

# The model of a permanent magnet DC motor in time domain and frequency domain based on Bond Graph modelling and design of position control using PID controller

Gy Korsoveczki<sup>1</sup>, G Husi<sup>1</sup>

<sup>1</sup>Mechatronics Department, Faculty of Engineering, University of Debrecen, Debrecen, Hungary

korsoveczki.gyula@gmail.com

**Abstract.** During the research work, the modelling of a permanent magnetic DC (PMDC) motor was realised in time domain and frequency domain via 4 different types of model construction. To achieve this, the Bond Graph model of the motor was determined, and based on them the system equations were defined in time domain firstly and then in frequency domain by using the Laplace transform. According to the system equations in time domain, the state-space model of the motor was also realised and implemented. Based on the system equations in frequency domain, the transfer function of the position control loop was realised. As the further part of the research work, PID control-based position control was designed in both time domain and frequency domain. The controller was tuned according to the Ziegler-Nichols method. Finally, the models were compared from the point of view of behaviour during the tuning process. The models were implemented in LabVIEW environment produced by the National Instruments.

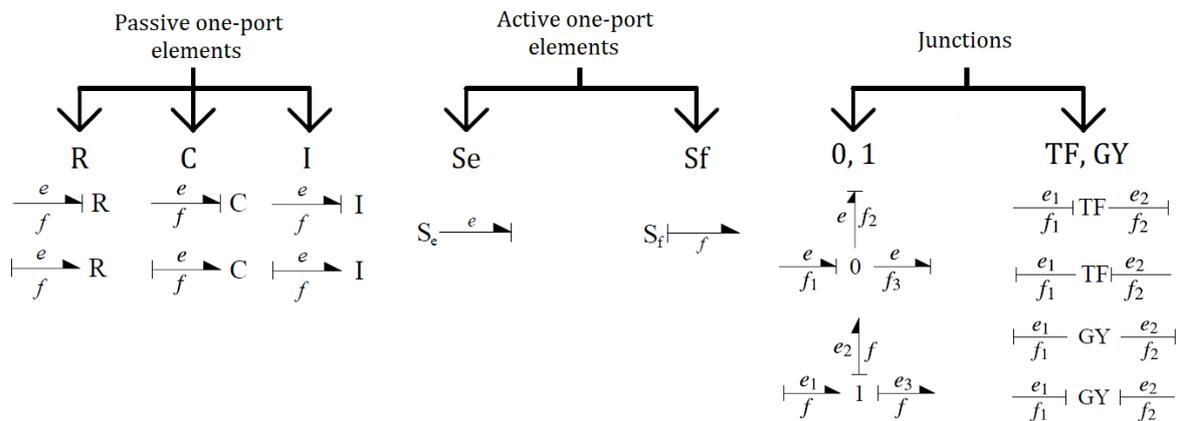
## 1. Introduction

Nowadays the technical life is producing high-rate development. As the available technological background evolves, the increase in complexity of systems becomes indispensable. Systems can operate only in one domain, but it is often necessary to establish a connection between systems of different domains. The field of mechatronics seeks to bridge the differences between systems. It is important to have accurately detailed planning and system modelling that represents the behaviour of the real physical system. The purpose of this research is to show the process and the possibilities of the modelling of a mechatronic system by using different modelling and implementation methods [1], [2].

## 2. The Bond Graph modelling

System modelling in engineering should distinguish between Single Domain systems and Multi-Domain systems. Single Domain systems can be purely mechanical, electrical, hydraulic or thermal etc. systems. They have their own description languages that can be used for modelling, understanding and analysing. If a system (such as a mechatronics system) contains more than one Single Domain system which is different types, it is called Multi-Domain system. It is necessary to realise physical or informational connection between them, as it happens in real physical work. It is difficult to establish mathematical relations between domains because the only way is to find a variable that is present in all systems. This physical variable is energy, and the language describing its change is Bond Graph modelling, developed by Henry Paynter in 1959. Bond Graph is an energy-based description language that tracks the change

of energy across the domains. It is an effective solution for modelling Multi-Domain systems such as mechatronic systems. The Bond Graph is based on the fact that the transmitted power is the multiplication of the power variables, which are effort and flow. The role of them is unchangeable and it is defined for all systems. The Bond Graph has its own graphical description language that contains basic one-port passive elements (R, C and I element) that do not add new energy to the system but dissipate (R elements) or store it as potential energy (C elements) or kinetic (I elements) energy of the system. There are active one-port elements ( $S_e$ ,  $S_f$ ), which can produce new effort or flow for the system. Junctions are also parts of Bond Graph, which can be two-port (TF, GY) or three-port (0, 1) junctions. The most important element of the Bond Graph modelling is the causality that defines the directions of effort and flow and allows us to determine the system equations [2], [3].

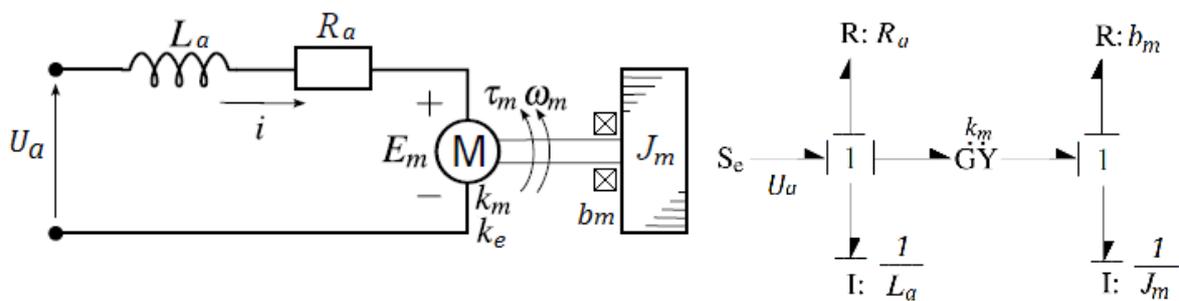


**Figure 1.** The basic elements of the Bond Graph [3].

### 3. The Bond Graph model of an ideal PMDC motor and its modelling in time domain

#### 3.1. The Bond Graph model of the PMDC motor

A permanent magnet DC motor (PMDC motor) converts electrical power into mechanical rotating motion. It is a typical Multi-Domain system as it uses both electrical and mechanical power. The electrical part can be modelled as a simple serial circuit that is the equivalent circuit of the motor. The circuit consists of a voltage source ( $U_a$ ), an armature resistance ( $R_a$ ) and an armature coil ( $L_a$ ) inductance. Since the circuit itself is a serial connection, the current is constantly flowing in it, but each electrical unit has the own voltage value. In an electrical system, the voltage is effort, the current is flow, so the electrical part has constant flow but variable effort, which can be modelled with 1 junction. The voltage source is a source of effort ( $S_e$ ), the armature resistance is an R element, and the armature coil is an I element. The strong Bond of the system is the I element, and in terms of causality, the voltage source has flow-in-effort-out, the armature resistance also flow-in-effort-out, while the armature coil has an effort-in-flow-out causality [3].



**Figure 2.** The schematic of the PMDC motor and its Bond Graph model [3].

The electromechanical conversion is represented by a Gyrator (GY) element that has flow-in-effort-out causality and connects the flow of the electrical domain to the effort of the mechanical domain and the flow of the mechanical domain to the effort of the electrical domain. The ratio is the motor constant  $km$  [Vs/rad] or [Nm/A]). The mechanic domain includes a bearing, rotational inertia and the electromechanical conversion of the motor. Bearing ( $bm$ ) is represented by an R element with flow-in-effort-out causality, the rotational inertia ( $Jm$ ) is in I element with effort-in-flow-out causality [3].

### 3.2. The space-state model of the motor

As a result of the Bond graph modelling the system equations could be realised. These differential equations can be written in state-space structure, which is a matrix structure for a better process of a larger number of system equations. The derivate of state variables ( $\dot{\mathbf{y}}$ ), the state matrix (A), the state variables ( $\mathbf{y}$ ), the input matrix (B) and the input variables ( $\mathbf{u}$ ) were realised [4].

$$\frac{di}{dt} = \frac{U_a}{L_a} - \frac{R_a}{L_a} \cdot i - \frac{ke}{L_a} \cdot \omega_m \quad (1)$$

$$\frac{d\omega_m}{dt} = \frac{km}{J_m} \cdot i - \frac{b_m}{J_m} \cdot \omega_m \quad (2)$$

$$\frac{d}{dt} \begin{bmatrix} i \\ \omega_m \end{bmatrix} = \begin{bmatrix} -\frac{R_a}{L_a} & -\frac{ke}{L_a} \\ \frac{ke}{J_m} & -\frac{b_m}{J_m} \end{bmatrix} \cdot \begin{bmatrix} i \\ \omega_m \end{bmatrix} + \begin{bmatrix} \frac{1}{L_a} \\ 0 \end{bmatrix} \cdot U_a \quad (3)$$

## 4. Modelling the motor in frequency domain

The difficulty of modelling in time domain is to manage a large number of differential equations. In the case of the frequency domain, the Laplace transform simplifies the differential equations into simple algebraic equations, which can be easier processed and simulated. Using the Laplace transform, time domain system equations were transformed into frequency domain. In the frequency domain, the transfer function allows us to model a system. The transfer function defines the relation between the system output in frequency domain and the system input in frequency domain [4].

$$L_a \cdot i(s) \cdot s = U_a(s) - R_a \cdot i(s) - ke \cdot \omega_m(s) \quad (4)$$

$$J_m \cdot \omega_s(s) = km \cdot i(s) - b_m \cdot \omega_m(s) \quad (5)$$

$$G(s) = \frac{i(s)}{U_a(s)} = \frac{1}{L_a \cdot s + R_a} \quad (6)$$

$$G(s) = \frac{\omega_m(s)}{i(s)} = \frac{1}{J_m \cdot s + b_m} \cdot km \quad (7)$$

$$G(s) = \frac{\theta(s)}{U_a(s)} = \frac{k_m}{J_m \cdot L_a \cdot s^3 + (R_a \cdot J_m + b_m \cdot L_a) \cdot s^2 + (k_m^2 + R_a \cdot B_m) \cdot s} \quad (8)$$

## 5. The design of position control using PID controller in different domains

### 5.1. Generally about PID controllers

The control of linear systems is important in control theory. In this research, a position control loop for PMDC motor was designed in time domain and frequency domain using PID controller. It operates based on error signal, which is the difference between the reference value and the output value. The controller uses parallel compensation, thus the execution signal is proportional to the error signal (P), to the integral of the error signal (I) and to the derivate of the error signal (D). The system equations of PID controller in time domain and frequency domain are the following [4], [5]:

$$e(t) = K_p \cdot \left( e(t) + \frac{1}{T_i} \cdot \int_0^t e(t) dt + T_d \cdot \frac{de(t)}{dt} \right) \quad (9)$$

$$e(s) = K_p \cdot \left( 1 + \frac{1}{T_i \cdot s} + T_d \cdot s \right) \quad (10)$$

### 5.2. The design of the position control loop

The system of the PMDC motor was modelled with 4 different system models in 2 different domains (time and frequency). All four system models were based on the description language of the Bond Graph. The software environment of the implementation was LabVIEW, which is provided by National Instruments. In the first case, the system model was built based on Block diagram structure in time domain. The second model was the state-space model, which is a numerical solution and it is equivalent to the Block diagram. Both system models represent the system in the time domain. In frequency domain, the best modelling method is to determine the transfer functions. The system was modelled based on its transfer functions using block structure and numerical implementation in MathScript Node. Each system model contains position control loop using PID controller that ensures the stable behaviour of the system. In the case of the model, the  $\theta$  angle of the rotation is the output variable, And the input variable is the  $U_a$  input armature voltage [4], [5].

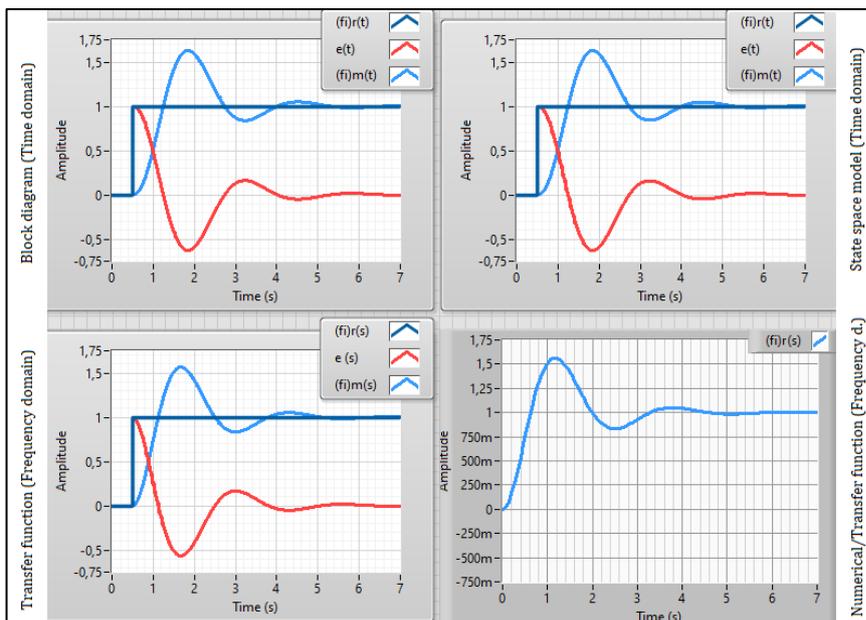
### 6. The tuning of the PID controller in different domains using the Ziegler-Nichols method

In cases of controllers which are based on the parallel compensation of linear systems such as PID, several methods are known for tuning such as the testing-based Trial and Error Method, the Cohen-Coon method that requires to open the loop of the control loop, and the Ziegler-Nichols method that is suitable and effective for tuning closed control loops. The Ziegler-Nichols method was developed by John G. Ziegler and Nathaniel B. Nichols. The method is when the closed control loop needs to be tuned up to the limit of instability, while the elements of the PID are determined one by one based on the critical amplification and the critical period time. The method is excellent for simulation modelling, where system unstableness does not cause physical damage and the single running of the control loop takes a short time. Changing the value of the proportional (P) part, the system must be controlled to the limit of instability, where critical amplification and its critical period time can be determined. In the meantime, the effects of the integrating (I) and derivative (D) part should be eliminated from the system by increasing the  $T_i$  integration time to infinite and reducing the  $T_d$  derivative time to zero. At the limit of stability, the system will resonate with constant amplitude around the reference signal. After defining the critical amplification ( $K_{Pkrit}$ ) and the critical period time ( $T_{krit}$ ), the values of the parts can be determined as the following relations show [4], [5]:

$$K_P = 0,6 \cdot K_{Pkrit} \quad (11)$$

$$T_i = 0,5 \cdot T_{krit} \quad (12)$$

$$T_D = 0,12 \cdot T_{krit} \quad (13)$$



**Figure 3.** The recognition of the pentagon object.

## 7. The simulation results

The simulation data were exported to Microsoft Excel environment, where it was evaluated. In the simulation, all 4 models produced the same behaviour, the simulation results were equivalent. The Ziegler-Nichols method was used in every case, the critical amplification ( $K_{pkrit}$ ) and critical period time ( $T_{krit}$ ) at the limit of instability were the same.

**Table 1.** The evaluated values of the controlling using PID controller.

	ODE Solver	Step Size (s)	Rise Time (s)	Overshoot (%)	Settling Time (s)	Steady-State Error
Block Diagram model (time)	Runge-Kutta 4	0,001	0,48	40	3,31	0,005
State-Space model (time)	Runge-Kutta 4	0,001	0,47	40	3,29	0,005
Transfer function (Block) (frequency)	Runge-Kutta 4	0,001	0,49	40	3,09	0,005
Transfer function (Numerical) (frequency)	Runge-Kutta 4	0,001	0,53	40	3,22	0,01

## Conclusion

In this study, an ideal PMDC motor was modelled using 4 different system models based on the graphical description language of the Bond Graph. In the case of every model, position control loop was implemented using PID controller. The controller was tuned using the Ziegler-Nichols method, in which the system had to be amplified to the limit of instability. In the tested domains, the systems had the same behaviour, provided equivalent data and their computing resource requirements were low.

## References

- [1] B. Rokseth, E. Pedersen, A Bond Graph Approach for Modelling of Robotic Manipulators, July 2014., [Online]. Available: [https://www.researchgate.net/publication/309410097\\_A\\_Bond\\_Graph\\_Approach\\_for\\_Modeling\\_of\\_Robotic\\_Manipulators](https://www.researchgate.net/publication/309410097_A_Bond_Graph_Approach_for_Modeling_of_Robotic_Manipulators). [Accessed: 10.05.2021.].
- [2] C. Apostoiaia, Multi-domain System Models Integration for Fault Detection in Induction Motor Drives, June 2014, [Online]. Available: [https://www.researchgate.net/publication/262974154\\_Multi-domain\\_System\\_Models\\_Integration\\_for\\_Faults\\_Detection\\_in\\_Induction\\_Motor\\_Drives](https://www.researchgate.net/publication/262974154_Multi-domain_System_Models_Integration_for_Faults_Detection_in_Induction_Motor_Drives). [Accessed: 10.05.2021.].
- [3] J. A. Kypuros, System Dynamics and Control with Bond Graph Modeling, CRC Press Taylor & Francies Group LLC, U.S., 2013.
- [4] M. Sulaiman, M. S. M. Aras, Tuning Methods of PID Controller for DC Motor Speed Control, August 2016, [Online]. Available: [https://www.researchgate.net/publication/311588542\\_Tuning\\_Methods\\_of\\_PID\\_Controller\\_for\\_DC\\_Motor\\_Speed\\_Control](https://www.researchgate.net/publication/311588542_Tuning_Methods_of_PID_Controller_for_DC_Motor_Speed_Control). [Accessed: 10.05.2021.].
- [5] M. M. Sabir, J. A. Khan, Optimal Design of PID Controller for the Speed Control of DC Motor by Using Metaheuristic Techniques, 2014, [Online]. Available: [https://www.researchgate.net/publication/277951362\\_Optimal\\_Design\\_of\\_PID\\_Controller\\_f\\_or\\_the\\_Speed\\_Control\\_of\\_DC\\_Motor\\_by\\_Using\\_Metaheuristic\\_Techniques](https://www.researchgate.net/publication/277951362_Optimal_Design_of_PID_Controller_f_or_the_Speed_Control_of_DC_Motor_by_Using_Metaheuristic_Techniques). [Accessed: 10.05.2021.].

## Acknowledgement

The publication was supported by the project EFOP-3.6.1-16-2016-00022. This project was co-funded by the European Union, from the European Social Fund.