The Validation of a Lyapunov Deep Learning Algorithm for an Inverted Pendulum Cart System

Student: Kristan Dötsch | k.l.c.dotsch@student.rug.nl | First supervisor: Bayu Jayawardhana | Second Supervisor: Mauricio Munoz Arias | June 2021

***Abstract*— In the study of nonlinear system stability, a deep learning algorithm was constructed in order to enlarge a system's stability region. Previous study has shown that the deep learning algorithm in [2] was able to enlarge the stability region for a nonlinear system consisting of two state variables. This study is about the implementation of a nonlinear system, consisting of four state variables, into the deep learning algorithm in order to enlarge the stability region. The inability of the deep learning algorithm to enlarge the stability region of a system consisting of four state variables is discussed. The study is finalized with a recommendation for further research.**

***Index Terms*— Nonlinear System, Lyapunov Stability, Deep Learning Algorithm, LQR Controller, Cart Inverted Pendulum System.**

I. Introduction

Most of modern day robots are nonlinear systems, where stability of these systems is critical for its use. As a consequence of the human-interference in certain robotics fields, like in healthcare [1], it is of importance that all movement is predictable, in order to prevent operational casualties. In the current study on the stabilization of nonlinear systems [2], the most classical settings in nonlinear control design are investigated, using neural networks with a deep learning approach.

A deep learning algorithm can be used to compute the stability region of nonlinear systems. The deep learning algorithm that was developed in [2] consists of a learner and a falsifier. The learner attempts to find the control and Lyapunov functions, while the falsifier finds counterexamples of instability and therefore guides the learner towards solutions. Once the falsifier is unable to find examples of instability, the system is considered Lyapunov stable. Using the developed deep learning algorithm instead of existing methods like sum-of-squares (SOS) and semidefinite programming (SDP) [3], the region of attraction is enlarged and therefore, the robustness of the system is improved.

In order to evaluate the efficacy of the deep learning algorithm, a nonlinear system has been investigated and the corresponding equations of motion, describing the dynamics of a nonlinear system, were determined. Implementing the equations of motion and the corresponding LQR controller into the deep learning algorithm [2], the region of attraction is calculated. The nonlinear system being investigated is the Cart-Inverted-Pendulum-System (CIPS) [4], due to the simplistic equations of motion and nonlinear nature of the system. A schematic display of the CIPS and the parameter values for this system are given in section III of this paper. The writers of [2] focused on the applicability of the deep learning algorithm on nonlinear systems with two state variables, as is discussed in section II, preliminaries. The CIPS system has four state variables. This study will be the first application of a nonlinear system consisting of four state variables, being applied to a deep learning algorithm.

Fig. 1. SpaceX's Falcon 9-R rocket after landing on a platform at sea.

CIPS is an important type of nonlinear system, as detailed in [5]. The subject of paper [5] is the controllability of the landing of the SpaceX Falcon 9-R rocket, which can be seen in figure 1. The importance of stability is given in this specific example, with the exact same dynamical movement of its parts, due to the objective set by the operating company $SpaceX$. The objective is to reuse rockets and have them land on platforms at sea. The instability of the system would result in a crash of the rocket. Having more knowledge on the stability region of this system could result in the possibility of landing the rocket in less ideal conditions, e.g., stormy weather, and therefore increase the usability of this technology.

This study is structured as follows: in section II, the preliminaries, concerning the theoretical background of certain aspects are described. Section III, the implementation and elaboration on key concepts, in which a more detailed description and more background is given on important subjects. In section IV, the experiment, the result of the implementation of the CIPS on the deep learning algorithm is discussed. Section V will provide a discussion, where the expected outcome is compared to the actual outcome. Finally, in section VI the conclusion of this research is given.

II. Preliminaries

In order for the deep learning algorithm in [2] to operate, the nonlinear system being researched must be a Lyapunov stable system. The following paragraph will give some theoretical background on Lyapunov functions. Section III will give the mathematical proof that the CIPS is a Lyapunov function. Furthermore, the conclusions of previous works like [2] and [8] will be discussed in this chapter.

A. Lyapunov Theory

For a non-negative function V to be Lyapunov stable, one of (2) or (3) must be true, as is explained in [6].

$\dot{V}(x)$ represents the time derivative of V along the trajectories of the system dynamics and $F(x)$ represents the displacement derivative of function V.

$$\dot{V}(x) = \frac{dV}{dx}\dot{x} = \frac{dV}{dx}F(x) \tag{1}$$

The stability of the origin at $x = 0$, which is the starting point of the V trajectory, is characterized by:

$$V(x) > 0, \dot{V}(x) \leq 0 \rightarrow x = 0 \quad \text{is stable} \tag{2}$$

$$V(x) > 0, \dot{V}(x) < 0 \rightarrow x = 0 \quad \text{asymptotically stable} \tag{3}$$

If V satisfies one of the conditions in (2) or (3), we say that function V is Lyapunov stable.

A function being Lyapunov stable means that the system is stable near the origin [6] and has a partially known stability region.

B. Other Work

The construction of deep learning algorithm was due to the lacking safety guarantees for nonlinear, safety-critical systems. In [8], the safety region for a standard simulated inverted pendulum is determined to become greater, as well as the given conclusion that their algorithm is applicable to a great range of autonomous nonlinear systems in uncertain and safety-critical environments. More information about the application of the simple pendulum in the deep learning algorithm is given in section IV, experiment.

In using the deep learning algorithm developed in [8], the writers of [2] have extended the range of applicable nonlinear systems in real-life situations by applying the algorithm on the Caltech ducted fan in hover mode and the N-link planar Robot balancing, which can be seen in figure 2. In this study, the stability regions for both systems were enlarged.

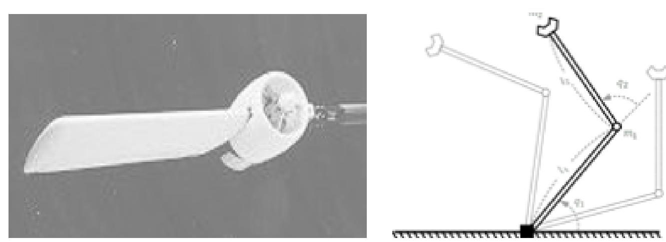

Fig. 2. (left): the Caltech ducted fan and (right) a 2-link planar robot.

III. Implementation and Elaboration on Key Concepts

In this section, more information on the nonlinear system of concern in this paper is given. As well as the corresponding equations of motion and the controller used to regulate the movement.

A. Equations of Motion

The equations of motion represent the movement of a dynamical system, by which the deep learning algorithm can eventually compute the region of attraction. The equations of motion for the Cart Inverted Pendulum System are, based on the schematic representation in figure 3, given in [7] and by (4) and (5)

$$(M + m)\ddot{x} - ml\ddot{\phi}cos\phi + ml\dot{\phi}^2 sin\phi = F \tag{4}$$

$$l\ddot{\phi} - gsin\phi - \ddot{x}cos\phi = 0 \tag{5}$$

In the equations above, x is the cart displacement, ϕ is the angular displacement of the pendulum, F is the external force applied to the cart by the controller, M is the mass of the cart, m is the mass of the pendulum, l is the pendulum length and g the gravitational constant. A schematic representation is given below.

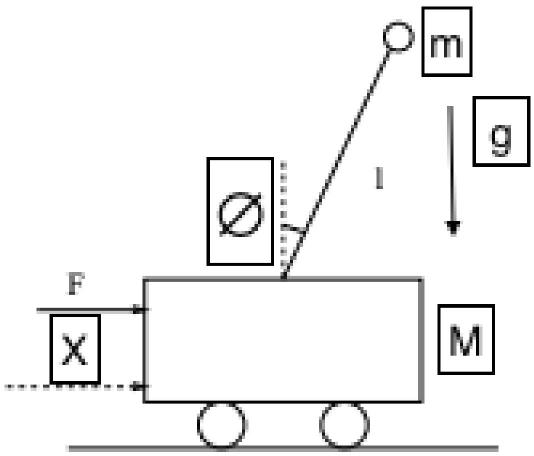

Fig. 3. Schematic representation of the CIPS and all forces applied.

B. Proof of Lyapunov Stability

The two-dimensional cart inverted pendulum system is proven to be Lyapunov stable in [9], in the following manner, using the theory described in the previous section, preliminaries.

Stabilizing the variables ϕ and $\dot{\phi}$ is done by introducing function V, as:

$$V(\phi,\dot{\phi}) = \frac{1}{2}(k_1 cos^2\phi - 1)\dot{\phi}^2 + (1 - cos\phi) \tag{6}$$

In which constant $k_1 > 1$. Therefore $V(\phi,\dot{\phi})$ becomes a positive definite, for all $|\phi| < \tilde{\phi} < \pi/2$,

where:

$$\tilde{\phi} = cos^{-1}(\sqrt{\frac{1}{k_1}})(7)$$

The time derivative of V is then given by:

$$\dot{V}(\phi,\dot{\phi}) = \dot{\phi}cos\phi(\varepsilon_0\alpha(\phi) + k_1\beta(\phi,\dot{\phi})), \tag{8}$$

where:

$$\alpha(\phi) = 1 - k_1 cos^2\phi, \tag{9}$$

$$\beta(\phi,\dot{\phi}) = (-\dot{\phi}^2 + cos\phi)sin\phi \tag{10}$$

The controller output is then given by:

$$\varepsilon_0 = -\frac{1}{\alpha(\phi)}(\dot{\phi}cos\phi + k_1\beta(\phi,\dot{\phi})), \tag{11}$$

Makes variables ϕ and $\dot{\phi}$ converge asymptotically to zero, because ε_0 produces:

$$\dot{V}(\phi,\dot{\phi}) = -\dot{\phi}^2 cos^2\phi \tag{12}$$

As is shown in (12), the derivative of V converges asymptotically to zero and therefore is a Lyapunov function.

C. Theory on Controller

Controlling the nonlinear CIPS is done with an LQR controller. As is described in [10], the LQR controller is used for complex nonlinear systems with strict performance requirements, consisting of matrices A, B and Q and value R. Matrices A and B depend on the dynamics of a particular system and can therefore not be adjusted. The specific values for A and B are found in appendix B. The controller is determined as the optimum of a linear quadratic cost function where the system dynamics are represented in a linearized state space matrix A and input vector B of the nonlinear system. The cost function is optimized using a weighting matrix Q and weighting vector R as parameters, rendering a vector K as usable controller values by using the $Matlab$ command $K = lqr(A, B, Q, R)$.

D. State Space Equation and LQR Controller Values

The equations of motion mentioned in the 'CIPS Equations of motion' paragraph can be rewritten in first order equations, in the state space form, as found in [4]:

$$\begin{bmatrix} \dot{z_1} \\ \dot{z_2} \\ \dot{z_3} \\ \dot{z_4} \end{bmatrix} = \begin{bmatrix} z_2 \\ \frac{\frac{1}{2}gmsin2\phi - lm(\dot{\phi}^2)sin\phi}{(msin^2\phi)+M} \\ z_4 \\ \frac{g(M+m)sin\phi - lm(\dot{\phi}^2)sin\phi cos\phi}{l(msin^2\phi+M)} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{msin^2\phi+M} \\ 0 \\ \frac{cos\phi}{l(msin^2\phi+M)} \end{bmatrix} [F] \tag{13}$$

The following table contains the values of the parameters of importance:

An initial LQR gain vector, as is shown in [4], is given by:

$$K_{initial} = \begin{bmatrix} -1000.00 & -946.79 & 4353.27 & 803.53 \end{bmatrix} \tag{14}$$

The computation of K can be found in Appendix B.

Combining the values given in the table above and (13) and (14), a simulation is made in $Matlab$, to verify the stability of the CIPS with these variables. The initial K vector will be improved in section IV, experiment.

Parameter	Values
Mass Cart (M)	1 kg
Mass Pendulum (m)	0.5 kg
Length Pendulum (l)	0.7 metres
Gravity (g)	9.81 m/s^2

TABLE I
PARAMETER VALUES

E. Proof of State-Space equations and Controller

Using the $Matlab$ file $main$, which can be found in [13], the controlled CIPS is simulated for eight seconds, as can be seen in figure 4.

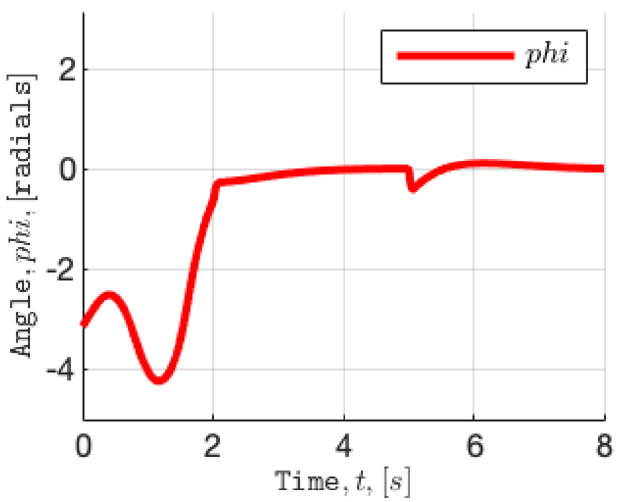

Fig. 4. Output of angle phi, with respect to time, of the controlled CIPS, using the initial controller values for K.

The starting position x_0 is in the downright position with angle $\phi = -\pi$. Controlling the inverted pendulum corresponds to getting the angle ϕ in a stable upright position, with a value of 0 radians.

Figure 4 shows five different stages. The first being the starting position $-\pi$ at $t = 0$, as discussed above. Secondly, the controller alters the angle of the pendulum, in order to reach the upright position, in the time period $0 < t < 2$. The third phase consist of the stabilization of the system at time period $2 < t < 5$. In the fourth phase, at $t = 5$, the system is externally effected for the last time, resulting in a small angle deviation. Finally at the fifth stage, the angle is reduced to 0 and considered stable.

IV. EXPERIMENT

The experimental stage of this research consists of improving the LQR controller, improving the parameters of the deep learning algorithm and the implementation of the CIPS in the algorithm in combination with the improved controller values and algorithm's parameters.

A. Improved LQR controller

Matrices A and B depend on the system's dynamics and can therefore not be altered in the simulation setup. Simulations have shown, by using the $Matlab$ file $main$ in [13], that the R value of 0.0001 is a value sufficient for this research. Therefore, the focus will be on the four Q values, Q_{11}, Q_{22}, Q_{33} Q_{44} as shown below in the sensitivity matrix Q. All four values for Q will be simulated one by one until the outcome of the stability graph is useful for this research.

$$Q = \begin{bmatrix} Q_{11} & 0 & 0 & 0 \\ 0 & Q_{22} & 0 & 0 \\ 0 & 0 & Q_{33} & 0 \\ 0 & 0 & 0 & Q_{44} \end{bmatrix} \tag{15}$$

The initial values for Q are: $Q_{11} = 100$, $Q_{22} = 1$, $Q_{33} = 1000$ and $Q_{44} = 1$.

Changing Q_{11} to 0.001 results in the graph shown in figure 5.

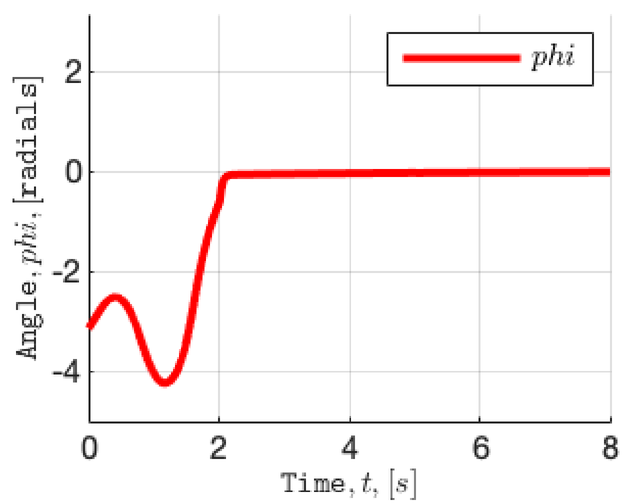

Fig. 5. Output of angle phi, with respect to time, of the controlled CIPS, using the value $\boldsymbol{Q_{11} = 0.001}$.

The result of adjusting Q_{11} makes the system reach stability faster at $t = 2$ and is not affected by the external force at $t = 5$, therefore the value of Q_{11} is seen as useful for this research. From here on we will use $Q_{11} = 0.001$ in every simulation.

Simulations have shown that the alteration of Q_{22} does not effect the system's stability. Therefore $Q_{22} = 1$ is considered good enough for this study.

Figure 6 shows the graph for $Q_{33} = 10000$, clearly showing improvement in stability rate at $t = 2$ seconds.

Therefore, $Q_{33} = 10000$ will be used for further simulations.

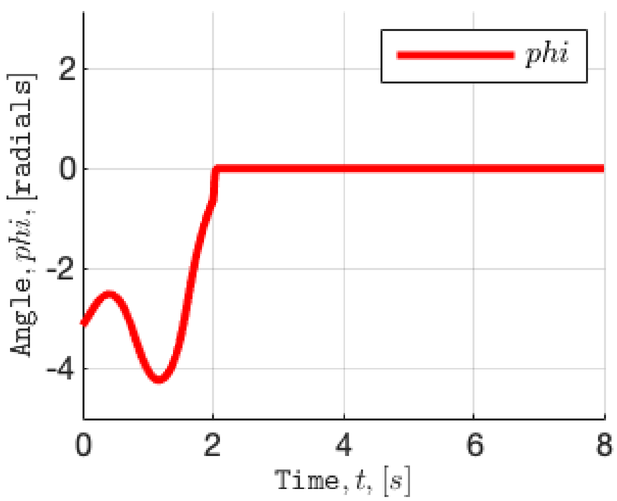

Fig. 6. Output of angle phi, with respect to time, of the controlled CIPS, using the value $\boldsymbol{Q_{33} = 10000}$

Value $Q_{44} = 0.01$ improves the stability of the system, reaching stability faster, as can be seen in figure 7. Therefore, $Q_{44} = 0.01$ will be used in further simulations.

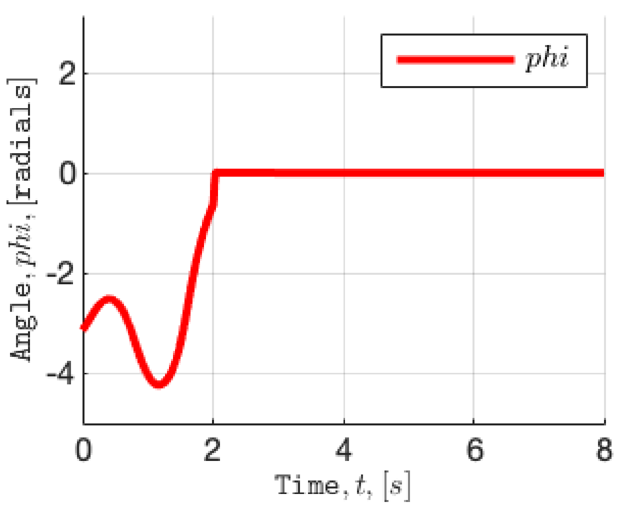

Fig. 7. Output of angle phi, with respect to time, of the controlled CIPS, using the value $\boldsymbol{Q_{44} = 0.01}$

The final Q matrix is composed as follows:

$$Q_{final} = \begin{bmatrix} 0.001 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 10000 & 0 \\ 0 & 0 & 0 & 0.01 \end{bmatrix} \quad (16)$$

The LQR controller is then represented by the K matrix, after implementing the A, B, Q and R values in the $Matlab$ command $K = lqr(A, B, Q, R)$, rendering the following result:

$$K_{final} = \begin{bmatrix} -3.16 & -129 & 10041 & 228 \end{bmatrix} \quad (17)$$

The K values are implemented in the deep learning algorithm.

B. Improved Parameters in deep learning Algorithm

In order for the deep learning algorithm to operate as effective as possible, another non linear system is introduced. The writers of [2] studied the simple pendulum with two state variables, ϕ and $\dot{\phi}$. The region of attraction of this simple pendulum was enlarged by using the deep learning algorithm. This algorithm consists of many parameters. In order to optimize the chances of satisfying the Lyapunov conditions for our CIPS with four variables, we must find the improved values for certain parameters for the simple pendulum. The parameters in the deep learning algorithm of importance are: *Sample size (SS)*, *Hidden Dimension (HD)*, $ball_{lowerbound}$, $ball_{upperbound}$ and the range of results for x. An improved result has the lowest *running time* and the fewest *iterations*. In Table II, an overview of the performed simulations and the corresponding results is given.

No.	SS	HD	ball$_{lb}$	ball$_{ub}$	x$_{range}$	time (s)	iterations
1	500	6	0.5	6	-6,6	42	540
2	1000	6	0.5	6	-6,6	-	-
3	100	6	0.5	6	-6,6	277	1940
4	200	6	0.5	6	-6,6	11	290
5	300	6	0.5	6	-6,6	34	590
6	200	8	0.5	6	-6,6	144	1440
7	200	4	0.5	6	-6,6	-	-
8	200	6	-4	4	-6,6	10	250
9	200	6	-2	2	-6,6	18	390
10	200	6	-3	3	-6,6	9	240
11	200	6	-3	3	-10,10	14	370
12	200	6	-3	3	-1,1	7	230
13	200	6	-3	3	-3,3	9	290
14	200	6	-3	3	-4,4	6	210

TABLE II
SIMPLE PENDULUM: ALGORITHM PARAMETERS

Lowering the *Sample Size* improves the *running time* and lowers the needed *iterations* to satisfy the Lyapunov conditions, for which $SS = 200$ turned out to be an improvement. Enlarging or lowering the amount of *Hidden Dimensions* in the neural network had a negative effect on both *running time* and the needed *iterations*. Therefore, the initial amount of 6 *Hidden Dimensions* is useful for this research. The *ball range* turned out useful for this research with the *upper boundary* being 3 and the *lower boundary* begin -3. Enlarging the x_{range} made the algorithm take longer to satisfy conditions. Lowering the x_{range} improved the *running time* and lowered the needed *iterations*. It turned out that $x_{range} = -4, 4$ rendered the lowest *running time*.

C. Outcome of Implementation of CIPS in Deep Learning Algorithm

Implementing the improved algorithm parameters as mentioned in $No.14$ in *Table II* in the algorithm composed for the CIPS in [13], answers the question if the deep learning algorithm is operable for the nonlinear Cart Inverted Pendulum System.

Due to the fact that the algorithm was unable to find a Lyapunov function that satisfied the conditions, this algorithm can not be validated in its current form. In the *Discussion* section, possible outcomes of this experiment are compared to the actual outcome and recommendations are made for further study.

V. Discussion

This section will provide the theoretical reason of the results given in section IV.

The Improvement of the Q variables and the K array was done through altering the diagonal values of the Q matrix. Changing Q_{11}, corresponding to the deviation of displacement x, from 100 to 0.001 improved the stability of the system. This was to be expected, due to the fact that the stability of the system has no correlation with the location, displacement x, of the system, therefore the Q_{11} value should be low. Having Q_{22} as a very large or small number slows down the stability time, therefore proving that either a very large or small deviation of the velocity v is disadvantageous for the system's stability. The value for Q_{33}, corresponding to the angle ϕ of the pendulum, can neither be too large nor too small, as the simulation results showed. The fact that the deviation of the angle should be as low as possible for improved control, combined with the instability of the system when Q_{33} is too big are the reason. Lowering Q_{44}, representing the relative importance of adjusting the velocity of the angle, $\dot{\phi}$, will lower the movement of the pendulum, resulting in the system being stable quicker.

Lowering the *sample size* results in less samples needed to be evaluated by the algorithm, therefore lowering the time needed to satisfy the *Lyapunov conditions*. Changing the *ball range* to points around central point 0, increases the chance of satisfying *Lyapunov conditions* since both the positive and the negative values are checked on satisfied conditions. Decreasing the x_{range} decreases the amount of possibilities for the state variables, therefore helped the algorithm decrease the time to satisfy the *Lyapunov conditions*.

The reason the deep learning algorithm was not able to compute values to satisfy the *Lyapunov conditions* have nothing to do with the system itself since the CIPS is a Lyapunov stable system. A recommendation for future study would be to adjust the deep learning algorithm in the neural network, in order to be able to compute the *Lyapunov conditions* for nonlinear systems with four state variables. Since the alteration of the neural network part of the algorithm is outside the focus of the study, this has not been included in the scope of this research.

VI. Conclusion

In order to validate the applicability of the deep learning algorithm from [2], simulations have been done, with improved parameters and improved LQR controller values, for a nonlinear system with four state variables. The conclusion can be drawn, that the deep learning algorithm in its current form is not suited for calculating values that satisfy the Lyapunov conditions for the cart inverted pendulum system.

VII. Appendices

A. Simplification of the Deep Learning Algorithm

Developed by the writers of paper [2]. As can be seen, the algorithm consists of a learner and a falsifier. Once the falsifier is unable to produce unstable points for the function, the function is considered Lyapunov stable.

Algorithm 1 Neural Lyapunov Control

```
function LEARNING(X, f, q^lqr)
    Set learning rate (0.01), input dimension (# of state variables), output dimension (1)
    Initialize feedback controller u to LQR solution q^lqr
    Repeat:
        V_θ(x), u(x) ← NN_θ,u(x)                              ▷ Forward pass of neural network
        ∇_{f_u} V_θ(x) ← Σ_{i=1}^{D_in} ∂V/∂x_i [f_u]_i(x)
        Compute Lyapunov risk L(θ, u)
        θ ← θ + α∇_θ L(θ, u)
        u ← u + α∇_u L(θ, u)                                  ▷ Update weights using SGD
    Until convergence
    return V_θ, u
end function
function FALSIFICATION(f, u, V_θ, ε, δ)
    Encode conditions in Definition 5
    Using SMT solver with δ to verify the conditions
    return satisfiability
end function
function MAIN()
    Input: dynamical system (f), parameters of LQR (q^lqr), radius (ε), precision (δ) and an
      initial set of randomly sampled states in D
    while Satisfiable do
        Add counterexamples to X
        V_θ, u ← LEARNING(X, f, q^lqr)
        CE← FALSIFICATION(f, u, V_θ, ε, δ)
    end while
end function
```

Fig. 8. Deep learning algorithm, as is explained in [2].

B. LQR values

Computing the LQR values was done by using $Matlab$. Needed components for this calculation were the A, B and Q matrices and the R-value. These follow the theory explained in [12] and are composed and chosen as follows:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \frac{-1}{M} & \frac{mg}{M} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{-1}{Ml} & \frac{(M+m)g}{Ml} & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -1 & 4.9050 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -1.4286 & 21.0214 & 0 \end{bmatrix} \tag{18}$$

$$B = \begin{bmatrix} 0 \\ \frac{1}{M} \\ 0 \\ \frac{1}{Ml} \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1.4286 \end{bmatrix} \tag{19}$$

$$Q_{initial} = \begin{bmatrix} 100 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1000 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{20}$$

$$R = 0.0001 \tag{21}$$

The command $K = lqr(A, B, Q, R)$ in $Matlab$ renders the K matrix in (14) as a result.

References

[1] Mohammed Davahli, Waldemar Karwowski, Krzysztof Fiok, Thomas Wan and hamid Parsaei. Controlling Safety of Artificial Intelligence-Based Systems in Healthcare. Symmetry 2021, 13, 102. Pages 1-5. January 2021.

[2] Ya-Chien Chang, Nima Roohi and Sicun Gao. Neural Lyapunov Control. 33rd Conference on Neural information Processing Systems (NeurIPS 2019), Vancouver, Canada. December 2019.

[3] Pablo A. Parrilo. Structured semidefinite programs and semi algebraic geometry methods in robustness and optimization. PhD thesis, California Institute of Technology. 2000.

[4] Lal Bahadur Prasad, Barjeev Tyagi and Hari Om Gupta. Optimal Control of Nonlinear Inverted Pendulum System Using PID Controller and LQR: Performance Analysis Without and With Disturbance Input. International Journal of Automation and Computing, December 2014. Pages 661-670.

[5] Alexander Nebylov and Vladimir Nebylov. Reusable Space Planes Challenges and Control Problems. 20th IFAC Symposium on Automatic Control in AerospaceACA, 2016. Pages 480-485.

[6] Karl Astrom and Richard Murray. Analysis and Design of Feedback Systems: An Introduction for Scientists and Engineers. 2003. Pages 41-52.

[7] R. Olfati-Saber. Fixed point controllers and stabilization of the cart-pole and rotating pendulum. In:Proc. of the 38th IEEE Conf. on Decision and Control, Phoenix Az., vol. 2, pp. 1174–1181 (1999)

[8] Spencer Richards, Andreas Krause and Felix Berkenkamp. The Lyapunov Neural Network: Adaptive Stability certification for Safe Learning of Dynamical Systems. 2nd conference on Robot Learning, 2018, Zürich.

[9] Carlos Aguilar-Ibanez and Miguel Suarez-Castanon. Stabilization of the Inverted Pendulum via a Constructive Lyapunov Function. Acta Appl Math 2010. Pages 15-20.

[10] Vinodh Kumar E Jovitha Jerome. Robust LQR Controller Design for Stabilizing and Trajectory Tracking of Inverted Pendulum. Procedia Engineering 64 (2013). Pages 169-178.

[11] Aamir H.O.A., Martino O.A. Matthew W.D. New Approach for Position Control of Induction motor. 45th Universities Power Engineering Conference.

[12] Steven Brunton Nathan Kutz. Data Driven Science and Engineering. *Pages 343-347*

[13] Kristan Dötsch. CIPS-LQR-Solution. https://github.com/Kristandotsch/CIPS-LQR-solution/tree/main, *2021*.