DESIGN AND REALIZATION OF PROGRAMMABLE EMULATOR OF MECHANICAL LOADS

Milan Žalman, Radovan Macko

The Slovak University of Technology in Bratislava, Slovakia

Abstract: This paper concerns design and realization of programmable emulator of mechanical loads for motors. The aim of this paper is to build integrated test equipment, which will allow realistic testing of drive parameters and mainly quality of control structures on controlling different kinds of loads. For this purpose create user-friendly software which will allow emulation of physical system in real-time using xPC Target under Matlab 6.1.

Keywords: Load Emulation, Mechanical Loads, Drive Control

1. INTRODUCTION

Electric drives have a wide range of application in all industries. They are used to drive, translation, stroke, positioning etc. These processes often contain linear, non-linear or time varying inertial loads. Therefore a demand for examination of motor characteristics and testing of motors has come up. For testing, research and development purposes it is not possible to have all kinds of different mechanical loads in laboratory due to financial and space reasons. Convenient solution is to emulate mechanical load electrically, using electromotor. This allows experimental testing of non-linear, adaptive, robust and intelligent control algorithms and testing of motor properties.

Aim of this paper is to design and physically realize load emulator, so the user can easily choose type of load and set up desired parameters and verify his own design of control algorithm.

2. BASIC LOAD EMULATION METHODS

In load emulation, the aim is to produce an OLTF which may be given by any relation between input M_m and output ω. It is perhaps best to start with the simplest load emulation which can be written by

$$\frac{\omega(s)}{M_m(s)} = \frac{1}{J_{em}s + B_{em}} = G_{em}(s) \qquad (1)$$

where J_{em} and B_{em} is the emulated inertia and friction and it is going to be defined by user. Now we need to produce load machine control structure so that the relation between shaft speed ω(s) and the electrical driving motor torque $M_m(s)$ is given by equation (1).

1.1 Inverse Model Method

Principle of this method will be explained using Figure (1).

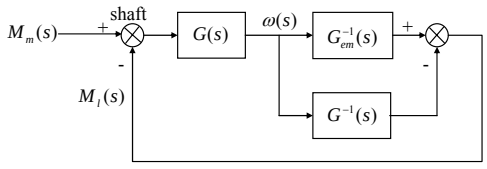

Fig. 1 Inverse model

In order to emulate mechanical load G_{em} the easiest way is to use Inverse model. $G_{em}(s)$ represents the mechanical load to be emulated, $G_{em}^{-1}(s)$ its inverse model, G(s) represents the system of motor and load machine and $G^{-1}(s)$ its inverse model. For example:

$$G(s) = \frac{1}{Js + B} \qquad (2)$$

$$G_{em}^{-1}(s) = J_{em}s + B_{em} \tag{3}$$

J is moment of inertia (kgm^2) and B is viscous-friction coefficient (Nms). According to Figure (1) we can say that equation (1) is valid.

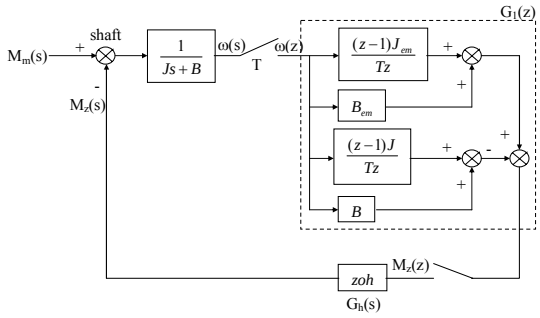

Fig. 2 Sampled inverse model system

In practice the load machine controller will be realized in a discrete form, which means we have to consider sampling of signals. Therefore we transform the continuous system into discrete system as you can see in Fig.2 using sample period T, where $G_h(s)$ is the zero order hold. If this control is implemented in discrete form, output of inverse model can be very noisy due to derivative terms in inverse function, and it is necessary to resolve stability of the system. From Figure (2) we can write

$$\omega(z) = \frac{M_m G(z)}{1 + G_h G(z) G_1(z)} \tag{4}$$

Then we can derive $G_1(z)$ as

$$G_1(z) = \frac{M(z)}{\omega(z)} = \frac{(\Delta J + \Delta BT)z - \Delta J}{Tz} \tag{5}$$

Term $G_h(s)G(s)$ after discretization becomes

$$G_h(z)G(z) = \frac{(z-1)}{z} Z\left\{\frac{1}{s} \cdot \frac{(1/B)}{(J/B)s+1}\right\} = \frac{1-D}{B(z-D)} \tag{6}$$

where $D = e^{-\frac{BT}{J}}$, $\Delta J = J_{em} - J$, $\Delta B_{em} = B_{em} - B$

Then OLTF ($G_h(z)G(z)G_1(z)$) is

$$OLTF = \frac{(\Delta J + \Delta BT)(1-D)(z-\alpha)}{BTz(z-D)} \tag{7}$$

where

$$\alpha = \frac{\Delta J}{\Delta J + \Delta BT} \tag{8}$$

The closed loop characteristic equation can be written as

$$z^2 + z\underbrace{\left(-D + \frac{(\Delta J + \Delta BT)(1-D)}{BT}\right)}_{\beta_1} + \underbrace{\frac{\alpha}{BT}(\Delta J + \Delta BT)(D-1)}_{\beta_2} = 0 \tag{9}$$

In general B will be close to zero. Therefore we use simplifications:

$$D = e^{-\frac{BT}{J}} \cong 1 - \frac{BT}{J} \qquad 1 - D = \frac{BT}{J} \tag{10}$$

After substitution into characteristic equation β_1 and β_2 becomes

$$\beta_1 = \frac{J_{em} - 2J + B_{em}T}{J} \text{ and } \beta_2 = -\frac{\Delta J}{J} \tag{11}$$

Take note that β_1 and β_2 are independent off the viscous friction value B, which means that stability is not affected by value of B. If we consider that B_{em} is small or zero, the characteristic equation becomes

$$(z-1)\left(z + \frac{\Delta J}{J}\right) = 0 \tag{12}$$

This indicates that the system is unstable when $\Delta J/J > 1$ or $J_{em} > 2J$.

If we consider that $\Delta J=0$ and B_{em} is not zero or small the characteristic equation becomes

$$z\left(z - 1 + \frac{B_{em}T}{J}\right) = 0 \tag{13}$$

In this case the system is unstable if $B_{em}T/J > 2$. The system can be stabilized using discrete filter of the form (14) to lower the noise of M_z and by doing this increase the use of this method.

$$\frac{z(1-\gamma)}{z-\gamma} \tag{14}$$

Using this filter we stabilized the system and increased the area of use significantly, however, the zero pole structure of desired mechanical load is violated which leads to unsatisfactory results in a closed loop control system. For this reason is the Inverse model method which is used in previous researches (Collins and Huang, 1994; Sandholdt, *et.al.*, 1996; Betz, *et.al.*, 1994; Newton, *et.al.*, 1995) not in place for our emulator since we want to preserve the real mechanical dynamics during emulation.

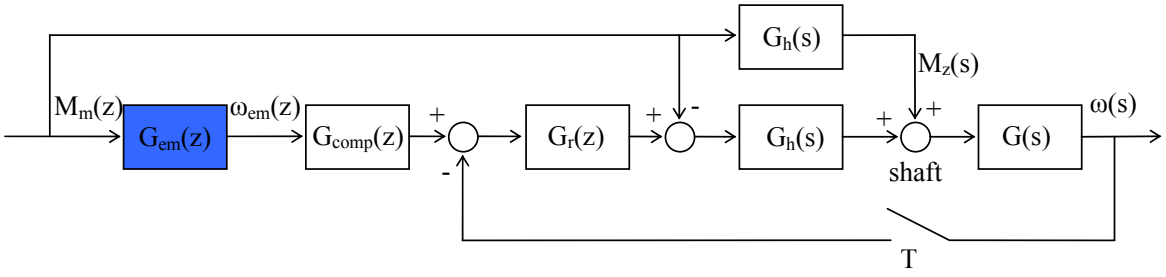

Fig. 3 Block system of "open loop" method

1.2 "Open loop" emulation Method

Consider the system in Figure (3). Based on Motor Torque M_m and required load to be emulated, ideal speed ω_{em} is computed. This is the speed at which would the real motor shaft rotate with the desired load. This ideal speed is compared with real shaft speed ω and modified by G_{comp}. Based on the difference the controller $G_r(z)$ derives load torque M_z for the load machine. The total of these two torques from both machines (load and motor), forces the system to travel at the desired ideal speed ω_{em}. The reason why this method is called "open loop" is that, the transfer function ω/M_m is controlled in open loop (Akpolat et.al., 1998). The function of G_{comp} is to compensate all parts of the open loop except the emulated load function, so that the open loop discrete transfer function, $\omega(z)/M_m(z)$ becomes equal to discretized equivalent of $G_{em}(s)$.The transfer function $G_{em}(s)$ includes mechanical parameters of both motors. Therefore the transfer function of G_{comp} is

$$G_{comp}(z)=\frac{1+G_r(z)G(z)}{G_r(z)G(z)} \qquad (15)$$

where $G(z)=Z\{G_h(s)G(s)\}$ and $G_r(z)$ is discrete PI controller. Controller $G_r(z)$ is given by

$$G_r(z)=\frac{K_r(z-A_r)}{z-1} \qquad (16)$$

In order to complete the control structure of "Open loop" method we need to define G_{comp}.

$$G_{comp}(z)=\frac{a_2z^2+a_1z+a_0}{b_2z-b_1}\frac{1}{z} \qquad (17)$$

Where the constants are given as

$b_2=1$, $b_1=-A_r$, $b_0=0$

$$a_2=\frac{J}{K_rT}\quad a_1=1-\frac{J(1+D)}{K_rT}\quad a_0=\frac{JD}{K_rT}-A_r$$

The delay which we added to G_{comp} is compensated by z, which we multiplied the load transfer function with. If the emulated load is to be that of inertia J_{em} and friction B_{em}, the discretized load transfer function G_{em} becomes

$$G_{em}(z)=z*Z\left\{\frac{1}{s}\frac{1}{J_{em}s+B_{em}}\right\}=z\frac{(1-D_1)}{B_{em}(z-D_1)} \qquad (18)$$

where $D_1=e^{-\frac{B_{em}T}{J_{em}}}$

Compensative term G_{comp} cancelled all parts of open loop system except of emulated load transfer function so, that the final reduced system matches system in Figure (4).

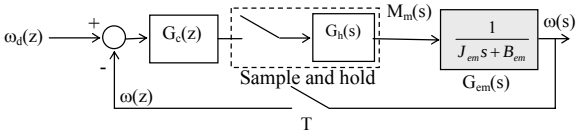

Fig. 4 Closed loop speed control system

Now we can apply speed or position control methods to verify the quality of emulation desired loads. This method preserves the dynamics of chosen load and is suitable for realization of our emulator. Notice that block G_{comp} is independent of G_{em}. This is very handy since we can calculate it once and then we can choose different kind of loads without the need to calculate it again.

Programmable load emulator in general is equipment which consists off controlled load machine and is tightly mechanically coupled with the motor (tested motor) as shown on Figure (5). Load machine with special control (substitutes) emulates technological load. Desired load type is obtained using a suitable control of load machine.

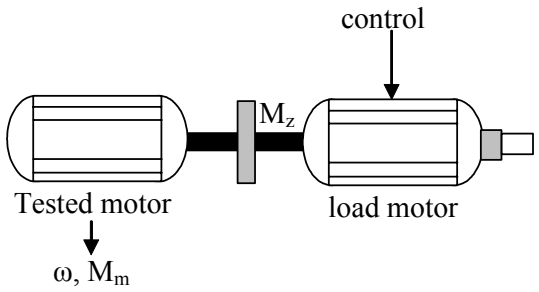

Fig. 5 General scheme of load emulator

3. EXPERIMENTAL AND SIMULATION RESULTS

Physically is emulator realized as a system of two tightly coupled 140W DC motors. Both motors are fed by transistor converter. Figure (6) shows Basic simulation/experimental system used to compare simulation and experimental results of load emulator using described method with different kind of loads.

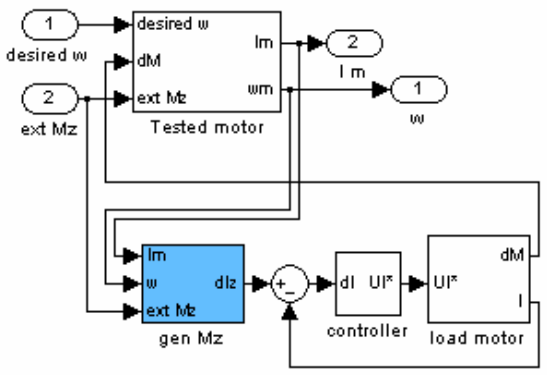

Fig. 6 Basic simulation/experimental system

This system is universal. To change the type of emulated load you simply change G_{em} in block gen Mz as shown on Figure (7).

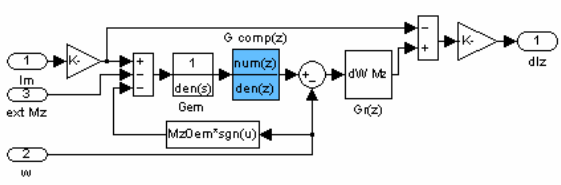

Fig. 7 load torque generator structure (linear load)

We created a simulation model of the motor with defined load so that we have some data to compare the experimental results with. First we checked the functionality of the emulator by doing an experiment

in which we applied linear load to the motor, where G_{em} was 10 times the value of system inertia and B_{em} was equal to B. We also applied static friction (M_{z0em}=0.05Nm) in the model as in the emulator because it causes problems to motion control systems. The result of this experiment is presented on Figure (8) and Figure (9).

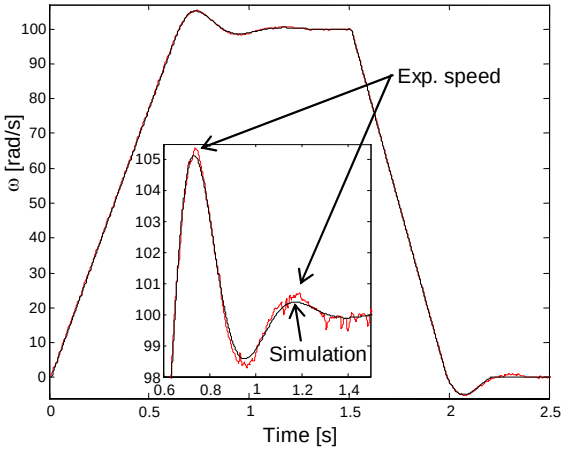

Fig. 8 Comparison of speed responses for J_{em}=10J, B_{em}=B and M_{z0em}=0.05 Nm

From these two figures you can see that emulation works very good and the simulation responses are very close to experimental responses of speed and torque.

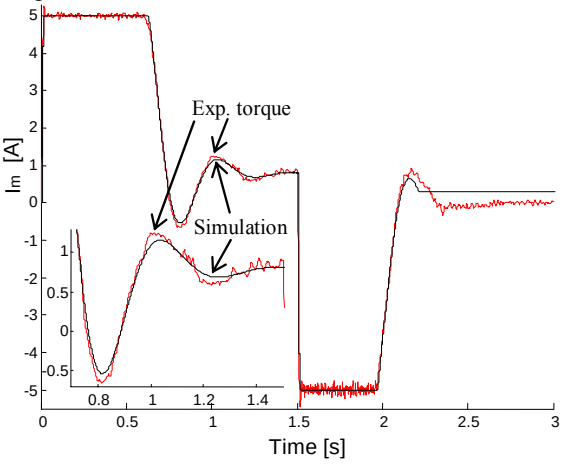

Fig. 9 Comparison of torque responses for J_{em}=10J, B_{em}=B and M_{z0em}=0.05 Nm

As you can see the only difference between simulation end experimental results are notable at the end when the motor is stopping and speed is zero. This difference is caused only in simulation by static friction because the exact model was not used. However this is only simulation problem when the angular speed is close to zero and the experimental results are satisfactory and correct.

One of the most complicated types of load is 2 mass system connected with flexible coupling, which is possible to find in various applications in praxis. In order to check, if the emulator is able to act also as 2 mass system, we used block diagram from Figure (6) with modified block gen M_z from Figure (10).

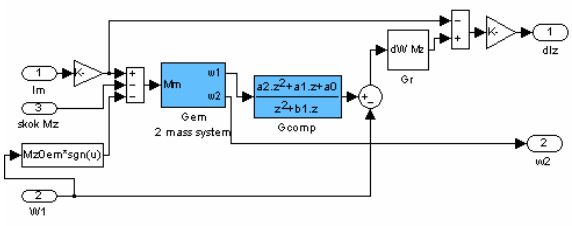

Fig. 10 Gen Mz structure for emulation of 2 mass system with flexible coupling

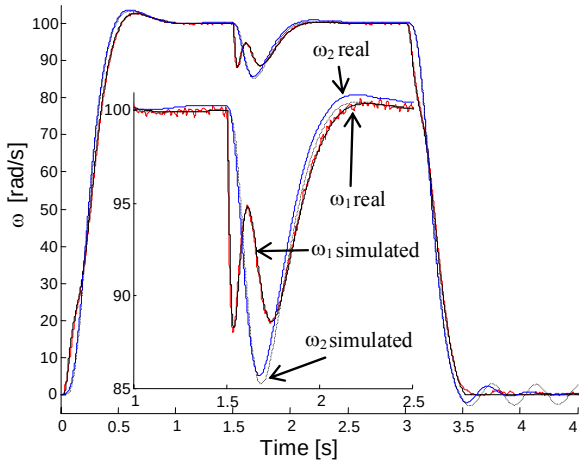

Fig. 11 Comparison of speed responses for J_{em}=3J, C_t=0.1 Nm.s/rad and zero dumping coefficient

In next experiment we emulated two-inertia system without dumping (most problematic case), the torsional spring constant C_t=0.1Nm.s/rad and inertia ratio $R=J_2/J_1=3$. We applied step of external load (50% of the motor rated torque) disturbance at t=1.5s. For speed control of this system we used controller designed with the method of pole placement by identical dumping coefficients (Zhang and Furusho, 2000). The coefficients used were $\xi_1=\xi_2$=0.865. Figure (11) and Figure (12) shows the comparison of experimental and simulation results of emulation two inertia load.

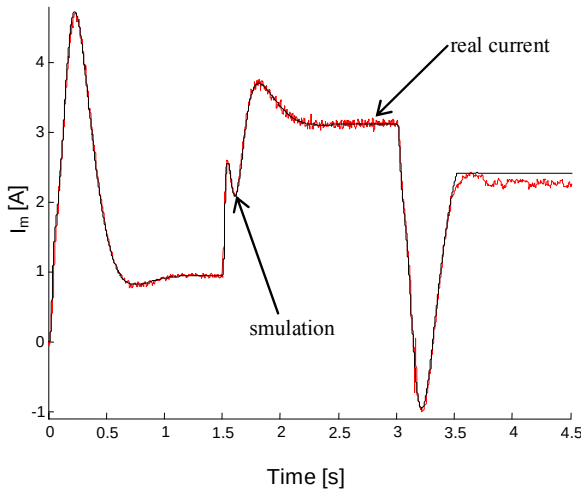

Fig. 12 Comparison of torque responses for J_{em}=3J, C_t=0.1 Nm.s/rad and zero dumping coefficient

As you can see the simulation results are very close to results of experimental system. Load speed (ω_2) control was indirect, in other words in closed loop was controlled only angular speed of (ω_1) motor. Even under these conditions emulation gives satisfactory results.
In general load emulator can be realized with any pair of motors. What we need to know is mechanical parameters of such motor couple (J and B), information about actual angular speed and information of motor torque M_m. This emulator as presented can be used to emulate different kind of linear or non-linear loads. We can reduce the area of use to schematic diagram on Figure (13).

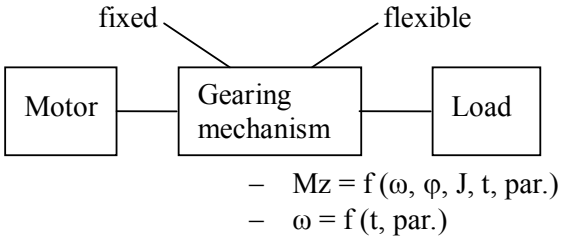

Fig. 13 Loads able to be emulated

3. HW AND SW REALIZATION

Emulator was realized with 2 DC motors. The control strategy was employed using Matlab real time and xPC target libraries. The model was created on the PC with windows and Matlab and then compiled using C compiler into executable file. These files were executed on second PC with xPC operating system. Transfer between the 2 computers is automatic via Ethernet adapters. This method allows us to emulate complicated loads with high computing power needs. The process of creating, compiling, transferring, running and getting measured data is quite complicated.
Therefore in order to ease the way this emulator is used in praxis, user friendly software was created under Matlab. You can see its main window on Figure (14).

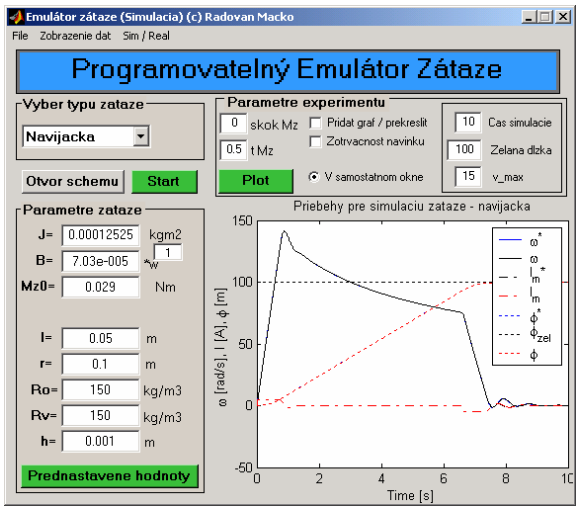

Fig. 14 Main window of software interface.

User interface is very convenient when testing a drive with same kind of loads only different parameter setup. There are 4 predefined load types:
1 – linear – centrifugal
2 – fan
3 – winding machine
4 – 2 mass system with flexible shaft

4. CONCLUSIONS

In this paper programmable load emulator for dynamic emulation of linear and non-linear mechanical loads has been presented. Dynamics of desired load is preserved during emulation. Realized tool will help future student generations to design and test robust, intelligent, adaptive or fuzzy control algorithms for controlling non-linear, two-inertia or time varying loads. Emulation and intelligent control of two inertia systems with flexible coupling and static friction will be the subject of future research.

REFERENCES

Collins, E.R. and Y. Huang (1994). A programmable Dynamometer for Testing Rotating Machinery Using Three-Phase Induction Machine. *IEEE Transactions on Energy Conversion 9,* **3**, 521-527.

Sandholdt, P., E. Ritchie, J.K. Pederson and R.E. Betz (1996). A Dynamometer Performing Dynamical Emulation of Loads with Non-Linear Friction. *IEEE Int. Symposium on Industrial Electronics,* **2**, 873-878.

Betz, R.E., H.B. Penfold and R.W. Newton (1994). Local Vector Control of an AC Drive System Load Simulator. *IEEE Conference on Control Applications-Proceedings,* **1**, 721-726.

Newton, R.W., R.E. Betz and H.B. Penfold (1995). Emulating Dynamical Load Characteristics Using a Dynamic Dynamometer. *Proceedings of Int. Conference on Power Electronics and Drive Systems*, **1**, 465-470.

Akpolat, Z.H., G.M. Asher and J.C. Clare (1998). Emulation of high bandwidth mechanical loads using vector controlled AC dynamometer. *Proceedings 8th Int. Power Electronics and Motion Control Conference*, **5**, 133 - 138.

Akpolat, Z.H., G.M. Asher and J.C. Clare (1999). Experimental dynamometer emulation of nonlinear mechanical loads. *IEEE , Transactions on industry applications*, **vol.35**, No.6.

Zhang, G. and J. Furusho, (2000). Speed Control of Two-Inertia System by PI/PID Control. *IEEE, Transactions on industrial electronics,* **vol. 47**, No.3.