Position and velocity predictions of the piston in a wet clutch system during engagement by using a neural network modeling

Yu Zhong[1], Bart Wyns[1], Abhishek Dutta[1], Clara-Mihaela Ionescu[1], Gregory Pinte[2], Wim Symens[2], Julian Stoev[2] and Robin De Keyser[1]

[1]. Department of Electrical energy, Systems and Automation (EeSA), Ghent University, 913 Technologiepark, Zwijnaarde, 9052, Belgium

{Yu.Zhong, Bart.Wyns, Dutta.Abhishek, Claramihaela.Ionescu, Robain.DeKeyser}@ugent.be

[2]. Flanders' Mechatronics Technology Center (FMTC), Celestijnenlaan 300D, Heverlee, 3001, Belgium

{Gregory.Pinte, Wim.Symens, Julian.Stoev}@fmtc.be

Abstract. In a wet clutch system, a piston is used to compress the friction disks to close the clutch. The position and the velocity of the piston are the key effectors for achieving a good engagement performance. In a real setup, it is impossible to measure these variables. In this paper, we use transmission torque and slip to approximate the piston velocity and position information. By using this information, a process neural network is trained. This neural predictor shows good forecasting results on the piston position and velocity. It is helpful in designing a pressure profile which can result in a smooth and fast engagement in the future. This neural predictor can also be used in other model-based control techniques.

Keywords: Neural predictor, wet clutch system, nonlinear system, process neuron model

1 Introduction

A clutch is a mechanical device which provides for the transmission power from input device to output device. Clutches are used whenever the transmission of power or motion needs to be controlled either in amount or over time. Normally inside of a clutch system, there are hydraulic, electrical and mechanical components. It is very difficult to identify and model the wet clutch system since this is a typical nonlinear system [1].

Recently, some advanced learning control methods are used to control and optimize the engagement performance of the wet clutch system. Zhong, *et al.* [2], used a genetic algorithm (GA) to optimize the parametric signal to improve the engagement performance. Depraetere, *et al.* [3, 4], introduced a two-level iterative learning controller (ILC) to generate and track an optimal pressure profile, so that a good engagement performance can be achieved. Dutta, *et al.* [5], proposed to solve this problem by using robust model predictive control (MPC) technique. Gagliolo, *et al.* [6], validated the potential of implementing reinforcement learning techniques on the wet clutch system. This paper focuses on the use of neural network to solve this complex problem.

An artificial neural network has many useful properties and capabilities such as nonlinearity; input-output mapping; adaptivity; and so on [7]. With such advantages, the applications of the neural network can be found in every engineering field such as vehicle system [8], target recognition [9], and so on. In this paper, we will use the neural network modeling algorithm to get the prediction of the position and velocity of the piston in a wet clutch system during its engagement process. The difficulties in training this neural network are:

1. we cannot measure the position and the velocity of the piston in the real test, so we need to find out the information based on the available measurements;
2. the inputs and the outputs of the neural network are time-varying.

To solve these problems, we will first model the wet clutch based on fundamental physical laws and trying to find out the relations between the position/velocity of the piston and the available measurements. Then a process neural network will be trained.

2 Principle of a wet clutch and theoretical modeling

2.1 Wet clutch system

A wet clutch is a clutch that is immersed in a cooling lubricating fluid which keeps the surfaces clean and gives smoother performance and longer life [10]. However, the wet clutch could lose some energy to the liquid since the surfaces can be slippery. One way to overcome this drawback is to stack multiple clutch disks. Fig. 1 shows the design of a wet clutch. An electro-hydraulic pressure-regulated proportional valve regulates the pressure inside the clutch, such that the position of a piston which presses the multiple clutch disks together can be controlled, thus the torque is transmitted. Once the command to engage the clutch is received, the left chamber of the clutch is filled with oil, so that the piston is pushed forward. A good engagement is then defined as decreasing the distance between the piston and the disks as fast as possible to zero, without the piston and the disks making brutal contact.

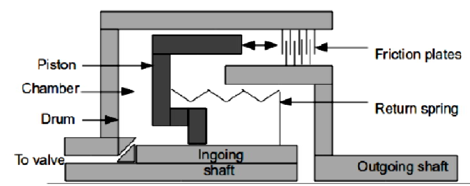


Fig. 1 Wet clutch design

The test bench used in this paper contains an electromotor and a flywheel. The electromotor drives a flywheel via two mechanical transmissions: one transmission is controlled in this project; the other transmission is used to vary the load and to adjust the braking torque. The sampling frequency on this setup is 1000Hz.

2.2 Feed forward control strategy and parameterized signal

Wet clutches used in industry are filled with a feed-forward parametric signal of the current to the electro-hydraulic valve. Fig. 2 shows a typical parameterized, feed-forward current signal, which is sent to the valve [11].

The shape of this signal perfectly illustrates the underlying idea behind the actual industrial control design. First, a step signal with height *a* and width *b* is sent to the valve to generate a high pressure level in the clutch. With this pressure, the piston will overcome the force from the return spring, and start to get closer to the clutch disks. After this pulse, the signal will give some lower current with height *c* and width *d* to decelerate the piston and trying to position it close to the clutch disks. Once the piston is close to the clutch disks and with very low velocity, a force is needed to push the piston forward so that the clutch disks are compressed together. This force will be generated by the pressure which is caused by the step current with height *e* and width *f*. Then a ramp current signal with slope alpha and the end height *g* is sent to the valve so that the pressure inside the clutch will increase again gradually. In order to secure the full closing of the clutch, the signal will be kept as a high current level afterwards. Many research efforts can be found in tuning such kind of signals in order to achieve a good engagement [2, 3, 12].

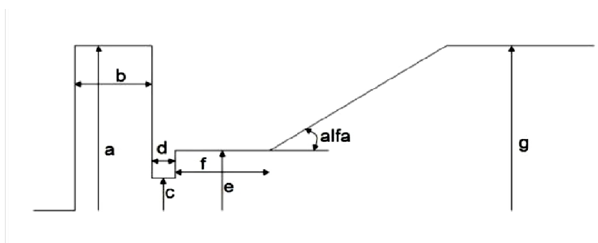


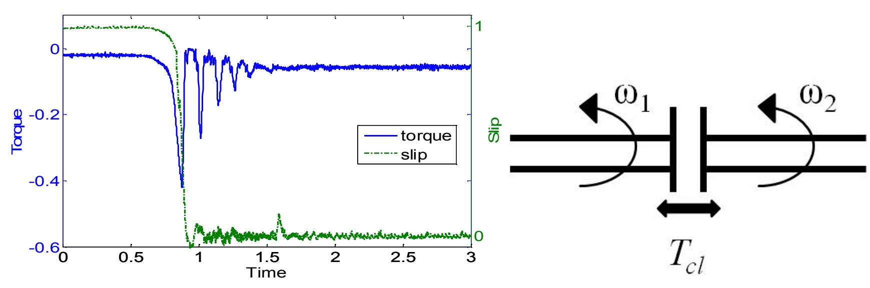
Fig. 2 Typical parameterized signal for controlling the wet clutch system

2.3 Theoretical modelling

In Fig. 3(a), an example of the output torque and slip of a wet clutch during the whole engagement process are illustrated. The unit for the torque is $N \cdot m$, while the slip is calculated by (Fig. 3(b)):

$$Slip = (\omega_1 - \omega_2)/\omega_1 \tag{1}$$

where ω_1 is the input speed, and ω_2 is the output speed, and the resulted slip has no unit. Since the existence of the breaking torque, the final transmission torque does not go to zero. For confidential reasons, all results in this paper will be scaled.



(a) (b)

Fig. 3 (a) Illustrative example of the performance of a wet clutch; (b) Torque transmission in the wet clutch

When closing a wet clutch, the inside of the system can be explained by some fundamental physical laws. An illustrative figure is given to show the closing of the wet clutch (Fig. 4).

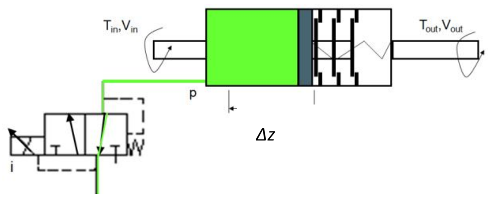


Fig. 4 The closing of a wet clutch, *i* is the triggle signal for the valve, *p* is the flow pressure, T_{in} and V_{in} are the input torque and velocity, T_{out} and V_{out} are the output torque and velocity, and Δz is the displacement of the piston.

The pressure of the fluid in the left side of the chamber is denoted as p_c, and if the area of the piston is known as A, then the normal force applied on the piston F can be expressed as:

$$F = p_c \cdot A \tag{2}$$

and by using Newton's law, we can have:

$$F = m \cdot \ddot{z} + b \cdot \dot{z} + k(z + z_0) \tag{3}$$

where m is the mass of the piston, b is the total damping efficient in the chamber, k is the spring constant of the return spring, and z_0 is the position of the piston when the clutch is fully open.

In Fig. 3(b), we schematically illustrate the torque transmission in a wet clutch. The rotational motion of the input shaft is on the left side with an angular velocity ω_1, and the output shaft is on the other side of the figure with an angular velocity ω_2. The transmission torque on the clutch is denoted as T_{cl}.

The transmission torque T_{cl} can be expressed as a function of slip [13]:

$$T_{cl} = g[(\omega_1 - \omega_2)] \tag{4}$$

and:

$$T_{cl} = \mu \cdot (\omega_1 - \omega_2) \tag{5}$$

where μ is a generalized coefficient which is proportional to the velocity of the piston. From the above analysis, we can notice that the position and the velocity of the piston are the key factors for achieving a good engagement. Unfortunately, it is impossible to use any sensor to measure either the position or the velocity of the piston. To overcome this drawback, a neural predictor can be used to predict the position and velocity of the piston.

3 A process neural network and its input and outputs

3.1 Inputs and outputs

From the analysis in section 2.3, we can understand that the purpose for training a neural predictor is to predict the position and the velocity of the piston with regards to the input pressure. Since there is no measurement for either position or velocity, we have to obtain the information based on the measureable outputs.

On the test bench, the available measurable variables are input rotational speed, output rotational speed, and transmission torque. Based on the input and output rotational speed, slip value can be calculated by (1). It can be assumed that when the piston has moved 90% of the complete distance to the friction plates, the output shaft starts to accelerate. From this position, the speed of the piston is critical since it will directly affect the transmission torque. Thus, we can approximate the position and velocity based on the slip and torque respectively. And for the position, it is very hard to define in accurate, so we set the position as a binary variable, and defined as [14]:

$$z(t) = \begin{cases} 0, & Slip \geq 0.95 \\ 1, & Slip < 0.95 \end{cases} \tag{6}$$

And the velocity:

$$v(t) = K_v[T(t) - T(t-1)] \tag{7}$$

where $T(t)$ is the torque measurement, and K_v is the gain. The value we get is not the real velocity, but this can give an indication of the speed changes according to the different input pressure profiles. The input of the system is the pressure $p(t)$. In the training process, since K_v and A are constants, the values can be arbitrary. For simplification, these values are set as 1.

3.2 Process neuron model

Since the inputs and the outputs are all time-varying functions, a conventional neuron model is not suitable for this case study. A process neuron and process neuron network [15], similar to additive model [16], will be implemented here to obtain a neural predictor.

The major difference between a process neuron and a traditional artificial neuron is in the process neuron, i.e., the inputs and the synaptic weights are time-varying functions. Additionally, besides the linear combiner to sum up the weighted inputs, there is also a time aggregation operator, which can integrate the input time-varying signal. Fig. 5(a) shows a process neuron model.

Therefore, we can write the output y as:

$$y = \varphi\left(\left(\int_0^T \left(\sum_{i=1}^{n} w_i(t)x_i(t)\right)\right) dt - B_k(t)\right) \tag{8}$$

If considering a neural network which is composed by neurons presented in (8), we can have the output of this neural network $y(t)$ as:

$$y(t) = \sum_{j=1}^{m} v_j(t)\, f_j\left(\int_0^T \left(\sum_{i=1}^{n} w_{ij}(t)x_{ij}(t)\right)dt - B_j(t)\right) \tag{9}$$

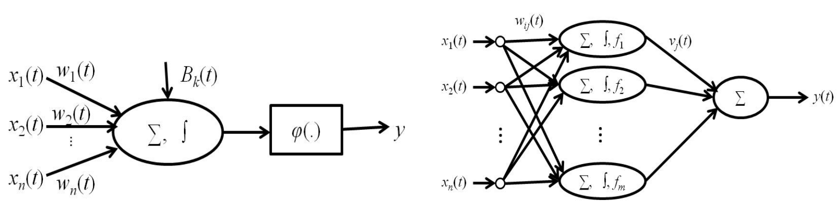
where $v_j(t)$ is the connection weight from the hidden layer to the output layer; n is the number of inputs; and m is the number of neurons in the hidden layer (see Fig. 5(b)).

Taking the pressure $p(t)$ as the input and the position $z(t)$ as the output, and $z'(t)$ as the desired output for the given input, the error of the neural network can be written as:

$$e = \left\| z(t) - z'(t) \right\| \tag{10}$$

where $z\,(t)$ is:

$$z(t) = \sum_{j=1}^{m} v_j(t) f\left(\int_0^T w_j p(t)dt - \varphi_j(t)\right) \tag{11}$$



(a) (b)

Fig. 5 (a)Process neuron model, $x_1(t)$, $x_2(t)$,…$x_n(t)$ are the time –varying inputs; $w_1(t)$, $w_2(t)$,…$w_n(t)$ are the time –varying connection weights; B_k is the bias of the neuron, also a time-varying function; $\sum$ is the adding operator; $\int$ is the time aggregation operator; $\varphi(.)$ is the activation function; and y is the output; (b) Process neural network

The learning rules for the connection weights and bias are:

$$w_{ij}^{k}(t) = w_{ij}^{k-1}(t) + \alpha\Delta w_{ij}^{k-1}(t) \tag{12}$$

$$v_j^{k}(t) = v_j^{k-1}(t) + \beta\Delta v_j^{k-1}(t) \tag{13}$$

$$\varphi_j^{k}(t) = \varphi_j^{k-1}(t) + \gamma\Delta\varphi_j^{k-1}(t) \tag{14}$$

where α, β, and γ are the learning rates, k is the number of iteration.

Finally, we have that:

$$\Delta w_{ij}^{k}(t) = -\partial e/\partial w_{ij}^{k-1}(t) \quad (15)$$
$$\Delta \mathrm{v}_{\mathrm{j}}^{\mathrm{k}}(\mathrm{t}) = -\partial e/\ \partial \mathrm{v}_{\mathrm{j}}^{\mathrm{k-1}}(\mathrm{t}) \quad (16)$$
$$\Delta \varphi_{\mathrm{j}}^{\mathrm{k}}(\mathrm{t}) = -\ \partial \mathrm{e}/\ \partial \varphi_{\mathrm{j}}^{\mathrm{k-1}}(\mathrm{t}) \quad (17)$$

And the same method is used to build the neural model for the velocity prediction.

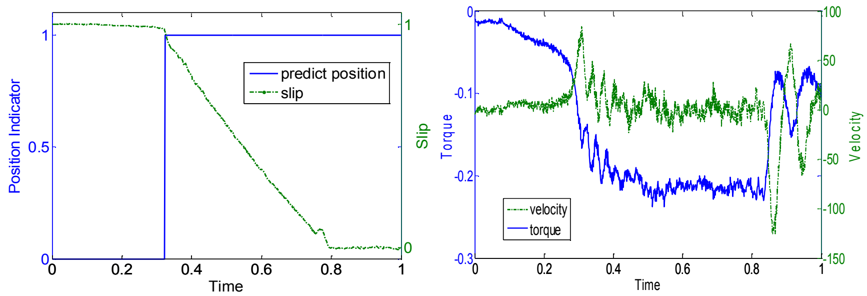
4 Experiments and Results

The data used to train and validate the neural network is from [2]. In this paper, engagement time and the torque loss are treated as two objectives to be optimized simultaneously. The tuning target is the parametric signal in Fig. 2. A nondominated sorting genetic algorithm (*ns*GA) [17] is used in order to obtain the Pareto solutions. In this method, the nondominated solutions constituting a nondominated front are assigned the same dummy fitness value. Nondomination here is defined as: in a minimization problem, if a vector $X1$ is partially less than $X2$, we say $X1$ dominates $X2$, and any member of such vectors that is not dominated by any other member is said to be nondominated [17]. These solutions are ignored in the further classification process. Then the front is extracted. This procedure is repeated until all individuals in the population are classified.

The genetic algorithm has a population size equal to 50, and stops at 8^{th} generation. So in that paper [2], totally there are 400 test runs. Some of the individuals didn't pass the safety check, and were penalized. The safety check is defined as "For safety reasons, the individuals are tested under low external load and low breaking inertial condition in laboratory environments. Before sending the individuals to test, all the individuals are first tested under no external load and no breaking inertial condition for safety reasons. If the reading of the torque loss under such working condition is larger than 200 N·m for any individual, then it is considered as "unsafe", and will not be tested further." Individuals which failed to pass the safety check are not used in this paper, so the total number of remaining test runs is 369. In this study, 70% of the data are used for training and the remaining data is used for validating the network. The neural network has 1 input and 2 outputs, and 5 neurons in the hidden layer, so the structure is 1-5-2.

Some validation results are shown in Fig. 6. We can find that the obtained neural network can provide a very good forecast on the position and velocity of the piston. In

Fig. 6(a), the slip drops below 0.95 at 0.327s, while the neural network gives the position indicator switch from 0 to 1 at 0.324s. In Fig. 6(b), we can notice that from 0.278s to 0.309s, the transmission torque experiences a sudden drop, and the neural network at this time gives the prediction of the velocity at this time will be very large.



(a) (b)

Fig. 6 (a) Predictive position indicator and slip; (b) Predictive velocity and transmission torque

5 Concluding remarks and future work

In this paper, a process neural network is developed for predicting the position and velocity of the piston in a wet clutch system during its engagement. The input and outputs of the neural network are selected based on the fundamental analysis of the physical behavior of the wet clutch. The resulted neural network shows a good prediction on the position and velocity for a given input pressure profile.

Next, this neural network can be used for designing a pressure profile which can have the output as: switch on the position indicator as fast as possible while keep the velocity at this moment as low as possible. We expect that such pressure profiles can ensure a smooth and fast engagement on the test bench.

Acknowledgements

This work has been carried out within the framework of the LeCoPro project (grant nr. 80032). C.M. Ionescu is financially supported by the Flanders Research Center (FWO).

References

[1] W.D., Widanage, J. Stoev, A. Van Mulders, J. Schoukens, and G. Pinte, "Nonlinear system-identification of the filling phase of a wet-clutch system", *Control Engineering Practice*, vol. 19(12), pp. 1506-1516, 2011.
[2] Y., Zhong, B., Wyns, R. De Keyser, G. Pinte, and J. Stoev, "An implementation of genetic-based learning classifier system on a wet clutch system", *14th Conference of the ASMDA International Society*, pp. 1924-1931, Rome, Italy, 2011.
[3] B. Depraetere, G. Pinte, and J. Swevers, "Iterative optimization of the filling phase of wet clutches", *The 11th International Workshop on Advanced Motion Control*, pp. 94-99, Nagaoka, Japan, 2010.
[4] B., Depraetere, G., Pinte, and J., Swevers, "A reference free iterative learning strategy", *30th Benelux Meeting on Systems and Control*, Lommel, Belgium, 2011.
[5] A. Dutta, R., De Keyser, Y., Zhong, G., Pinte, J., Stoev and B., Wyns, "Robust predictive control of a wet-clutch using evolutionary algorithm optimized engagement profile", *15th International Conference on System Theory, Control, and Computing*, Sinaia, Romania, 2011.
[6] M., Gagliolo, K., Van Vaerenbergh, A., Rodriguez, A., Nowe, S., Goossens, G., Pinte, and W., Symens, "Policy search reinforcement learning for automatic wet clutch engagement", *15th International Conference on System Theory, Control, and Computing*, Sinaia, Romania, 2011.
[7] S., Haykin, "Neural networks: a comprehensive foundation", Printice Hall, USA, 1999.
[8] A., Ghazi Zadeh; A., Fahim, and M., El-Gindy, "Neural network and fuzzy logic applications to vehicle system: literature survey", *International Journal of Vehicle Design*, volume 18(2), pp. 93-132, 1997.
[9] M.W., Roth, "Survey of neural network technology for automatic target recognition", *IEEE Transactions on Neural Networks*, volume 1(1), pp 28-43, 1990.
[10] J., Lake, and B., Romas, "Self-contained wet clutch system", US Patent No. 3,314,513, 1967.
[11] K.V., Hebbale, C.-K., Kao, and D.E., McCulloch, "Adaptive electronic control of power-on upshifting in an autamatic transmission" US Patent No. 5,282,401, 1994.
[12] C., Lazar, C.-F., Caruntu, and A.-E., Balau, "Modelling and predictive control of an electro-hydraulic actuated wet clutch for automatic transmission", *IEEE International Symposium on Industrial Electronics*, pp. 256-261, Bari, Indonesia, 2010.
[13] M., Dassen, "Modelling and control of automotive clutch systems", in: Report number 2003.73, Eindhoven, 2003. http://alexandria.tue.nl/repository/books/615474.pdf
[14] G. Pinte, J. Stoev, W., Symens, A., Dutta, Y., Zhong, B., Wyns, R., De Keyser, B., Depraetere, J., Swevers, M., Gagliolom and A., Nowe, "Learning Strategies for wet clutch control", *15th International Conference on System Theory, Control, and Computing*, Sinaia, Romania, 2011
[15] X.G., He, and S.H., Xu, "Process neural networks and its application in time-varying information processing", *CAAI Transactions on Intelligent Systems*, volume 1(1), pp. 1-6, 2006.
[16] W., Rall, "Cable theory for dendritic neurons", in: *Methods in Neuronal Modeling*, pp. 9-62, MIT Press, Cambridge, MA, USA, 1989.
[17] N., Srinivas, and K., Deb, "Multiobjective Function Optimization Using Nondominated Sorting Genetic Algorithms", *Evolutionary Computation*, vol. 3, pp. 221-248, 1995