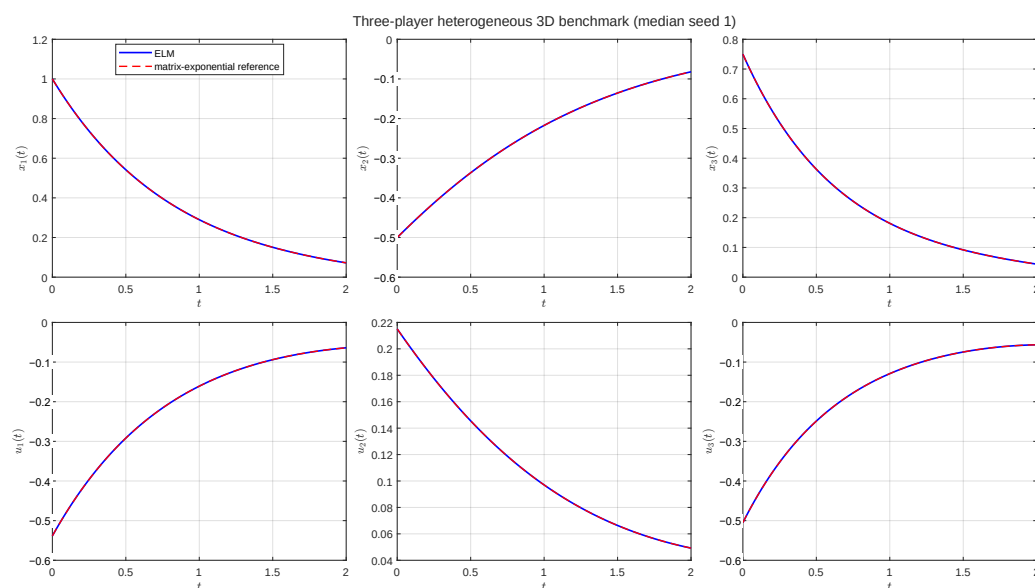


Figure 5. State and control trajectories for the heterogeneous three-player 3D benchmark: ELM solution for the median-MAE seed (blue solid) and matrix-exponential reference (red dashed).

3.1. Cross-Benchmark Activation-Function Assessment

We evaluate sigmoid, tanh, sine, and Gaussian RBF $\exp(-s^2)$ on all three benchmarks using $N = 1000$, $L = 40$, and 30 paired seeds. The same Gaussian hidden parameters are used across activations for each seed. All four activations satisfy Assumption 1. Errors are computed against the analytical reference for Example 1 and the matrix-exponential references for Examples 2 and 3. Table 6 reports the mean overall MAE and 95% confidence-interval half-width.

Table 6. Paired activation-function sweep over 30 seeds with $N = 1000$ and $L = 40$. Entries are mean overall MAE (95% CI half-width); the lowest mean in each example is bold.

Activation	Example 1	Example 2	Example 3
Sigmoid	2.704×10^{-8} (1.242×10^{-8})	1.356×10^{-9} (4.161×10^{-10})	2.548×10^{-11} (1.322×10^{-11})
Tanh	3.831×10^{-10} (4.852×10^{-10})	2.392×10^{-10} (9.712×10^{-11})	2.630×10^{-12} (1.434×10^{-12})
Sine	2.949×10^{-5} (2.253×10^{-5})	4.162×10^{-9} (1.696×10^{-9})	5.703×10^{-11} (1.979×10^{-11})
Gaussian RBF	6.169×10^{-10} (2.487×10^{-10})	2.139×10^{-11} (9.918×10^{-12})	6.232×10^{-13} (2.538×10^{-13})

MAE and RLE give the same ranking: tanh is most accurate in Example 1, whereas Gaussian RBF is most accurate in Examples 2 and 3. The corresponding lowest RLE values are 2.213×10^{-10} , 2.180×10^{-11} , and 1.481×10^{-12} , respectively. All 360 solutions completed successfully.

The preferred activation is benchmark-dependent; no activation is recommended universally.

3.2. Controlled Comparison with PINNs

We compare the ELM with gradient-trained PINNs (Raissi et al., 2019) on all three benchmarks. Both methods use the same dynamics, horizon, initial and terminal conditions, cost weights, and analytical or matrix-exponential references.

Each PINN uses separate four-hidden-layer tanh networks for the state, controls, and costates, with widths 50, 30, and 40, respectively. Adaptive uncertainty weights balance the initial-condition, state, costate, stationarity, and terminal residuals. Adam training uses 30,000 iterations, with learning rates 10^{-4} for the network parameters and 10^{-3} for the adaptive weights.

The ELM uses $(N, L) = (1000, 40)$ in Examples 1 and 2, and $(N, L) = (400, 40)$ in Example 3. The corresponding PINN collocation counts are 1000, 3000, and 400.

Table 7 reports the combined state–control MAE and RLE. MAE averages the componentwise maximum errors, and RLE is the relative Frobenius norm of the concatenated trajectories. ELM values are 30-seed means; PINN values are from one fixed-seed run.

Table 7. Accuracy comparison between ELMs and PINNs on the three manuscript benchmarks. ELM values are 30-seed means; PINN values are from one fixed-seed run.

Example	Method	MAE	RLE
1	ELM	3.831×10^{-10}	2.213×10^{-10}
1	PINN	3.943×10^{-1}	2.359×10^{-1}
2	ELM	2.392×10^{-10}	2.223×10^{-10}
2	PINN	1.398×10^{-3}	1.010×10^{-3}
3	ELM	1.653×10^{-12}	3.572×10^{-12}
3	PINN	6.058×10^{-4}	6.375×10^{-4}

Under this protocol, the ELM errors are lower on all three benchmarks. Because the ELM values average 30 seeds whereas each PINN value represents one fixed seed, the

comparison is limited to the reported runs and does not establish universal superiority, particularly for nonlinear problems.

4. Robustness and Statistical Validation

The robustness tests for Example 1 measure seed variability, sensitivity to the hidden-neuron count, and boundary-condition residuals. Unless stated otherwise, they use tanh activation, $N = 1000$ collocation points, and 30 independent seeds.

4.1. Sensitivity to the Number of Hidden Neurons

We evaluate $L \in \{20, 40, 60, 80, 100, 120\}$ using 30 independent Gaussian hidden-parameter initialisations for each value. Table 8 reports the MAE statistics, and Figure 6 shows the resulting trend.

Table 8. Sensitivity to hidden-neuron count L in Example 1. Values are the mean, standard deviation, and 95% confidence interval of MAE over 30 seeds ($N = 1000$, tanh).

L	MAE Mean	MAE Std	MAE 95% CI
20	7.89×10^{-8}	2.41×10^{-7}	$\pm 8.62 \times 10^{-8}$
40	1.35×10^{-8}	7.06×10^{-8}	$\pm 2.53 \times 10^{-8}$
60	1.16×10^{-10}	2.12×10^{-10}	$\pm 7.58 \times 10^{-11}$
80	1.47×10^{-10}	5.66×10^{-10}	$\pm 2.02 \times 10^{-10}$
100	9.69×10^{-11}	1.75×10^{-10}	$\pm 6.25 \times 10^{-11}$
120	9.43×10^{-11}	1.83×10^{-10}	$\pm 6.54 \times 10^{-11}$

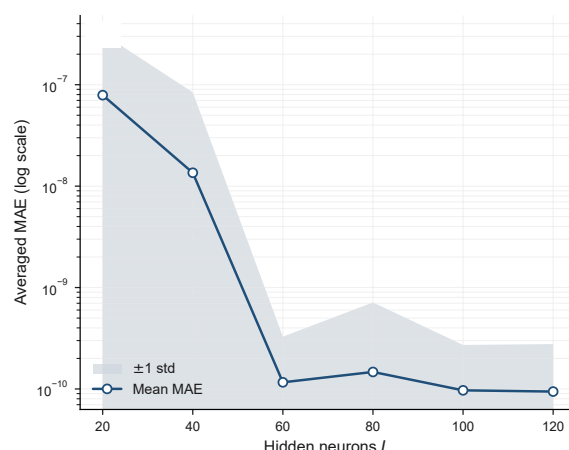


Figure 6. Mean MAE versus hidden-neuron count L on a logarithmic scale. Error bars indicate $\pm$ one standard deviation over 30 seeds.

Mean MAE decreases by approximately two orders of magnitude from $L = 20$ to $L = 60$ and is non-monotone thereafter. The coefficient of variation at $L = 80$ is approximately 3.85, indicating substantial seed variability. Thus, neither seed-independent performance nor a universal choice of L is supported; the observed behaviour over $L \in [60, 120]$ is specific to this benchmark. In practice, L should be increased progressively using a fixed, prespecified seed batch, with both mean error and dispersion monitored; the smallest stable value meeting the target accuracy can then be retained, and regularisation considered if conditioning deteriorates.

4.2. Boundary-Condition Residuals

The ELM enforces three boundary conditions through dedicated rows of $\Theta\xi = Y$:

$$x(0) = 1, \quad \lambda_1(T) = 0, \quad \lambda_2(T) = 10x(T).$$

We evaluate the corresponding residuals over 30 independent seeds. Table 9 summarizes the boundary-condition residuals over these seeds.

Table 9. Boundary-condition residuals in Example 1 over 30 seeds. The table reports the median, mean, standard deviation, and maximum absolute residual.

Quantity	Median	Mean	Std	Max
$ x_L^*(0) - 1 $ (IC)	5.33×10^{-13}	1.53×10^{-12}	2.28×10^{-12}	9.78×10^{-12}
$ \lambda_{1,L}^*(T) $ (player-1 TC)	1.99×10^{-12}	1.27×10^{-11}	3.96×10^{-11}	2.17×10^{-10}
$ \lambda_{2,L}^*(T) - 10x_L^*(T) $ (player-2 TC)	3.96×10^{-12}	2.19×10^{-11}	6.71×10^{-11}	3.69×10^{-10}

Figure 7 shows the distribution of these residuals on a logarithmic scale. The median initial-condition residual is 5.33×10^{-13} , and the median transversality residuals are below 4×10^{-12} . The maximum residual across all seeds and conditions is 3.69×10^{-10} . Thus, the dedicated boundary rows enforce all three conditions with small numerical residuals.

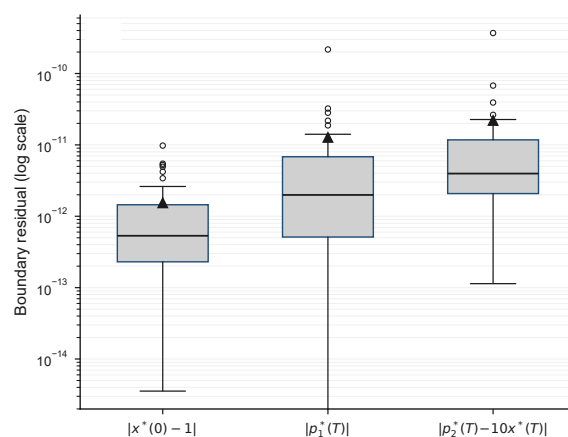


Figure 7. Distribution of boundary-condition residuals on a logarithmic scale over 30 independent seeds. Boxes span the interquartile range; whiskers extend to the most extreme non-outlier values; the green triangle marks the mean.

5. Discussion and Conclusions

We proposed an ELM-based numerical framework for LQ nonzero-sum differential games. The PMP recasts the LQ game as a linear TPBVP. Because the SLFN output is linear in its output weights, the optimal weights follow from a single Moore–Penrose pseudoinverse, with no gradient-based iteration. Analytical and matrix-exponential references verify the ELM solutions, whose errors are lower than the fixed-seed PINN errors on the three tested games; this does not, however, establish universal superiority over PINNs or nonlinear solvers. For the scalar LQ optimal-control TPBVP, the residual-stability theorem shows that vanishing continuous equation and boundary residuals imply uniform convergence of the state, costate, and control and convergence of the cost. It does not claim an a priori rate in L or h , nor an unchanged multi-player rate.

The framework has two limitations. First, it inherits the PMP's requirement of linear dynamics and quadratic cost, so it does not extend to nonlinear or non-quadratic games. Second, the TPBVP becomes hard to assemble and solve in very high dimensions or when only part of the dynamics is known.

For nonlinear open-loop games, the PMP produces a nonlinear TPBVP, so the residual is no longer linear in the ELM output weights and an iterative nonlinear solver would be required. Feedback Nash equilibria generally require coupled Hamilton–Jacobi–Bellman equations over the time–state domain and therefore lie outside the present trajectory-

based formulation. These extensions require separate analyses of solvability, iteration convergence, and residual stability.

Author Contributions: Conceptualisation, C.D.; methodology, C.D. and Y.Y.; software, Y.Y.; validation, C.D. and Y.Y.; formal analysis, C.D.; writing—original draft preparation, Y.Y.; writing—review and editing, C.D. and Y.Y.; visualisation, Y.Y.; supervision, C.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data that support the findings of this study are available from the corresponding author upon reasonable request.

Acknowledgments: During the preparation of this manuscript, the authors used DeepSeek V4 for language polishing and to improve the readability of the text.

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

References

- Alomari, M. W. (2025). High-accuracy spectral-like Legendre–Darboux method for initial value problems. *Mathematics*, *13*, 3319.
- Barghi Jond, H., & Nabiyev, V. (2019). On the finite horizon Nash equilibrium solution in the differential game approach to formation control. *Journal of Systems Engineering and Electronics*, *30*, 1233–1242.
- Chen, X., Lu, Z., Shao, Y., & Yuan, D. (2026). Optimal control and zero-sum differential game for stochastic uncertain systems with randomness and uncertainty. *Transactions of the Institute of Measurement and Control*, *48*, 46–65.
- Clemhout, S., & Wan, H. Y., Jr. (1994). Differential games—Economic applications. In R. J. Aumann & S. Hart (Eds.), *Handbook of game theory with economic applications* (Vol. 2, pp. 801–825). Elsevier.
- Cole, D., & Sands, T. (2023). Complex dynamical orbit re-entry navigation and control. *Preprints*. <https://doi.org/10.20944/preprints202312.0787.v1>.
- Engwerda, J. (2005). *LQ dynamic optimization and differential games*. John Wiley & Sons.
- Ge, X., Yuan, H., Li, Z., Chen, H., Ke, Z., & Li, L. (2026). Density target assignment pursuit strategy for multi-agent pursuit-evasion games in bounded region. *International Journal of Systems Science*, 1–15. <https://doi.org/10.1080/00207721.2026.2617146>.
- Ha, R. P., & Halme, A. (1976). A solution of nonlinear TPBVP's occurring in optimal control. *Automatica*, *12*, 403–415.
- Huang, G.-B., Zhu, Q.-Y., & Siew, C.-K. (2006). Extreme learning machine: Theory and applications. *Neurocomputing*, *70*, 489–501.
- Kashiri, N., Ghasemi, M. H., & Dardel, M. (2011). An iterative method for time optimal control of dynamic systems. *Archives of Control Sciences*, *21*, 5–23.
- Li, M., Rizvi, S. U. A., & Lin, W. (2024). A deep reinforcement-learning framework for open-loop Nash strategies in two-player differential games. *Automatica*, *167*, 111806.
- Mowlavi, S., & Nabi, S. (2023). Optimal control of PDEs using physics-informed neural networks. *Journal of Computational Physics*, *473*, 111731.
- Na, K.-M., & Lee, C.-H. (2024). Physics-informed deep learning approach to solve optimal control problem. In *AIAA SciTech 2024 forum* (p. 945). AIAA.
- Nikooeinejad, Z., Delavarkhalafi, A., & Heydari, M. (2016). A numerical solution of open-loop Nash equilibrium in nonlinear differential games based on Chebyshev pseudospectral method. *Journal of Computational and Applied Mathematics*, *300*, 369–384.
- Raissi, M., Perdikaris, P., & Karniadakis, G. E. (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, *378*, 686–707.
- Ren, H., Ma, Y., Jiang, B., Polycarpou, M. M., & Panayiotou, C. (2026). Zero-sum differential game based fault-tolerant control for a class of nonlinear multi-agent systems with communication faults. *IEEE Transactions on Automatic Control*, *71*(6), 4223–4230.
- Sadana, U., Reddy, P. V., & Zaccour, G. (2023). Feedback Nash equilibria in differential games with impulse control. *IEEE Transactions on Automatic Control*, *68*, 4523–4538.
- Saltari, E., Semmler, W., & Di Bartolomeo, G. (2022). A Nash equilibrium for differential games with moving-horizon strategies. *Computational Economics*, *60*, 1041–1054.
- Sun, Z., Liu, Y., & Karimi, L. B. (2023). Physics-informed neural networks for solving differential games with non-smooth Hamiltonians. *IEEE Transactions on Automatic Control*, *68*, 8121–8134.
- Wang, T. (2026). Safety-oriented cooperative control for connected and autonomous vehicle platoons using differential game theory and risk potential field. *World Electric Vehicle Journal*, *17*, 67.

- Wu, F., Li, X., Xiong, J., & Zhang, X. (2026). Stochastic linear-quadratic differential game with regime-switching in an infinite horizon. *SIAM Journal on Control and Optimization*, 64, 175–210.
- Xu, C., Zhang, Y., Wang, W., & Dong, L. (2022). Pursuit and evasion strategy of a differential game based on deep reinforcement learning. *Frontiers in Bioengineering and Biotechnology*, 10, 827408.
- Zhang, B.-Q., Wang, B.-C., & Cao, Y. (2024). An online Q-learning method for linear-quadratic nonzero-sum stochastic differential games with completely unknown dynamics. *Journal of Systems Science and Complexity*, 37, 1907–1922.
- Zhang, L., Ghimire, M., Xu, Z., Zhang, W., & Ren, Y. (2024a). Pontryagin neural operator for solving general-sum differential games with parametric state constraints. In *Proceedings of the 6th annual learning for dynamics and control conference (L4DC)* (pp. 1728–1740). PMLR.
- Zhang, L., Ghimire, M., Zhang, W., Xu, Z., & Ren, Y. (2024b). Value approximation for two-player general-sum differential games with state constraints. *IEEE Transactions on Robotics*, 40, 4631–4649.
- Zhang, Y., Wang, Y.-W., Liu, Z.-W., & Luo, Y.-F. (2025). Differential graphical game for linear multi-agent systems with deceptive agents or traitorous agents. *IEEE Transactions on Network Science and Engineering*, 13, 1311–1328.
- Zhuang, H., Shen, Q., Wu, S., Razoumny, V. Y., & Razoumny, Y. N. (2026). Design of zero-sum differential game control based on model-free reinforcement learning method and disturbance observer. *Aerospace Systems*, 1–9. <https://doi.org/10.1007/s42401-025-00441-2>.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.