



---

# ANALYSIS OF CONTROLLED MECHANISM WITH SIGNIFICANT NONLINEARITIES

V. Rulc\*, H. Purš†, J. Kovanda‡

---

**Abstract:** Solving inverted pendulum by co-simulation between multi-body solver MotionSolve and signal processing control in solidThinking Activate. The simulation of inverted pendulum uses an innovative model of friction which is physically and mathematically more accurate than usual CAE friction models. This model of friction adds nonlinearity to the system. Two types of controlling mechanism for active balancing of inverted pendulum are used: PID and ANN controller. A non-traditional false angular deviation approach for returning a cart to its initial position was used.

Key words: *inverted pendulum, non-linearity, controller, co-simulation, friction model*

Received: July 13, 2017

DOI: 10.14311/NNW.2017.27.019

Revised and accepted: August 10, 2017

## 1. Introduction

An inverted pendulum is such a pendulum whose center of gravity lies above the axis of rotation. A control of an inverted pendulum is a classic task of controlling an unstable system. This phenomena we encounter in day to day life. An inverted pendulum is also a human body in a vertical position. It is noticed not only in an elementary example as a rapid changes of speed and direction of a city bus, but also in banal activities such as walking or standing. Understanding and controlling the inverted pendulum is necessary for development of robotic humanoid [7]. In the field of transportation [10], the principle of the inverted pendulum is also used in Segway vehicles.

Example from the other sides of human knowledge: a typical inverted pendulum is a rocket. A propulsion system at the bottom (or anywhere else, because of “The Pendulum Rocket Fallacy” [2]) must control the position of a center of mass. From industrial perspective it can be seen in an active driven crane in seafaring.

---

\*Vojtěch Rulc – Corresponding author; Czech Technical University in Prague, Faculty of Transportation Sciences, Konviktská 20, 110 00 Praha 1, Czech Republic, E-mail: [rulcvojt@fd.cvut.cz](mailto:rulcvojt@fd.cvut.cz)

†Hynek Purš; Advanced Engineering s.r.o., Na Ostrohu 16, 160 00 Praha 6, Czech Republic, E-mail: [hpurs@advanced-eng.cz](mailto:hpurs@advanced-eng.cz)

‡Jan Kovanda; University of West Bohemia, Faculty of Mechanical Engineering – Regional Technological Institute, Univerzitní 8, 306 14 Plzeň, Czech Republic, E-mail: [kovanda@rti.zcu.cz](mailto:kovanda@rti.zcu.cz)

In this article, the inverted pendulum has been chosen because of its simplicity. Its dynamic system is easy to describe. Verifying the accuracy of the method is not affected by error incurred by the complexity of the system. After verifying the correct operation of the controllers on a simple dynamic system, the authors can use the control mechanisms on more complex systems in the future. An example of such system can be an active wheel suspension of vehicle, which is mentioned in [17]. The verified methods can also be used in follow-up work with [16] and extend the possible control methods.

One of the ways to deal with the complexity of mechatronic systems developed is so-called Model-Based Development (MBD). Model-Based Development is a development method that establishes a common framework for communication during the design of a dynamic system. The development is based on the following 4 steps:

- modeling of product and environment,
- analysis and assembly of the controller,
- simulation of the product and controller's operation,
- combining all of previous steps by controller deployment.

This can be advantageously used not only when control system development, but also during a design of human-machine interface as a [14] is.

The purpose of this scientific paper is to verify that a co-simulation of multi-body solver and signal processing control can handle the first three steps of MBD for dynamic mechanisms. And moreover it should compare results of both of used controllers.

## 2. Model description

The inverted pendulum is a classical problem solved countless times. From the recent time we can mention for example [6, 11, 13] or [18]. Our model consists of four bodies: frame, cart, Hooke's cross and pendulum. The kinematic constraints are flat (frame-cart) and rotational (cart-Hooke's cross, Hooke's cross-pendulum). The number of degree of freedom is:

$$i = 6 \cdot (n - 1) - \sum j \cdot d_j,$$

where  $n$  is a number of bodies, and  $j$  is a number of kinematic joints of  $d_j$  category (= number of reduced degrees of freedom – DOF).

Number of bodies:

$$n = 4.$$

Number of kinematic joints of given category:

- Flat joint:  $j = 1$ ,  $d_j = 4$  reduces the relative mobility by 4, it has 2 DOF controlled by the control system.
- Rotational joint:  $j = 2$ ,  $d_j = 5$  reduces the relative mobility by 5, it has 1 DOF and these joints are modelled with friction.

Finally

$$i = 6 \cdot (4 - 1) - (1 \cdot 4 + 2 \cdot 5) = 4 \text{ DOF.}$$

Therefore the system has 4 DOF, the flat joint is active, Hooke's joint kinematic quantities are control inputs.

This three-dimensional task is solved by splitting into two subtasks. Pendulum in  $xz$  plane and pendulum in  $yz$  plane. A scheme of the planar case is shown in Fig. 1. (Authors' note: When angular stability is mentioned, it means the stabilisation in  $xz$  plane. The case in  $yz$  plane is completely the analogous and will not be described.)

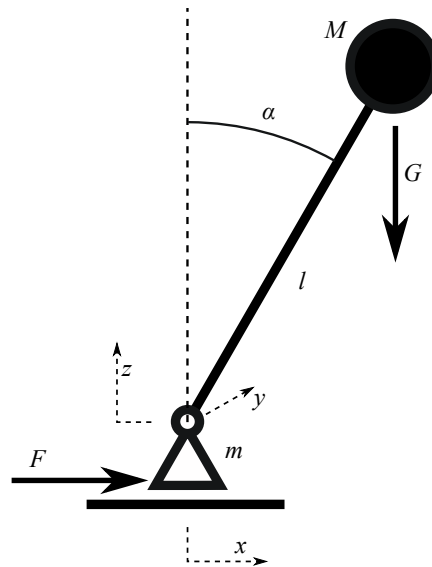


Fig. 1 Schema of the model pendulum.

The pendulum of mass  $M$  with the idealized center of gravity position in the center of a ball is via an ideally fixed intangible rod of length  $l$  attached by a Hooke's joint to a cart of mass  $m$ .

Stabilization of the system is carried out by the action of the external force  $F$  on the cart. There is no friction in the flat joint between the cart and ground, but there is an innovative model of friction in the Hooke's joint between the pendulum and the cart.

The mathematical model of the pendulum does not need to identify and derive motion equations here because the used control methods work without a mathematical description of the controlled system. So they are more easily adaptable to different changes and can be more easily coped with the fact that the system parameters can not describe the system perfectly. In other words, they are more robust.

Parameters of the system were chosen as follows in Tab. I.

Quantity	Value
$M$	0.20 kg
$m$	0.45 kg
$l$	300 mm
$\alpha$	10°

**Tab. I** *Parameters of the model pendulum.*

### 3. Friction model

The aim is to utilise the adequate friction model. The available models are based on a principle of a massive increase in the coefficient of static friction. It is usually defined to such a size which burdens a simulation by computational load and numerical errors. They occur naturally due to the fact that computers work only with limited precision numbers. If the number exceeds the specified precision limit, it is either cut or rounded and, with an increasing number of subsequent arithmetic operations, the original minor rounding error increases significantly. A physically and mathematically correct solution should be a complete removal of the relevant DOF. Therefore, the static friction in our model is replaced by a change in the structure of the motion mechanism. In case of static state, the Hooke's cross is replaced by a fixed link. The friction model used in the system causes a relatively strong non-linearity. This is a major challenge for the control system.

## 4. Simulation tools

### 4.1 SolidThinking Activate

SolidThinking Activate is used to build block diagrams and for graphical programming. This approach has the advantage of simplicity and intuitiveness. Thanks to this, programming is also available to a person who has not yet been in contact with it, and it is not necessary to know any programming language. However, the possible knowledge of the programming language can be advantageously used to create custom block diagrams. Activate supports, in addition to HyperWorks' Open Matrix Language, the traditional C language. It will be also used in our case. Another useful feature is a presence of a block that allows you to link the flowchart to a multibody model created in MotionView and co-simulate along with MotionSolve. This approach is useful because it allows you to quickly and efficiently change the system described without the need to rewrite equations which are often very complicated.

### 4.2 Altair MotionView and MotionSolve

Altair MotionView is a pre- and post-processing multibody systems modeling environment. It enables to build and analyze mechanical system design. In many cases (such as vehicle dynamics for example) it is possible to use mechanical templates and prepare a model just by changing parameters. In connection with MotionSolve,

MotionView mediates a simulation of multibody kinematic, (quasi)static, dynamic or linear analysis. Naturally it supports flex body modeling too if needed [20].

“MotionSolve is a scalable and integrated multibody solution that enables to graphically build systems, numerically solve the underlying equations, evaluate system behavior by examining plots and animations and improve their performance through design exploration and optimization.” [19]

## 5. Simulation method

To simulate our model and control, we chose a combination of SolidThinking Activate, Altair MotionView and Altair MotionSolve software. In MotionView, the physical model of the inverted pendulum was modeled. The control system and external interventions that periodically set the pendulum into a new unstable position within a specified interval were modeled in Activate. The simulation then proceeded as a co-simulation between Activate and MotionSolve.

Co-simulations are a current trend in the industry. It helps to combine easily the models of each sub-system part and simulate each one by an appropriate solver. The method saves time and reduces errors in the development phase.

### 5.1 Control systems in Activate

Two control systems are programmed and generally used: a PID controller and an artificial neural network controller. A simplified flowchart in Fig. 2 shows the method of simulation in Activate. It is scheme of feedback system, where controlled system is co-simulated out of the Activate environment. There is also a generator of random destabilising forces generating short force impulses in specified periods. The force is applied to center of mass of the pendulum and causes deviation of pendulum angle. It demonstrates that controllers are able to control a pendulum continuously not only the initial deviation.

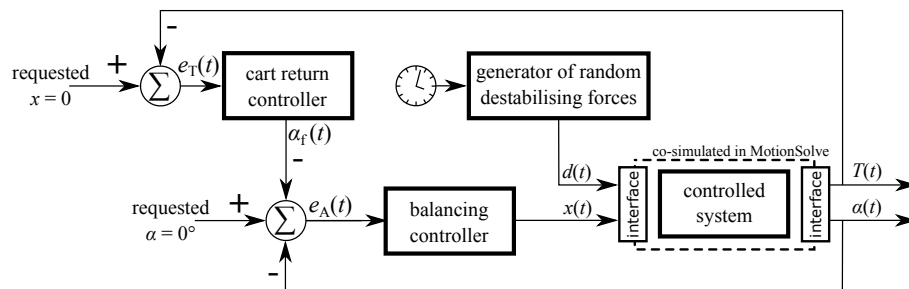


Fig. 2 Schema of the control system.

Returning the cart to its initial position is solved in non-traditional way. Usually forces from a balancing controller and a cart return controller are summed and the resulting force is applied to a cart. In case of this article, the only one control force is used. Returning of the cart is ensured by creating false angular deviation of pendulum. This approach is shown in Fig. 2. A longitudinal deviation of the cart

$e_t(t)$  is input into the cart return PID controller. Its output is not control force, but a false angle  $\alpha_f(t)$ , which is add to angular deviation  $\alpha(t)$ . By this operation a false deviation  $e_A(t)$  is created and put forward to balancing controller. The balancing controller then act as if the false deviation is real and pushes the cart more or less than just when balancing. This results in movement of whole pendulum to relevant side and return to requested coordinations.

### 5.2 PID controller

The proportional controller has the disadvantage that it usually causes a static or constant static error [1]. So the first chosen selected control method is a PID controller. It is a continuous controller that consists of three main parts: proportional, integration and derivation. The control deviation  $e(t)$  enters into

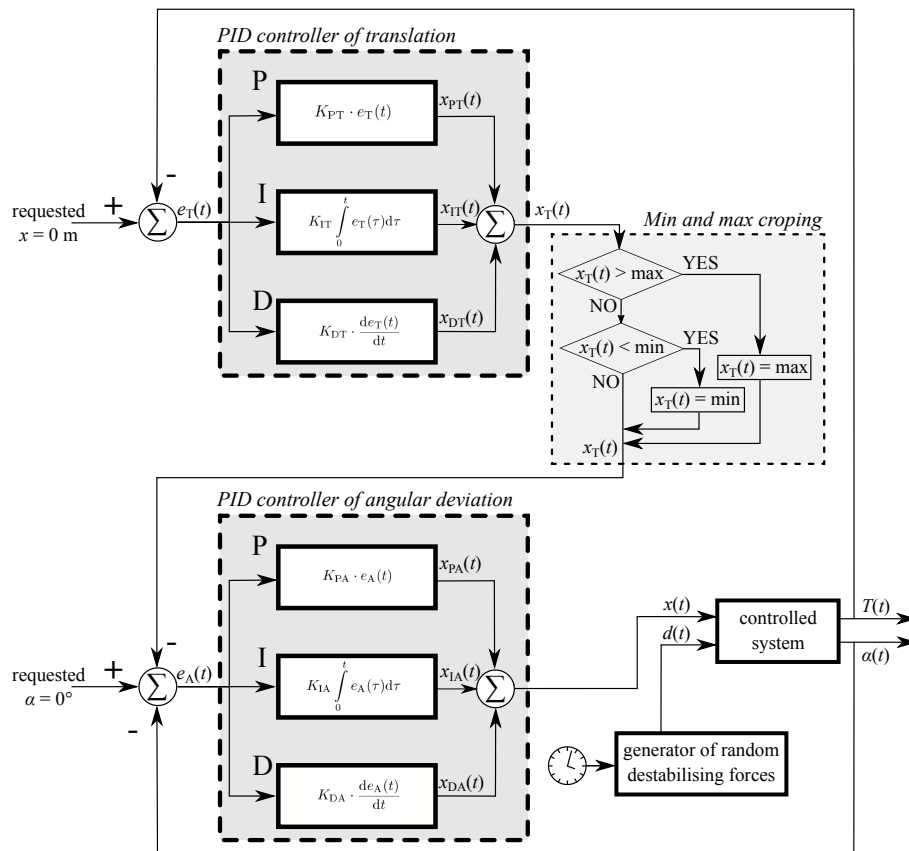


Fig. 3 Schema of the PID control system.

all of them and, when outputting them, the partial action quantities  $x_i(t)$  are summed up and the manipulated variable  $x(t)$  is generated. In the proportional part, the manipulated variable  $x_P(t)$  is directly proportional to the control deviation  $e(t)$ . In the integration part  $x_I(t)$  is directly proportional to the deviation integrity

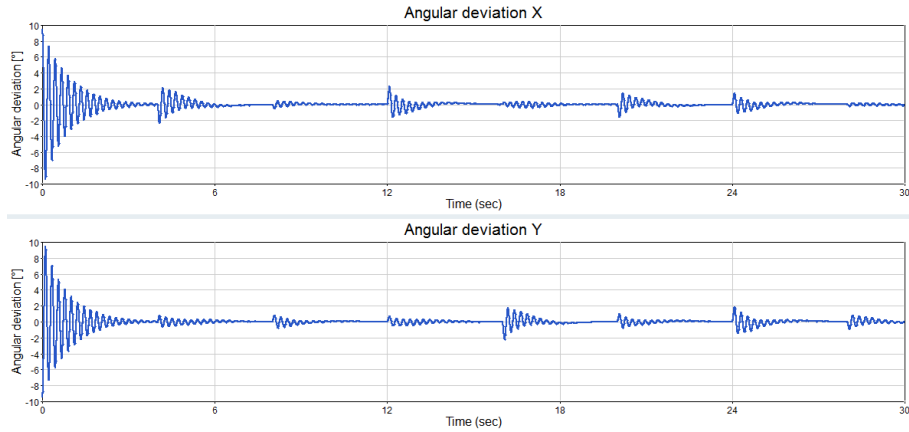


Fig. 4 Angular deviation of pendulum with PID controller.

$\int_0^t e(\tau) d\tau$ . And analogously in the derivation part,  $x_D(t)$  is directly proportional to the derivation of the control deviation  $de(t)/dt$ .

The intended simple false angular deviation approach for returning the cart had to be edited a little. At the moment, when longitudinal deviation was too large, manipulated variable was so large, that it disturbed a balancing of pendulum. The most effective solution of this problem is to insert cropping block just after the cart returning PID controller. This block cuts off all the values higher or lower than thresholds set. The threshold setting must respect that too high values causes instability and too low values have no power to influence the final manipulated variable and the cart will not return to initial position.

Both of the PID controllers were set with manual iterative tuning method. Parameters of the PID coefficients and thresholds are described in Tab. II.

Parameter	Value
$K_{PA}$	0.2
$K_{IA}$	0.8
$K_{DA}$	0.8
$K_{PT}$	0.012
$K_{IT}$	0.003
$K_{DT}$	0.010
threshold <sub>min</sub>	-60
threshold <sub>max</sub>	60

Tab. II Parameters of the PID controller system.

The PID controllers are able to quieten a pendulum almost completely and push the cart with pendulum to initial coordinates.

### 5.3 Artificial neural network

Artificial neural network is mathematical approach inspired by elemental neurophysiology [5]. At the beginning of work on ANN was a recognition that “human brain computes in entirely different way from the conventional digital computer. The brain is a highly complex, nonlinear and parallel computer” [8] and in many type of tasks it achieves much better results. So ANN tries to emulate real neural network which in much larger scale forms our brains. The similarity with brain enables to ANN adapt itself to its surrounding system (or system changes) and to be fault tolerant [8].

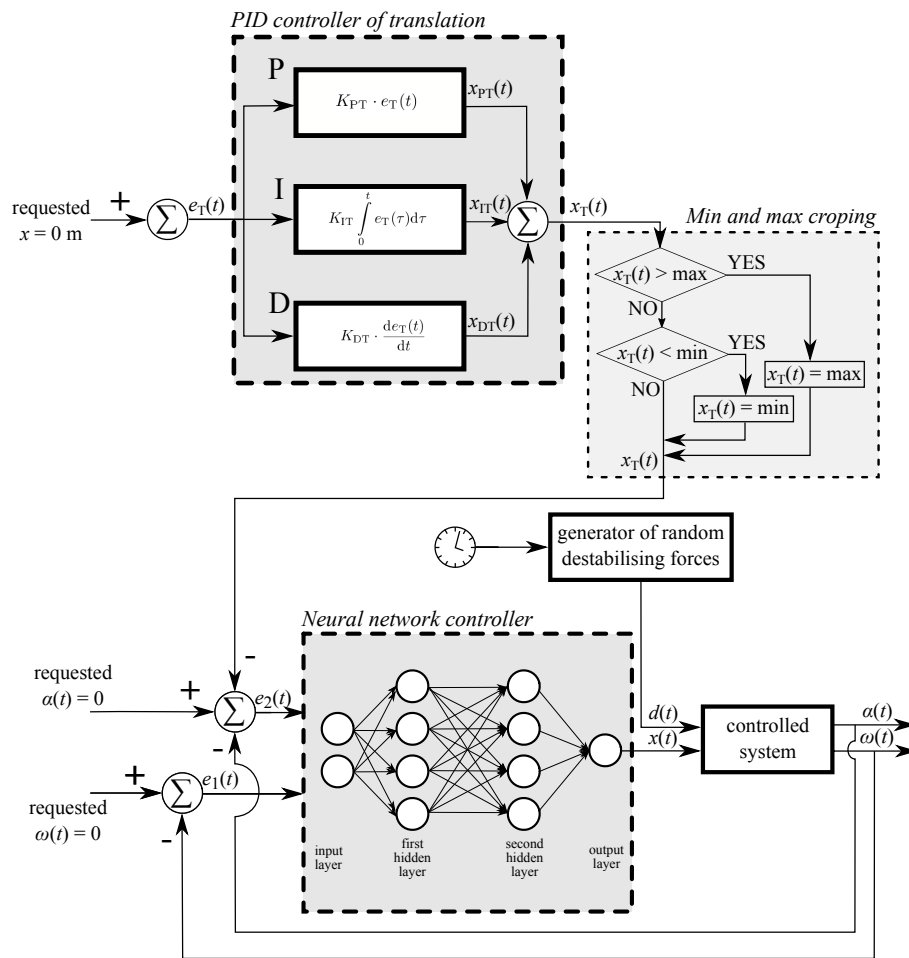


Fig. 5 Schema of the ANN control system.

To insert an ANN controller, the opensource library FANN [12] have been implemented through CCustomBlock. The library was chosen because it is programmed in C language (Activate supports neither C++ nor C#) and it provides tools for



complex usage of ANN. It offers several training algorithms and a lot of activation functions.

The following steps were performed to prepare an ideal ANN:

- (1) Execute the model with PID controller and export training data.
- (2) Create basic ANN.
- (3) Test training methods on basic ANN, choose the best one and use it in next steps.
- (4) Test combinations of activation functions in hidden and output layers, choose the best combination and use it in next steps.
- (5) Test different sizes of ANN, choose the best one and use it in following steps.
- (6) Integrate defined structure of ANN into model. Execute simulation with PID controller and let the ANN train online parallel to PID.
- (7) Use the trained ANN as a controller.

The process was realized as follows:

- (1) The model with PID controller was edited at the earliest. The module for returning the cart to its default position was turned off. Then the simulation was executed and a text file with signals was exported. It contained scaled angular deviation of pendulum (divided by 100) and angular velocity of pendulum (divided by 1000) as input values and actuator force (divided by 100) as an output. These values were captured for every simulation step. 10 seconds of simulation exported 1 406 samples.
- (2) A four layer network was chosen as a basic ANN. This network had two inputs, 5 neurons in each hidden layer and one output. Activation function in all of neurons was linear with steepness  $s = 0.5$ .
- (3) Five training methods were tested and a MSE (Mean Squared Error) and Bit Fail value after 2 000 training epochs was recorded. A Bit Fail value is a count of output values their absolute value error from an estimated output value is higher than a threshold. We have chosen threshold as 0.0001. The following training methods were tested:
  - **Incremental method** – Standard backpropagation algorithm, where the weights are updated after each training pattern.
  - **Batch method** – Standard backpropagation algorithm, where the weights are updated after calculating the mean square error for the whole training set.
  - **RPROP** – Resilient backpropagation method is a more advanced batch training algorithm described by [9].
  - **QPROP** – Quick resilient backpropagation method is another more advanced batch training algorithm. This described by [4].

- **SARPROP** – Simulated Annealing resilient backpropagation method is a simulated annealing enhancement to resilient back propagation described by [15].

It is evident from Tab. III that the Incremental method is the most efficient one for our problem. This method was selected for following steps.

Train method	After 2000 epochs	
	MSE	Bit fail
Incremental	0.000002	0
Batch	0.011393	0
RPROP	0.000036	0
QRPROP	0.011393	0
SARPROP	0.000036	0

**Tab. III** Results of tested training methods.

- (4) A test of the most convenient activation functions was performed. Following functions were tested:

- Linear
- Sigmoid
- Sigmoid stepwise
- Sigmoid symmetric
- Sigmoid symmetric stepwise
- Gaussian
- Gaussian symmetric
- Elliot
- Elliot symmetric
- Linear piece
- Linear piece symmetric
- Sin symmetric
- Cos symmetric

All of these functions were combined in hidden and output layers and their results after 3000 epochs were recorded. The best ten combinations are assembled in Tab. IV. Based on the results, the combination of *symmetric Elliot* function for hidden layers and *symmetric stepwise sigmoid* for output layer was chosen. Parameters of the chosen functions are following:

- Symmetric Elliot function:
  - fast symmetric sigmoid-like activation function defined by David Elliot [3],
  - span:  $-1 < y < 1$ ,
  - $y = (x \cdot s) / (1 + |x \cdot s|)$ ,
  - $d = s \cdot 1 / ((1 + |x \cdot s|) \cdot (1 + |x \cdot s|))$ .
- Sigmoid symmetric stepwise function:
  - linear approximation to symmetric sigmoid,
  - span:  $-1 < y < 1$ ,

Activation function		After 3000 epochs	
Hidden layers	Output layer	MSE	Bit fail
Elliot symmetric	Sigmoid symmetric stepwise	9.705e-08	0
Elliot symmetric	Sigmoid symmetric	9.949e-08	0
Elliot symmetric	Cos symmetric	9.951e-08	0
Elliot	Cos symmetric	9.953e-08	0
Elliot symmetric	Sin symmetric	9.986e-08	0
Elliot symmetric	Linear	9.994e-08	0
Elliot	Sigmoid symmetric	9.998e-08	0
Elliot symmetric	Linear piece symmetric	1.071e-07	0
Elliot	Sigmoid symmetric stepwise	1.083e-07	0
Sigmoid symmetric	Sigmoid symmetric	1.553e-07	0

**Tab. IV** Results of the best 10 combinations of activation functions.

$$- y = \tanh(s \cdot x) = 2 / (1 + \exp(-2 \cdot s \cdot x)) - 1,$$

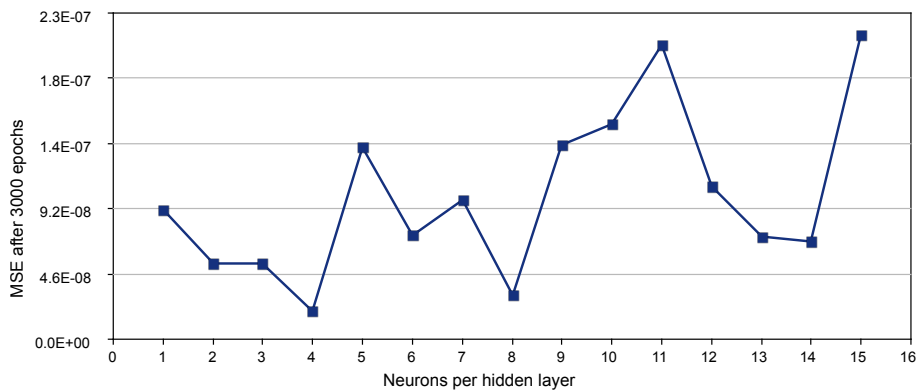
$$- d = s \cdot (1 - (y \cdot y)),$$

where:

- $x$  is the input to the activation function,
- $y$  is the output,
- $s$  is the steepness and
- $d$  is the derivation.

Steepness of both of activation functions was chosen  $s = 0.5$ .

- (5) The previously chosen parameters were applied on basic network and test of ideal size of network was started. Networks from 1 neuron per each hidden layer up to 15 neurons per each hidden layer were tested. Results are described in Tab. V and in Fig. 6. For next steps an ANN with 4 neurons per hidden layer was chosen.



**Fig. 6** Results of testing various size of ANN.

Neurons per hidden layer	MSE after 3000 epochs
1	$9.146 \times 10^{-8}$
2	$5.312 \times 10^{-8}$
3	$2.767 \times 10^{-8}$
4	$1.990 \times 10^{-8}$
5	$1.353 \times 10^{-7}$
6	$7.280 \times 10^{-8}$
7	$9.806 \times 10^{-8}$
8	$3.144 \times 10^{-8}$
9	$1.372 \times 10^{-7}$
10	$1.518 \times 10^{-7}$
11	$2.077 \times 10^{-7}$
12	$1.073 \times 10^{-7}$
13	$7.273 \times 10^{-8}$
14	$6.910 \times 10^{-8}$
15	$2.141 \times 10^{-7}$

**Tab. V** Results of testing various size of ANN.

- (6) In the Activate model, an ANN with properties which were selected depending on test results in previous steps were created. The ANN watched scaled input and output parameters online from the Multi-body model and from the PID regulator. The training process ran real-time in the simulation. The training in this step is not comparable with training from previous steps. In previous steps the training was based on limited count of samples which were served to ANN again and again. The MSE in this repeating process can be downsized to the served data faster, but it does not say about quality of network so much. The MSE in real-time training with constantly new data is more accurate. The training had been under way until the MSE reach our required value  $MSE = 1 \times 10^{-6}$ . This had taken 483 seconds of simulation time. The prolonged training process (until  $MSE = 1 \times 10^{-7}$ ) can be seen in Fig. 7.
- (7) Outputs from Multi-Body model were accordingly scaled for input to ANN controller. Outputs from ANN were scaled too, and the ANN replaced the PID. The balancing of pendulum was successful. Its results are described in following chapter.

## 6. Results comparison

To compare the results of the PID and ANN controller, an identical signal of destabilizing impulses was used. In previous cases it was usually randomly generated, but this time, during one simulation, destabilizing impulses were recorded, exported to a file, and imported into further simulations. A signal of the used destabilizing pulses is shown in Fig. 8. As can be seen from the Fig. 9, stabilization of the pendulum by

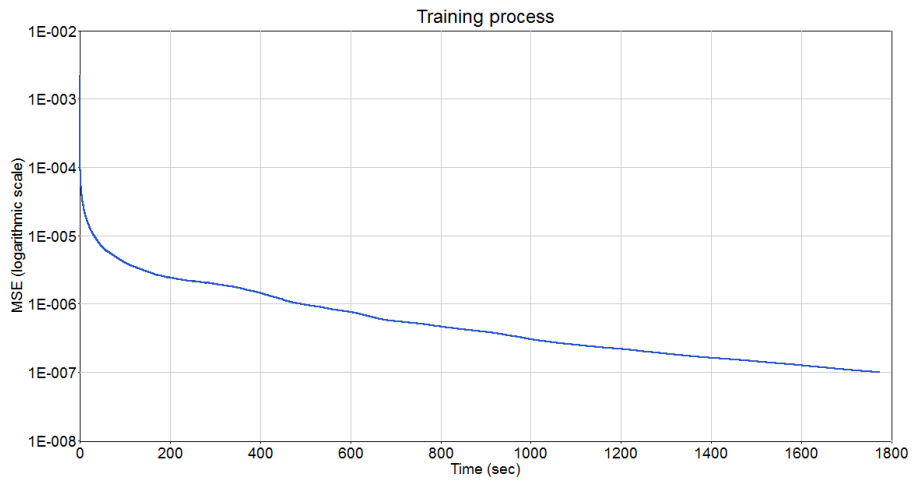


Fig. 7 Process of training ANN.

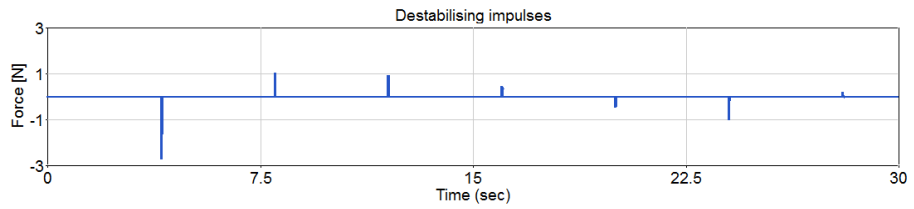


Fig. 8 Shape of the destabilizing impulses.

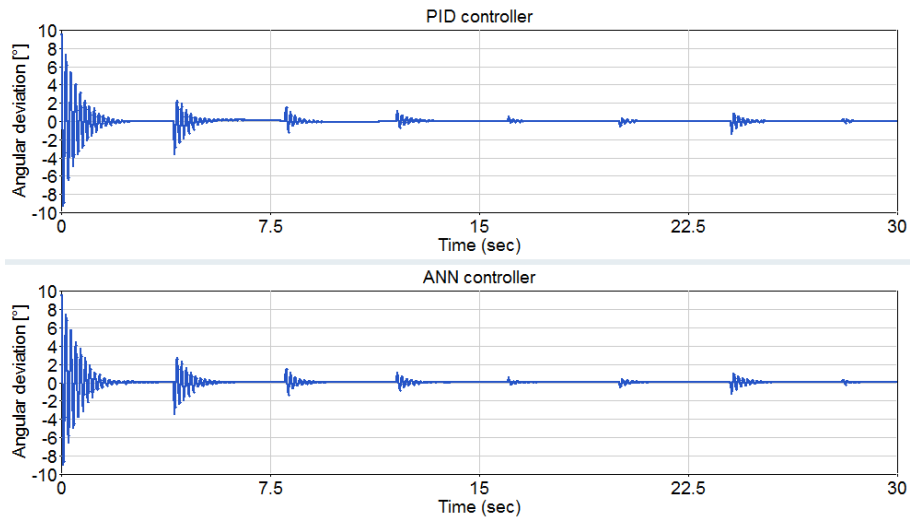


Fig. 9 Comparison of PID and ANN controller – oscillation in XZ plane.

the both of controllers is almost identical with only very little differences. These are noticeable in Fig. 10, that shows integral of absolute value of angular deviations.

Fig. 11 relates to an energy consumption of balancing the pendulum. The curves in the pictures represent value of integral of absolute value of control force at specific time. That corresponds with energy consumption. Final integration values ( $25.0 \cdot F \cdot s$  for the PID controller and  $16.4 \cdot F \cdot s$  for the ANN controller) imply

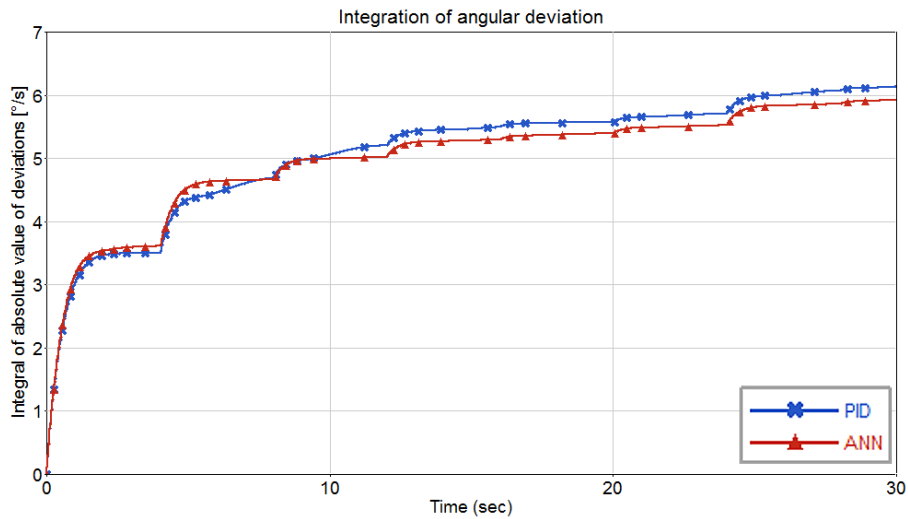


Fig. 10 Comparison of PID and ANN controller – integral of absolute values of deviations.

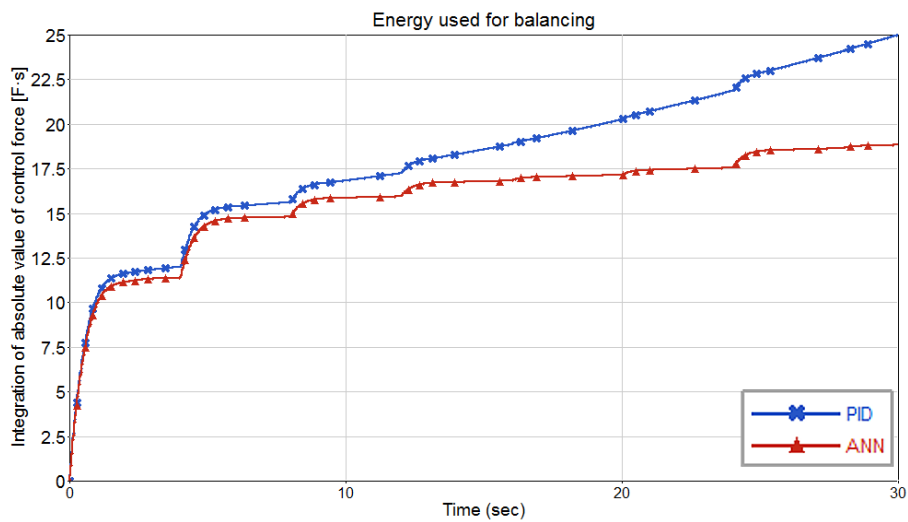


Fig. 11 Comparison of PID and ANN controller – integral of absolute values of control forces.

that PID controller was approximately 50 % more energy-intensive than the ANN controller in our case.

The Fig. 12 shows longitudinal deviation of cart. It is obvious, that the false angular deviation approach is suitable for both of controllers and the results are almost same – similarly as the angular deviation results.

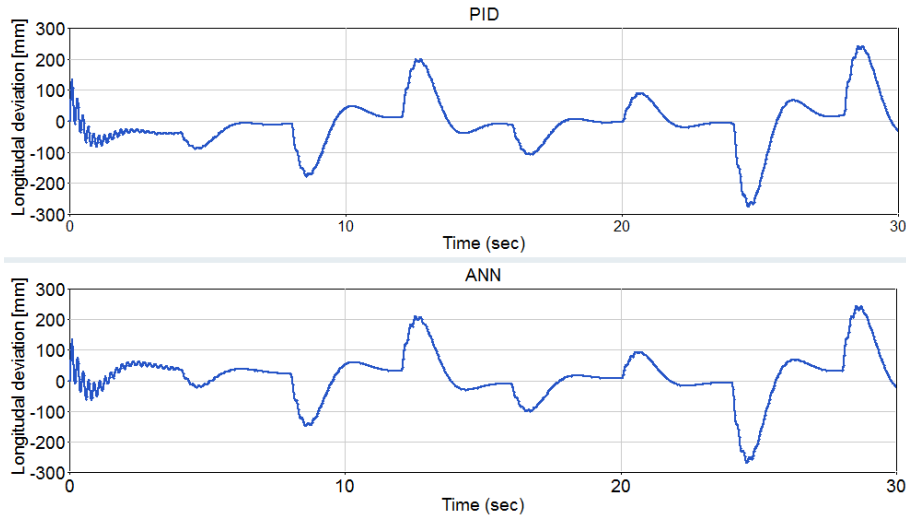


Fig. 12 Comparison of PID and ANN controller – longitudinal deviation of cart.

## 7. Conclusions

By trying different parameters of ANN step by step, an ideal ANN was found. This ANN was trained real-time side by PID controller and the resulting sufficiently accurate ANN was used as a substitution of the balancing PID controller.

Abilities of these two controllers were then compared and no significant differences were observed except energy consumption. The PID controller was approximately 50 % more energy-intensive than the ANN controller in our case.

Balancing the pendulum is a nonlinear problem by nature, but other nonlinearity has been added by the friction model. Despite this, however, both of the controllers were successful and made their task effectively. It is not possible to clearly determine which controller is better. It depends on the area of application and defined requirements..

The research confirmed that the false angular deviation approach for returning cart to its initial position is applicable for both of used balancing controllers. Both controllers also have a very similar results in returning the cart.

The combination of Activate plus external opensource C-language library can be advantageously used. The Internet is full of many free libraries of various disciplines. It is not necessary to develop new and new scripts for every problem. Also, there is no need to program in not user-friendly text code, but use of a user-friendly option to build flowchart. Also the co-simulation between signal processing software and

multi-body solver was beneficial. It highly saves time and errors during a controlled model describing.

## Acknowledgement

This article was supported by the project LO1502 ‘Development of the Regional Technological Institute’ under the auspices of the National Sustainability Programme I of the Ministry of Education of the Czech Republic aimed to support research, experimental development and innovation, and by the project TE01020155 ‘Transport systems development centre’ under the auspices of the Technology Agency of the Czech Republic.

## References

- [1] ÅSTRÖM K.J., HÄGGLUND T. *PID controllers*. 2nd ed. Research Triangle Park: Instrument Society of America, 1995. ISBN: 1556175167.
- [2] BOWERY J. *The Pendulum Rocket Fallacy* [online]. 2001 [viewed 2017-05-15]. Available from: [https://web.archive.org/web/20091018232320/http://geocities.com/jim\\_bowery/pendrock.html](https://web.archive.org/web/20091018232320/http://geocities.com/jim_bowery/pendrock.html)
- [3] ELLIOT D.L. *A Better Activation Function for Artificial Neural Networks*. ISR Technical Report. 1993.
- [4] FAHLMAN S.E. *Faster-Learning Variations on Back-Propagation: An Empirical Study*. Proceedings of the 1988 Connectionist Models Summer School. San Francisco, CA: Morgan Kaufmann, 1989, pp. 38–51.
- [5] FREEMAN James A. *Neural Networks: Algorithms, Applications, and Programming Techniques*. Reading: Addison-Wesley, 1991. ISBN: 0201513765.
- [6] GRASSER F., D’ARRIGO A., COLOMBI S., RUFER A.C. *JOE: a mobile, inverted pendulum*. IEEE Transactions on Industrial Electronics. 2002, 49(1), pp. 107–114, doi: [10.1109/41.982254](https://doi.org/10.1109/41.982254)
- [7] GRUBER S., SCHIEHLEN W. *Inverse Dynamics Power Saving Control of Walking Machines*. In: Ro.Man.Sy. 14 – Theory and Practice of Robots and Manipulators, Proceedings of the Fourteenth CISM-IFTToMM Symposium, Udine, Italy. Udine: 2002.
- [8] HAYKIN Simon S. *Neural networks and learning machines*. 3rd ed. New York: Pearson, 2009. ISBN: 9780131471399.
- [9] IGEL CH., HÜSKEN M. *Improving the Rprop Learning Algorithm*. Proceedings of the Second International Symposium on Neural Computation. ICSC Academic Press, 2000, pp. 115–121.
- [10] KORTÜM W., LUGNER P. *Systemdynamik und Regelung von Fahrzeugen*. Springer, 1994, doi: [10.1007/978-3-642-47623-5](https://doi.org/10.1007/978-3-642-47623-5)
- [11] MANDIC P., LAZAREVIC M., STOJANOVIC S., RISTANOVIC M. *Real time control of rotary inverted pendulum*. Annals of the Faculty of Engineering Hunedoara. 2014, 12(2), pp. 211.
- [12] NISSEN S. FANN 2.2.0 [software]. [accessed 2015-03-22]. Available from: <http://leenissen.dk/fann/wp/>
- [13] OLIVARES M., ALBERTOS P. *Linear control of the flywheel inverted pendulum*. ISA transactions. 2015, 53(5), pp. 1396–403, doi: [10.1016/j.isatra.2013.12.030](https://doi.org/10.1016/j.isatra.2013.12.030)
- [14] ROZHDESTVENSKIY D., BOUCHNER P., MASHKO A., ABISHEV K., MUKANOV R. *Dynamic Human-Machine Interface for Electrical Vehicle design guidelines*. Smart Cities Symposium Prague (SCSP). 2015. ISBN: 9781467367271
- [15] TREADGOLD N.K., GEDEON T.D. *The SARPROP Algorithm: A Simulated Annealing Enhancement to Resilient Back Propagation*. Proceedings International Panel Conference on Soft and Intelligent Computing. Budapest, 1996, pp. 293–298.



- [16] VALÁŠEK M., STEJSKAL M., SIKÁ Z., VACULÍN O., KOVANDA J. *Dynamic model of truck for suspension control*. Vehicle System Dynamics. 1998, 29, pp. 496–505, doi: [10.1080/00423119808969582](https://doi.org/10.1080/00423119808969582)
- [17] VRÁNA T., BRADÁČ J., KOVANDA J., PURŠ H., RULC V. *The Effect of Arm Stiffness on the Elasto-Kinematic Properties of Single-Axle Suspension by Using the MBS Simulation Model*. The International Journal Of Engineering And Science. 2017, 6(2), pp. 52-61, doi: [10.9790/1813-0602015261](https://doi.org/10.9790/1813-0602015261)
- [18] WANG J. *Simulation studies of inverted pendulum based on PID controllers*. Simulation Modelling Practice and Theory. 2011, 19(1), pp. 440–9, doi: [10.1016/j.simpat.2010.08.003](https://doi.org/10.1016/j.simpat.2010.08.003)
- [19] Altair Engineering, Inc. *Altair MotionSolve* [online]. [viewed 2017-05-21]. Available from: <http://web.archive.org/web/20170518223327/http://altairhyperworks.com/product/MotionSolve>
- [20] Altair Engineering, Inc. *Altair MotionView* [online]. [viewed 2017-05-21]. Available from: <http://web.archive.org/web/20170103064012/http://www.altairhyperworks.com/product/motionsolve/motionview>