Preprints of the 23rd IFAC World Congress
Busan, Republic of Korea, August 23-28, 2026

# Applying Kolmogorov-Arnold Networks to Improve Linear Quadratic Control Performance

**Tamás Hegedűs** * **Balázs Németh** *,** **Dániel Fényes** *
**Péter Gáspár** *,**

* *HUN-REN Institute for Computer Science and Control (SZTAKI), Kende u. 13-17, H-1111 Budapest, Hungary. (e-mail: [tamas.hegedus;balazs.nemeth;daniel.fenyes;peter.gaspar]@sztaki.hu).*
** *Department for Control of Transportation and Vehicle Systems, Budapest University of Technology and Economics, Stoczek u 2., H-1111 Budapest, Hungary. (e-mail: [balazs.nemeth;peter.gaspar]@kjk.bme.hu)*

**Abstract:**
This paper presents a control architecture in which a Linear Quadratic Regulator (LQR) is integrated with a machine learning-based nonlinear compensator realized through a Kolmogorov-Arnold Network (KAN). The goal of the proposed approach is to achieve a balance between guaranteed closed-loop stability, computational efficiency, and enhanced control performance. First, a Reinforcement Learning (RL) framework is used to learn an effective nonlinear control policy across the entire operating range of the system. Then, the trained RL-based control method serves as the teacher network for the KAN-based algorithm. During the knowledge distillation, gradient regularization and coefficient constraints are applied to achieve a smooth and Lipschitz-bounded neural network-based controller. Stability is guaranteed for the combined LQR-KAN closed-loop system through the computation of the maximum allowable Lipschitz constant of the neural network. The proposed control structure is validated on a double inverted pendulum system. The results show that the combined KAN-based controller achieves nearly the same performance level as the RL-based method, while the Lipschitz constant and the complexity of the network are significantly reduced. This property directly supports simple stability analysis and reliable real-time implementation.

*Keywords:* Kolmogorov-Arnold Networks, combined control structure

## 1. INTRODUCTION AND MOTIVATION

The combination of classical control and machine learning has become a promising direction for handling highly nonlinear systems. In complex systems such as autonomous or aerial vehicles and robotics, purely linear controllers may suffer from performance loss, while data-driven methods often lack theoretical stability guarantees. Therefore, hybrid control structures combine a stabilizing classical controller with a learning-based component to improve the performance level of the control loop.

In the literature, several methods can be found for the combined control structure. For instance, machine learning can be combined with $H_\infty$ control for safety supervision (Lelkó and Németh (2024)), machine learning-based solutions can be used to adapt PID parameters (Qin et al. (2018)), and a faster learning convergence and improved tracking performance can be reached through the hybrid control structure (Yoo et al. (2021)). Moreover, the machine learning-based algorithms can be effectively used for modeling purposes in the control loop (Nicodemus et al. (2021)). Although combined methods can provide significant performance improvements, they often suffer from a key limitation: the high complexity of the neural networks, which makes it difficult to guarantee closed-loop stability.

A novel neural network architecture, which is called Kolmogorov-Arnold Networks (KANs), was proposed by Liu et al. (2025). The core idea of the method comes from the Kolmogorov-Arnold theorem (Schmidt-Hieber (2021)). While classical Multi-Layer Perceptrons (MLPs) are widely used in combined control structures, the fixed activation functions and large parameter sets often lead to high complexity, limited smoothness, and increased

[1] The paper was funded by the National Research, Development and Innovation Office (NKFIH) under NKKP Grant Agreement No. STARTING 153675. The research was also supported by the National Research, Development and Innovation Office through the project "Digital-Twin-Driven Resilience Modeling Framework for International Rail Freight Corridors" (NKFIH: 2025-1.2.4-TÉT-2025-00039. The work of Dániel Fényes was supported by the János Bolyai Research Schol-arship of the Hungarian Academy of Sciences.

sensitivity to perturbations. Considering these properties, the stability analysis is challenging, and the resulting control system may become conservative. In contrast, KANs are constructed using smooth univariate spline functions, with which higher smoothness, improved interpretability, and much lower model complexity can be reached. It is pointed out by Dong et al. (2025) that the KAN-based solutions can outperform MLPs in accuracy, and the tests also showed improved robustness. These properties are essential in safety-critical control systems, with which more reliable and robust operation can be reached.

Inspired by these findings, this paper proposes a combined control structure in which system stability is guaranteed by a Linear Quadratic Regulator (LQR), while the nonlinear and unmodeled dynamics are taken into account through a Lipschitz-constrained KAN. LQR-based approaches are widely used in practice due to their computational efficiency (Mehrmann (1991)). Furthermore, the feedback gains obtained from the quadratic cost function provide robustness against disturbances and moderate parameter variations around the operating point. In the proposed control structure, the output of the KAN is treated as an additional control input.

The proposed method can be divided into three main design steps. First, a classical LQR feedback controller is designed at the linearized operating point of the system. Second, an additional control input is applied using a reinforcement learning-based method to improve the tracking performance of the control system. It is important to note that the paper does not focus on designing a new RL-based additional controller, instead, it relies on previously published RL solutions, and the contribution lies in how these results are integrated into a KAN-based structure and analyzed for the closed-loop stability. Furthermore, the proposed framework does not rely solely on an RL-based algorithm. In the architecture, the RL component acts as a teacher network, which can be replaced by any other machine-learning-based solution. Thus, the method remains flexible and is not restricted to reinforcement learning. The RL-based approach is chosen because it can learn a near-optimal control policy and perform effectively on nonlinear systems. Moreover, RL-based approaches have been developed for various control systems in recent years (Kiumarsi et al. (2018)). In the presented method, the KAN is trained as a supervised student network. The KAN is chosen to exploit its advantages over the classical MLPs, which lie in the possibility to produce smooth and constrained control actions, which significantly simplifies the stability analysis of the closed-loop system. Third, the goal of the knowledge transfer is to ensure an accurate approximation of the additional control input, while maintaining Lipschitz-based stability guarantees for the combined control system.

Finally, the paper provides a comparative analysis in terms of tracking error, network accuracy, and the resulting Lipschitz constants. To ensure a fair comparison, a classical feedforward neural network is trained using the same parameters and cost functions as the KAN.

### *1.1 Structure of the paper*

In the first step, a reinforcement learning-based combined control framework is implemented. Using the trained RL-based combined controller, an extended simulation is carried out for data collection purposes. Based on the collected dataset and a density estimation procedure, the training of the KAN-based algorithm is performed. The training of the KAN-based component is carried out via supervised learning in a teacher-student structure. The training is performed iteratively, where the mini-batches are selected based on the density of the dataset and the approximation error of the KAN-based network. At every iteration step, the density is updated according to the error. Finally, the KAN-based algorithm is implemented in the control problem to evaluate the performance of the combined solution.

The paper is organized as follows. In section 2, the KAN-based framework is presented. The knowledge distillation procedure, including the optimization method, is detailed in Subsection 2.3. Moreover, Section 3 shows the stability analysis, while simulation results are provided in Section 4. Finally, the paper is concluded in Section 5.

## 2. KOLMOGOROV - ARNOLD NETWORKS

The purpose of this section is to detail the Kolmogorov-Arnold Networks (KANs) and to present the training procedure of the network. KAN is a novel neural network structure whose foundation is based on the Kolmogorov-Arnold representation theorem. Unlike classical neural networks, in KAN-based models, the activation functions are not fixed, but are learned during the optimization process. The connections between the neurons are also derived from the Kolmogorov-Arnold theorem. The main advantage of this neural network over classical approaches lies in its improved data efficiency and interpretability. In general, the mapping between the input and output data points can be formulated as:

$$\hat{y}(x) = \sum_{m=1}^{M} \psi_m \left( \sum_{i=1}^{N} \phi_{i,m}(x_i) \right) \tag{1}$$

where $x \in \mathbb{R}^N$ is the input vector, $x_i$ denotes its $i$-th component, and $N$ is the input dimension. The number of KAN components is given by $M$, the number of spline basis functions is denoted by $K$, while $k$ defines the order of the splines. To obtain smooth and continuous approximations, B-splines are employed in the KAN-based framework. The KAN is constructed using the following inner functions:

$$\phi_{i,m}(x_i) = \sum_{j=1}^{K} c_{i,m}^{(j)} B_j^{(k)}(x_i), \tag{2a}$$

$$\psi_m(z_m) = \sum_{j=1}^{K} c_{j,m} B_j^{(k)}(z_m), \tag{2b}$$

where $z_m$ denotes the intermediate variable defined as $z_m = \sum_{i=1}^{D} \phi_{i,m}(x_i)$, which serves as the input to the function $\psi_m$. Moreover, the spline coefficients, which are calculated during the training, are denoted by $c_{i,m}^{(j)}$ and $c_{j,m}$.

## 2.1 Training of the KAN

This subsection aims to present the supervised training of the KAN-based part of the control algorithm. The KAN is trained based on a Deep Deterministic Policy Gradient (DDPG) algorithm, which previously learned the optimal control policy through reinforcement learning (Sumiea et al. (2024)). In the student-teacher training structure, the RL-based algorithm is treated as the teacher network, and the goal is to transfer its knowledge to the KAN. The main advantage of the proposed framework comes from the combination of the exploration capability of the DDPG algorithm with the strong generalization performance and efficient functional representation provided by the KAN. During the training process, the optimal parameters of the spline coefficients, denoted by $\theta = c_{d,m}^{(j)}, c_m^{(j)}$, are determined. The loss function is formulated in quadratic form as follows:

$$\mathcal{L}_{\mathrm{MSE}}(\theta) = \frac{1}{N}\sum_{i=1}^{N}\Big(\mu_{\theta_a^*}(x_i) - \hat{y}(x_i,\theta)\Big)^2, \tag{3}$$

where $\mu_{\theta_a^*}$ represents the actor network, whose optimal parameters are denoted by $\theta_a^*$.

## 2.2 Kernel Density Estimation

In this subsection, the selection of the training data points is presented. To guarantee a high performance level of the student network, the training must be carried out over the entire operational range of the dynamical system. First, the closed-loop system is simulated under different conditions to explore the operational range, and the corresponding system states are recorded. Then, using the collected data, an $N$-dimensional convex hull can be defined as $\mathcal{M}_x = \{x_1, x_2, \ldots, x_N\} \subset \mathbb{R}^N$, where $x_i$ denotes an arbitrary measurement vector. In principle, uniform sampling can be performed within the constructed convex hull by placing an equal number of grid points along each coordinate axis. However, it is important to highlight that assigning $k$ grid points per dimension results in $k^N$ sampling points in an $N$-dimensional space. As $N$ increases, the size of the training dataset grows exponentially, which makes the approach computationally infeasible. To overcome this limitation, Kernel Density Estimation (KDE) is used on the measured data to identify a diverse set of training points, which cover appropriately the full operational range. For a set of $M$ measurement points, the kernel density estimate at an arbitrary point can be expressed as Parzen (1962):

$$\hat{f}(x) = \frac{1}{M}\sum_{i=1}^{M} K_{\mathbf{H}}(x - x_i) \tag{4}$$

where $K_{\mathbf{H}}$ denotes a multivariate kernel function parameterized by the bandwidth matrix $\mathbf{H} \in \mathbb{R}^{N\times N}$. In the proposed method, a multivariate Gaussian kernel is applied as follows:

$$K_{\mathbf{H}}(\mathbf{u}) = \frac{1}{(2\pi)^{N/2}|\mathbf{H}|^{1/2}} e^{\left(-\frac{1}{2}\mathbf{u}^T\mathbf{H}^{-1}\mathbf{u}\right)} \tag{5}$$

where $\mathbf{u} = \mathbf{x} - \mathbf{x}_i$, and $|\mathbf{H}|$ denotes the determinant of the bandwidth matrix. In this work, the bandwidth matrix is assumed to be diagonal. For bandwidth selection, Silverman's rule of thumb is employed Läuter (1988).

## 2.3 Limitation of the gradients

During the knowledge distillation process, the aim is to approximate the control policy learned by the DDPG algorithm with high accuracy. In addition to accurate computation of the additional control input, smoothness of the generated signal is also enforced in order to avoid sharp changes in the control action. This behaviour is advantageous for safe system operation and stability analysis. To involve smoothness in the training process, the derivatives of the B-spline functions are included in the optimization process. Using the chain rule, the partial derivative of the KAN output with respect to the $i^{th}$ input can be formulated as:

$$\frac{\partial \hat{y}}{\partial x_i} = \sum_{m=1}^{M} \psi_m'(z_m)\,\phi_{i,m}'(x_i). \tag{6}$$

For bounded inputs, a KAN constructed using B-splines is Lipschitz continuous Lebair and Shen (2015). Accordingly, the Lipschitz constant ($L$) of the KAN can be defined at each iteration step as $|\hat{y}(x_1) - \hat{y}(x_2)| \leq L\,|x_1 - x_2|, \quad \forall x_1, x_2 \in \mathcal{X}$. Based on this, an upper bound on the Lipschitz constant of the KAN can be derived as:

$$L(\theta) \leq \sum_{m=1}^{M}\left(\max_z |\psi_m'(z)| \sum_{i=1}^{D} \max_x \left|\phi_{i,m}'(x)\right|\right) \tag{7}$$

The goal during the computation of the parameters of the KAN is to reach an accurate estimation and, also a low Lipschitz constant, therefore, the optimization problem can be formed as:

$$\min_\theta \mathcal{L}_{\mathrm{MSE}}(\theta) + \lambda L(\theta), \quad \text{s.t.} \tag{8a}$$

$$|c_{d,m}^{(j)}| \leq C_{\max},\ |c_m^{(j)}| \leq C_{\max}, \tag{8b}$$

where upper bounds ($C_{\max}$) are applied on the coefficients to enforce smoothness over the entire operating range.

## 2.4 Error-based density update and sampling

During the training process, the goal is to approximate the output of the previously trained RL network $\mu_{\theta_a^*}(x)$ using the KAN-based model $\hat{y}(x,\theta)$. For a chosen mini-batch $\{x_i\}_{i=1}^N$, the error between the teacher and the student network is computed as:

$$e_i = \left\|\hat{y}(x_i,\theta) - \mu_{\theta_a^*}(x_i)\right\|^2. \tag{9}$$

The training loss is computed as the mean squared error (MSE):

$$\mathcal{L}_{\mathrm{MSE}}(\theta) = \frac{1}{N}\sum_{i=1}^{N} e_i. \tag{10}$$

The goal during the iteration is to highlight the regions where the accuracy of the student network is low. Therefore, in addition to the probability values, the error values are updated after the evaluation of the given mini batch. The density is given by $p^{(k)}(x_i)$ at the iteration $k$. After the evaluation of the student network along the selected data points, the density function is updated as a convex combination of the previously computed density values and the error:

$$p(x_i) \leftarrow (1-\alpha)\,p(x_i) + \alpha \cdot \min\left(\frac{1}{e_i},\, e_{\max}\right), \tag{11}$$

where $\alpha \in (0,1)$ is a weighting parameter, which balances the effect of the density and the error value. Moreover,

$e_{\max}$ defines the upper bound applied to the inverse error term, ensuring numerical stability in the neighborhood of zero error values. Finally, the normalized error is given as:

$$\tilde{e}_i = \frac{e_i}{\sum_{j=1}^{N} e_j}. \tag{12}$$

After computing the new density values, the whole density field is renormalized to ensure that it defines a valid probability distribution:

$$\sum_{i=1}^{N} p^{(k+1)}(x_i) = 1. \tag{13}$$

At the next iteration step, the data points in the mini-batch is selected using the probability distribution inversely proportional to the density:

$$P(x_i) \propto \frac{1}{p(x_i)}, \tag{14}$$

This strategy guarantees that data points with low density values and high approximation errors are selected more frequently. This helps achieve improved convergence and increases the accuracy of the approximation.

## 3. DESIGN OF LQR WITH THE ANALYSIS OF THE MAXIMUM LIPSCHITZ CONSTANT

This section aims to present the stability analysis of the proposed control method. The state-space representation of a linear time-invariant (LTI) system can be formulated as:

$$\dot{x}(t) = Ax(t) + Bu(t), \qquad x(0) = x_0, \tag{15}$$

where $x(t) \in \mathbb{R}^n$ is the state vector and $u(t) \in \mathbb{R}^m$ denotes the control input of the system. Using the nominal model of the system, the LQR controller can be designed, and with the computed feedback gain ($K_L$), the closed-loop system can be expressed as:

$$\dot{x} = (A - BK_L)x + f_{KAN}(x, \theta). \tag{16}$$

Using the quadratic Lyapunov function $V(x) = x^T P x, P = P^T \succ 0$, the time derivative of $V$ is given by:

$$\begin{aligned}\dot{V} = x^T\big((A - BK_L)^T P + P(A - BK_L)\big)x + \\ 2\, x^T P\, f_{KAN}(x, \theta).\end{aligned} \tag{17}$$

Using the Algebraic Riccati equation and the corresponding feedback gains, the following expression can be derived (Gießler et al. (2025)):

$$(A - BK_L)^T P + P(A - BK_L) = -\big(Q + K_L^T R K_L\big). \tag{18}$$

Using the Lyapunov derivative (17) and the notation that $Q_c = (Q + K_L^T R K_L)$, the Lyapunov derivative can be formalized as:

$$\dot{V} = -x^T Q_c x + 2\, x^T P f_{KAN}(x, \theta). \tag{19}$$

Using Cauchy-Schwarz inequality for the term $2x^T P f_{KAN}(x, \theta)$, the following expression can be defined:

$$|2\, x^T PB f_{KAN}(x, \theta)| \le 2\, \|x\|\, \|PB\|\, \|f_{KAN}(x, \theta)\|, \tag{20}$$

where $\|\cdot\|$ denotes the Euclidean norm. Assuming that $f_{KAN}$ is Lipschitz continuous with the maximum value of the constant $L_{max}$, i.e.,

$$\|f_{KAN}(x)\| \le L_{max} \|x\|, \tag{21}$$

Using the Lipschitz bound, the following quadratic bound can be defined:

$$\begin{aligned}\dot{V} &\le -\lambda_{\min}(Q_c)\, \|x\|^2 + 2\, \|PB\|\, L_{max}\, \|x\|^2 \\ &= -\big(\lambda_{\min}(Q_c) - 2\|PB\| L_{max}\big)\, \|x\|^2,\end{aligned} \tag{22}$$

where $\lambda_{\min}$ gives the smallest eigenvalue of the matrix $Q_c$. For asymptotic stability the coefficient of $\|x\|^2$ must be positive:

$$\lambda_{\min}(Q_c) - 2\|PB\| L_{max} > 0. \tag{23}$$

This results in the following inequality, considering the Lipschitz constant (Cui et al. (2024)):

$$L_{max} < \frac{\lambda_{\min}(Q_c)}{2\|PB\|}. \tag{24}$$

This indicates that, once the LQR controller has been designed, the closed-loop system can be analyzed to determine the maximum allowable Lipschitz constant ($L_{\max}$). This bound can then be used to assess the stability of the closed-loop system in the presence of the KAN-based additional control input.

## 4. SIMULATION EXAMPLE

This section provides a validation of the proposed approach. It begins with a brief overview of the dynamical system under consideration, followed by a description of the training procedure applied to the RL-based algorithm. As noted in the paper, the details of the DDPG training are not provided. After the training of the RL network, knowledge distillation is performed. It is important to highlight that during the training of the RL-based algorithm, no constraints or minimization are applied on the maximum Lipschitz constant, since the primary objective is to achieve a high-performance control policy. Because the Lipschitz constant of a classical neural network is not minimized during training, knowledge distillation is also applied to a classical feedforward neural network to ensure a fair comparison with the KAN-based method. The training parameters and dataset are identical for both cases.

For validation purposes, the combined control strategy is implemented on a cart-mounted double inverted pendulum system, which exhibits highly nonlinear dynamics. For the control design, the system is linearized around its equilibrium point. The state vector of the system is defined as $x = [s, \dot{s}, \phi_1, \dot{\phi}_1, \phi_2, \dot{\phi}_2]^T$, where $s$ denotes the position of the cart, $\phi_1$ and $\phi_2$ are the angles of the arms, and the control input is the force applied to the cart, i.e., $u = F$.

Based on the linearized model, an LQR controller is designed (Abut (2023)). The parameters of the considered system are as follows: $M = 0.6\,\mathrm{kg}$, $m_1 = 0.25\,\mathrm{kg}$, $m_2 = 0.2\,\mathrm{kg}$, $L_1 = 0.25\,\mathrm{m}$, $L_2 = 0.2\,\mathrm{m}$, where $M$ denotes the mass of the cart, $L_1$ and $L_2$ are the lengths of the links, and $m_1$ and $m_2$ are their corresponding masses. Finally, the weigths of the LQR controller are selected to: $Q = [40, 200, 200, 40, 80, 80]$ and $R = 0.01$. Based on the analysis of this system and the feedback controller, the maximum allowed Lipschitz value is $L_{max}$=0.995.

### *4.1 Knowledge distillation*

The training process is conducted on 4000 data points. The control problem addressed by the combined structure is the stabilization of a double inverted pendulum. In addition, the cart is required to track a reference position that varies randomly. During training, the mini-batch size is set to $B = 256$, while the Lipschitz regularization parameter is $\lambda = 0.01$. The resulting optimization problem is solved using the Sequential Quadratic Programming (SQP) algorithm implemented in the MATLAB environment. Finally, the coefficient bound of the KAN-based component is set to $|C_{\max}| = 0.75$. The classical neural network is also trained in MATLAB using the built-in Adaptive Moment Estimation (ADAM) optimization algorithm to iteratively optimize the network parameters during backpropagation. The main parameters of the training are set to the same as in the case of the KAN training.

### *4.2 Compare results between classical neural network and KAN*

This subsection presents a comparison between the classical and KAN-based control structures. Before the evaluation, the effect of the main parameters of the KAN is analyzed. In this framework, the parameter $k$ defines the degree of the B-splines, $M$ determines the number of additive terms in the model, while $K$ specifies the resolution of the splines. In general, using a large number of parameters may lead to overfitting, whereas a small number of parameters may result in underfitting. The goal is to examine how the structure of these networks influences their accuracy and the resulting Lipschitz constants. Therefore, the following structures are considered in the test scenario:

- *Classical, feedforward neural network:* In this case, the ReLu activation function is used, and the sizes of the hidden layers and neurons are selected as follows: $N_{hid} = 1-2-3, N_{neur} = 32-64-128$. Note that the selected number of neurons is not varied in one setup, which results in 9 different neural network structures.
- *KAN*: The results are analyzed for the following parameter combinations: $M = 3, 4, 5$, $k = 2, 3, 4$, and $K = 4, 5, 6$.

The normalized Mean Squared Errors (MSE) and the Lipschitz constants are presented in Figure 1. The maximum value of the Lipschitz constant, computed according to (24), is also shown. Blue crosses represent the KAN-based solutions, while red crosses represent the classical neural networks. It can be seen that the KAN-based solutions achieve a better trade-off between the Lipschitz constant and accuracy. For further comparison, one KAN-based network and one classical neural network are selected that satisfy the maximum Lipschitz constraint while also providing the lowest error.

The neural networks are compared in terms of tracking accuracy, since the primary objective of this control problem is to minimize tracking errors. Figure 2 shows histograms composed of the error values saved during the simulations. The dimensions of the tracking errors shown in Figure 2 are given in centimeters (cm). During this test, the same nonlinear dynamical model is used, on which the RL-based solution was trained. Three main cases can be examined: First, the performance of the LQR controller enhanced with the RL-based training is compared to the purely LQR-based solution. Note that this network served as the teacher network during the knowledge distillation process. Second, the results of the LQR combined with a classical neural network are presented. Finally, the KAN + LQR-based solution is shown. It can be observed that, in all cases, using neural networks as an additional control input decreases the tracking errors.

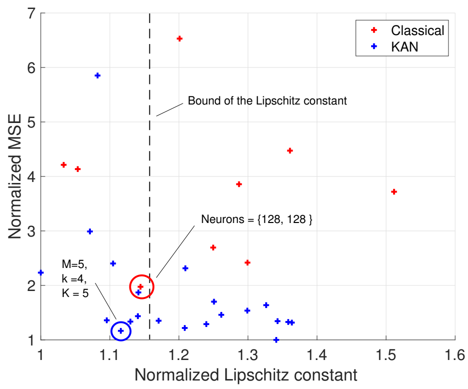


Fig. 1. Results of the different neural network structures

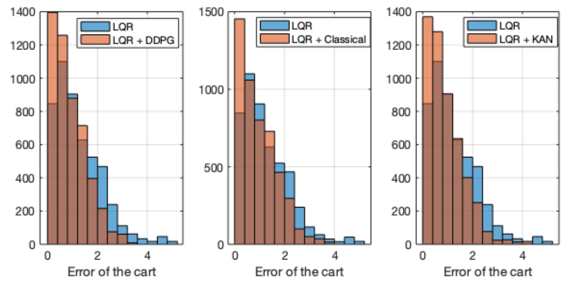


Fig. 2. Histogram of the tracking errors

In Table 1, the results are summarized. In this case, the Sum of Squared Errors (SSE) and the Standard deviations (Std) are normalized to achieve better comparison. The best performance level is reached using the RL-based algorithm, in which the Lipschitz constant is not minimized during the training, and serves as a teacher network. However, it can be seen that the KAN-based solution outperforms the case when the classical neural network is used.

| | LQR | DDPG | Classical | KAN |
|---|---|---|---|---|
| SSE | 1.4098 | 1 | 1.0820 | 1.0322 |
| Std | 1.3835 | 1 | 1.11 | 1.0574 |

Table 1. Computed error values

Finally, the outputs of the distilled networks are compared to each other. Since the goal of the distillation process is to reproduce the behavior of the teacher network, the outputs of both the KAN-based and classical neural network models are saved and plotted, see: Figure 3. The results indicate that the KAN-based solution more accurately matches the output of the teacher network than the classical neural network. This leads to improved overall accuracy in the resulting control algorithm. After normalizing the MSE values of the control signals, it comes out that the KAN-based solution provides 2.53 times more accurate results.

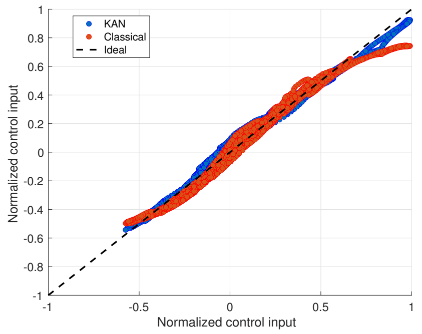


Fig. 3. Comparing the control inputs

### *4.3 Summarizing the comparison between the networks*

In this subsection, the results are summarized. The comparison is made between the selected neural networks, as shown in Figure 1. The KAN-based solution uses only 225 parameters, whereas the classical feedforward neural network has 18,177 parameters. Moreover, the classical network reaches a poorer balance between the Lipschitz constant and tracking accuracy compared to the KAN-based solution when considering the computed bound. Furthermore, the analysis shows that the Lipschitz constant of the teacher network (DDPG) is nearly 40 times higher than that of the selected KAN-based solution. Finally, the output of the KAN-based network more accurately reproduces the output of the teacher network than the classical neural network. This demonstrates that, despite its simplified structure, the KAN-based solution reaches better performance for the control problem presented in the paper.

## 5. CONCLUSION

This paper presents a combined control structure that integrates a classical LQR feedback controller with a KAN-based machine learning algorithm. First, an RL-based additional controller is trained to improve the tracking performance of the system. Using the trained RL policy, a knowledge transfer process is performed, with the aim of compressing the learned policy into a less complex and more structured neural network. During this process, not only is the approximation error minimized, but the Lipschitz constant of the network is also reduced. The results show that the KAN-based approach achieves nearly the same level of performance as the RL-based method while also ensuring closed-loop stability. Furthermore, the KAN-based solution requires fewer parameters compared to a classical neural network. By adopting a KAN-based approach for the additional control input, the Lipschitz constant of the reinforcement learning network can be significantly reduced. Future research will focus on extending this framework. The main goal is to perform stability analysis across varying operating conditions, with which a less conservative and more adaptive control strategy can be reached. Another important direction is the incorporation of system uncertainties into the design process to ensure reliable closed-loop operation over a wider range of conditions.

## REFERENCES


Abut, T. (2023). Optimal lqr controller methods for double inverted pendulum system on a cart. *Dicle Üniversitesi Mühendislik Fakültesi Mühendislik Dergisi*, 14(2), 247–255. doi:10.24012/dumf.1253331.

Cui, L., Jiang, Z.P., and Sontag, E.D. (2024). Small-disturbance input-to-state stability of perturbed gradient flows: Applications to lqr problem. *Systems and Control Letters*, 188, 105804.

Dong, C., Zheng, L., and Chen, W. (2025). Kolmogorov-arnold networks (kan) for time series classification and robust analysis. 15390. ADMA 2024.

Gießler, A., Malan, A.J., and Hohmann, S. (2025). Bridging continuous-time lqr and reinforcement learning via gradient flow of the bellman error. In *2025 IEEE 64th Conference on Decision and Control (CDC)*, 3761–3768. IEEE.

Kiumarsi, B., Vamvoudakis, K.G., Modares, H., and Lewis, F.L. (2018). Optimal and autonomous control using reinforcement learning: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 29(6), 2042–2062. doi:10.1109/TNNLS.2017.2773458.

Lebair, T.M. and Shen, J. (2015). Uniform lipschitz property of nonnegative derivative constrained b-splines and applications to shape constrained estimation. URL `https://arxiv.org/abs/1510.05314`.

Lelkó, A. and Németh, B. (2024). Optimal motion design for autonomous vehicles with learning aided robust control. *IEEE Transactions on Vehicular Technology*, 73(9), 12638–12651. doi:10.1109/TVT.2024.3394699.

Liu, Z., Wang, Y., Vaidya, S., Ruehle, F., Halverson, J., Soljačić, M., Hou, T.Y., and Tegmark, M. (2025). Kan: Kolmogorov-arnold networks. URL `https://arxiv.org/abs/2404.19756`.

Läuter, H. (1988). Silverman, b. w.: Density estimation for statistics and data analysis. chapman & hall, london, new york 1986, 175 pp. *Biometrical Journal*, 30(7), 876–877. doi:https://doi.org/10.1002/bimj.4710300745.

Mehrmann, V. (1991). *The Autonomous Linear Quadratic Control Problem: Theory and Numerical Solution*, volume 163 of *Lecture Notes in Control and Information Sciences*. Springer-Verlag, Berlin, Heidelberg.

Nicodemus, J., Kneifl, J., Fehr, J., and Unger, B. (2021). Physics-informed neural networks-based model predictive control for multi-link manipulators.

Parzen, E. (1962). On estimation of a probability density function and mode. *The Annals of Mathematical Statistics*, 33(3), 1065–1076.

Qin, Y., Zhang, W., Shi, J., and Liu, J. (2018). Improve pid controller through reinforcement learning. In *2018 IEEE CSAA Guidance, Navigation and Control Conference (CGNCC)*, 1–6.

Schmidt-Hieber, J. (2021). The kolmogorov?arnold representation theorem revisited. *Neural Networks*, 137, 119–126. doi:https://doi.org/10.1016/j.neunet.2021.01.020.

Sumiea, E.H., Abdulkadir, S.J., Alhussian, H.S., Al-Selwi, S.M., Alqushaibi, A., Ragab, M.G., and Fati, S.M. (2024). Deep deterministic policy gradient algorithm: A systematic review. *Heliyon*, 10(9), e30697.

Yoo, J., Jang, D., Kim, H.J., and Johansson, K.H. (2021). Hybrid reinforcement learning control for a micro quadrotor flight. *IEEE Control Systems Letters*, 5(2), 505–510. doi:10.1109/LCSYS.2020.3001663.