

**Fault analysis using the bond graph  
approach for a PI-controlled DC motor  
velocity signal generated by an FPGA-  
based design for a small robotic  
prosthesis**



In the present paper, we perform a fault diagnosis by bond graph (BG) approach for a DC motor used in small robotic prostheses. It consists of diagnosing and detecting the motor faults using residuals thanks to the BG modeling of the system element by element. The 3 V motor supply voltage is generated by a Xilinx Spartan 3A/AN field programmable gate array (FPGA) board. Indeed, this latter has a 3.3 V reference voltage that presents the maximum value to be generated. So beyond this value, we are not going to detect a fault that can cause an increase in the supply voltage. Thus, a fault analysis by BG is needed to detect such failure. On the other side, FPGAs are used to design an autonomous prosthesis characterized by low energy consumption and rapid response. For this purpose, a hardware design is also performed using the HDL (Hardware Description Language) Coder Tool of Matlab and implemented on the FPGA target. It includes the DC motor and a Proportional Integrator (PI) controller that determines the velocity control signal, which is generated then by the FPGA through an integrated digital-to-analog converter. The simulation results show the effectiveness of the proposed method for detecting and localizing faults. They include a friction torque rise and a low/high supply voltage. Furthermore, the simulation and implementation results show that the hardware model allows reaching the desired velocity and generating the corresponding pulse width modulation (PWM) signal and control one that supplies the motor.

Keywords: FPGA, DC motor, Speed control, Bond graph, Fault analysis

## 1. Introduction

Nowadays, supervision and diagnosis of automated systems are essential tasks in the industrial processes to guarantee the required production in safe conditions. The improvement of the security of operating systems remains crucial in the algorithms of Fault Detection and Diagnosis (FDD) [1]. To detect faults, they compare the system behavior with a reference reflecting the normal operation. They also deal with various kinds of faults to analyze and diagnose failures. In particular, robotic prostheses facilitating human movement have become widely used and necessary to help people to overcome a movement handicap. Hence, they allow improving the life quality of these people by recovering the lost mobility and assisting them to achieve their daily activities. Consequently, they should be effective, and each malfunction should be fixed, as well as the element which causes it. Thus, fault diagnosis is crucial, notably for robotic prostheses equipped with DC motors, which ensure the joints' operation and may undergo failures [3]. Therefore, they should be permanently diagnosed to early detect malfunctions that can occur. Moreover, the DC motor speed control is a deal-breaker in such systems. On the other hand, robotic prostheses usually require a quick response time to reach the desired movement. Consequently, their design needs specific materials and flexible and rapid

\* Corresponding author: E. Aridhi, Department of Physics, Université de Tunis El Manar, Campus Universitaire Farhat Hached, Tunis, Tunisie, E-mail: emna.aridhi@fst.utm.tn

<sup>1</sup> Laboratoire d'Application de l'Efficacité Énergétique et des Énergies Renouvelables-LAPER, Faculté des Sciences de Tunis, Université de Tunis El Manar, Campus Universitaire Farhat Hached, Tunis, Tunisie

hardware/software solutions to achieve lower consumption, high efficiency, the desired comfort, complete autonomy, and a longer lifetime [4].

Reconfigurable architectures such as FPGAs and Systems on Chip (SoC) are increasingly used for their high flexibility. This latter characterizes the ability of functional blocks to be reconfigured statically or dynamically. As a result, these hardware architectures are rapid and device-rich platforms. Thus, the latency is reduced between logic blocks thanks to their flexible architecture and support of parallel processing of signals compared to classical microcontrollers. Consequently, they are considered suitable for developing a wide variety of applications requiring a lot of computational and processing time. On the other hand, the hardware platforms allow rapid prototyping of the envisaged application and ensure its autonomy. Several research studies are interested in modeling and controlling robotic prostheses using various embedded solutions. For example, in [5], the authors used a DC motor to manipulate a knee joint. They also developed an embedded torque control system for the corresponding prosthesis using a 32-bit Digital Signal Processing (DSP) controller. It constitutes the computation unit. The prototype was validated by some walking experiments. In [6], the authors developed a Proportional Integral Derivative (PID) controller to support an active ankle-foot orthosis equipped with a brushed DC motor. They implemented it on a LabVIEW FPGA Module, which reads the sensor values and generates the PWM signals to run the DC motor. Other works focused on implementing speed controllers applied to DC motors on a Xilinx FPGA circuit [7,8]. For instance, the controller mainly includes a debouncing circuit and a PWM generator [7].

Furthermore, the diagnosis and the localization of faults are arduous processes because the system is taken from an overall view or divided into main parts [9,10]. In this case, we can not immediately decide what the component causing a given failure is, and searching for it will take a lot of time. From here comes the pertinence of the BG approach. It is a system modeling element by element. Indeed, the system parameters are usually presented by elements (resistive R, capacitance C, inertial I elements, etc.). Thus, the fault will be detected easily in the envisaged BG element [11]. In addition, these systems are governed by several physical phenomena. The BG approach is well-suitable for modeling them, given that it is an energy and multi-physical analysis [12,13, 14]. By its causal structure, browsing the causal paths on the BG model allows for generating the residuals using the energy conservation equations stemming from 0 and 1 junctions [15]. Then, the process faults (leak, blocking, abnormal rise or drop of a data value, etc.) can be systematically allocated to the residuals sensitive to them. This property also facilitates the construction of fault signature tables that show the dependence between residuals and the BG elements. Hence, the faults will be localized [16,17]. For instance, in [18], the authors developed a robust fault analysis of a DC motor based on Linear Fractional Transformations (BG-LFT) at the level of its induced current and speed. In [19], the authors developed a Luenberger observer design using the BG approach. They performed a residual sensitivity analysis to detect the faults of a DC motor powered by 47.3 V. These faults closely influence the motor velocity. They consist of only a voltage decrease of 20 V.

Within this framework, we propose a BG model that analyzes and detects faults in a 3 V DC motor suitable for small robotic prostheses. The faults consist of an abrupt variation of the motor supply voltage and a resistive torque. In normal operation, we must obtain a stable voltage that does not exceed 3V. It is delivered by a Xilinx Spartan 3A/AN FPGA board. Otherwise, we can detect a given fault and the element that causes it through the proposed BG analysis model and thus its influence on the DC motor velocity. The FPGA technology allows an autonomous operation with low power consumption and rapid response.

The paper is structured as follows: Section 2 explains the diagnosis of systems using the BG approach. Section 3 gives the model of the PI-controlled DC motor to reach the desired velocity. Section 4 describes the hardware design of the complete system. Section 5 depicts the analysis BG model of the proposed system. The simulation and implementation results are reported in section 6. The last section is devoted to concluding the paper.

## 2. Diagnosis of systems using Bond Graphs

### 2.1. Construction of an industrial system using a BG model

An industrial system can be modeled by bond graphs, as shown in Figure 1. The input signals are modeled by effort ( $Se$ ) or/and flow ( $Sf$ ) sources. However, the system dynamics are modeled by resistive ( $R$ ) and storage ( $I$  or  $C$ ) elements. Moreover, the detectors are modeled by detector elements ( $De$  or  $Df$ ).

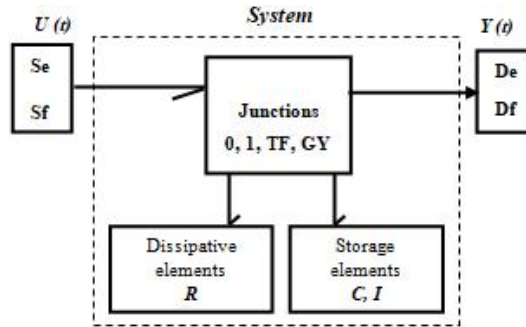


Figure 1. Industrial system described by the BG approach

The BG approach models a system element by element by representing the system elements as BG components, defining their causality and constructive relations, and connecting them to form a complete BG model. Then, the system's behavior will be simulated and analyzed. This graphical approach provides a powerful tool for modeling and analyzing complex systems systematically and visually.

### 2.2. Construction of faults using a bond graph model

Any fault can affect any component of the industrial system and can also be modeled by the BG model [20-23], in particular, affecting the following components:

- The sources ( $Se$ ,  $Sf$ ).
- The components ( $R$ ,  $C$ , and  $I$ ).

#### 2.2.1. Fault model affecting the effort and flow sources ( $Se$ , $Sf$ )

If a fault affects an effort source ( $Se$ ) or a flow source ( $Sf$ ), then the actual outputs from these sources will be expressed in (1) and (2), respectively.

$$\mathcal{S}e: e = e_n + F_a \quad (1)$$

with  $e_n$  is the nominal value and  $F_a$  is the fault value on the effort source ( $Se$ ).

$$Sf: f = f_n + F_b \tag{2}$$

where,  $f_n$  is the nominal value and  $F_b$  is the fault value on the flow source ( $Sf$ ).

The faults  $F_a$  and  $F_b$  are respectively modeled by a modulated effort source (MSe:  $F_a$ ) and a modulated flow source (MSf:  $F_b$ ), as shown in Figure 2.

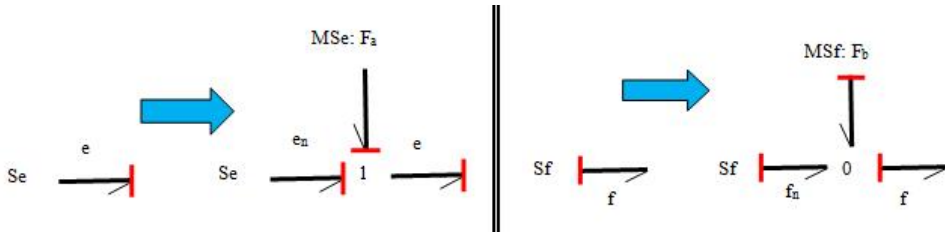


Figure 2. Fault BG model affecting the effort and the flow sources

### 2.2.2. Fault model affecting the components (R, C and I)

#### a) R element

The characteristic law of an  $R$  element in integral and derivative causalities is given by (3) and (4), respectively.

$$e = R \cdot f \tag{3}$$

$$f = \frac{1}{R} \cdot e \tag{4}$$

We consider the presence of a multiplicative fault of a relative value  $F_R$  due to a variation in parameter  $R$  as given by (5) and (6).

- Integral causality:

$$R = R_n(1 + \beta_R) = R_n + F_{FR} \tag{5}$$

where  $R_n$  is the nominal effort, and  $F_{FR} = R_n \beta_R$  is the fault value on the  $R$  element in integral causality.

The part that represents the fault  $F_{FR} = (R_n \beta_R) f$  is modeled by a modulated effort source MSe ( $MSe: F_{FR} f = (R_n \beta_R) f$ ) and illustrated in Figure 3.

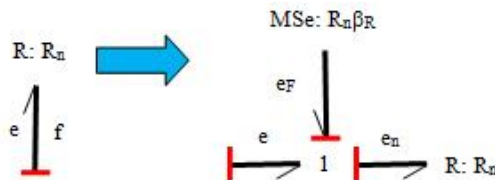


Figure 3. Fault BG model affecting the R element in integral causality

- Derivative causality:

$$\frac{1}{R} = \frac{1}{R_n(1+\beta_R)} = \frac{1}{R_n}(1-\beta_R) \mathcal{F} = \frac{1}{R_n} + F_R \quad (6)$$

where  $(1/R_n)$  is the nominal value, and  $F_R = (1/R_n)(-\beta_R)$  is the fault value on the R element in derivative causality.

The part representing the fault  $F_R e_F = (1/R_n)(-\beta_R) e_F$  is modeled by a modulated flow source *MSf* (*MSf*:  $F_R e_F = (1/R_n)(-\beta_R) e_F$ ) and shown in Figure 4.

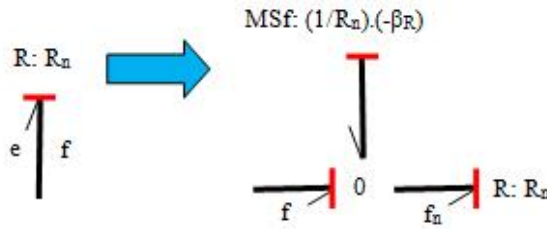


Figure 4. Fault BG model affecting the R element in derivative causality

#### b) C element

The characteristic law of a C element in integral and derivative causalities is given by (7) and (8), respectively.

$$e = \frac{1}{C} \int f dt \quad (7)$$

$$f = C \frac{de}{dt} \quad (8)$$

We consider the presence of a multiplicative fault of a relative value  $F_C$  due to a variation in the parameter C according to (9) and (10).

- Integral causality:

$$\frac{1}{C} = \frac{1}{C_n(1+\beta_C)} = \frac{1}{C_n}(1-\beta_C) \mathcal{F} = \frac{1}{C_n} + F_C \quad (9)$$

with,  $(1/C_n)$  is the nominal value, and  $F_C = (-\beta_C/C_n)$  is the fault value on the C element in integral causality.

The part representing the fault is modeled by a modulated effort source *MSe* and shown in Figure 5.

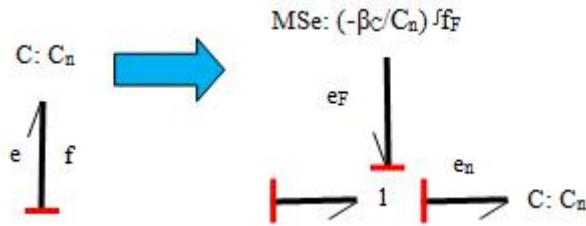


Figure 5. Fault BG model affecting the C element in integral causality

- Derivative causality:

$$C = C_n(1 + \beta_c) = C_n + F_c \quad (10)$$

where  $C_n$  is the nominal value, and  $F_c = C_n \beta_c$  is the fault value on the  $C$  element in derivative causality.

The part representing the fault  $F_c (de_f/dt) = (C_n \beta_c) (de_f/dt)$  is modeled by a modulated flow source  $MSf (MSf: F_c (de_f/dt) = (C_n \beta_c) (de_f/dt))$  and described in Figure 6.

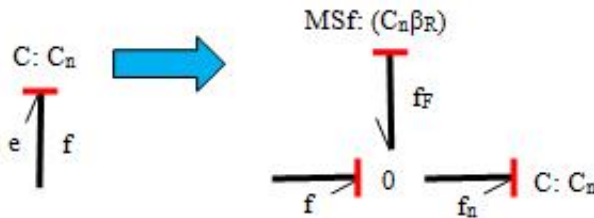


Figure 6. Fault BG model affecting the C element in derivative causality

### C) I element

The characteristic law of an I element in integral and derivative causalities is given by (11) and (12), respectively.

$$f = \frac{1}{I} \int e dt \quad (11)$$

$$e = I \frac{df}{dt} \quad (12)$$

We consider the presence of a multiplicative fault of a relative value  $F_I$  due to a variation in the parameter  $I$ , as shown in (13) and (14).

- Integral causality:

$$\frac{1}{I} = \frac{1}{I_n(1 + \beta_I)} = \frac{1}{I_n} (1 - \beta_I) = \frac{1}{I_n} + F_I \quad (13)$$

where  $(1/I_n)$  is the nominal value and  $F_I = (-\beta_I/I_n)$  is the fault value on the  $I$  element in integral causality.

The part representing the fault is modeled by a modulated flow source MSf, as shown in Figure 7.

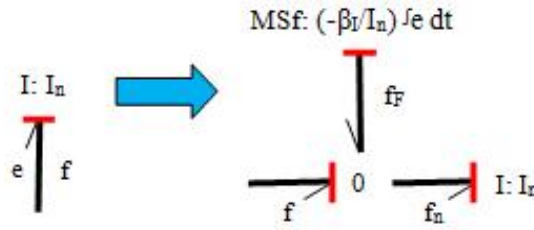


Figure 7. Fault BG model affecting the I element in integral causality

- Derivative causality:

$$I = I_n(1 + \beta_I) = I_n + F_I \quad (14)$$

where  $I_n$  is the nominal value, and  $F_I = I_n \beta_I$  is the fault value on the  $I$  element in derivative causality.

The part representing the fault  $F_I (df_F/dt) = (I_n \beta_I) (df_F/dt)$  is modeled by a modulated effort source MSe (MSe:  $F_I (df_F/dt) = (I_n \beta_I) (df_F/dt)$ ) and shown in Figure 8.

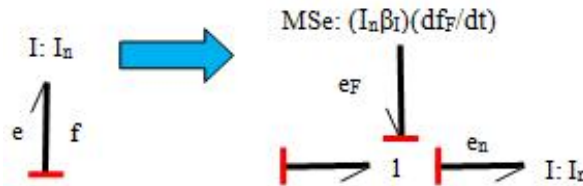


Figure 8. Fault BG model affecting the I element in derivative causality.

### 3. Modeling of the DC motor and the PI controller

In this section, we depicted the modeling of the system to diagnose. It consists of a DC motor controlled by a PI controller. This latter generates the supply voltage that drives the motor velocity through an integrated digital-to-analog converter (DAC) in the FPGA board. It is worth noting that the supply voltage is around 3V. For this purpose, we have first determined and tested the complete model using the Matlab/Simulink environment. Afterwards, we used the HDL Coder tool to get the VHDL code to implement on FPGA.

#### 3.1. Model of the DC motor

The DC motor (rotor) consists of a series connection of an inductance  $L$ , a resistor  $R$ , and a back electromotive force voltage  $e$ . Then, we modeled it using (15), the electromagnetism equations (16) and (17), as well as equation (18) of the motor shaft dynamics [10].

$$U_{motor} = e(t) + Ri(t) + L \frac{di(t)}{dt} \tag{15}$$

$$e(t) = k_e \cdot \omega_{motor}(t) \tag{16}$$

$$C_{motor} = k_c \cdot i(t) \tag{17}$$

$$C_{motor} - C_{res} - f \cdot \omega_{motor}(t) = J \frac{d\omega_{motor}(t)}{dt} \tag{18}$$

where  $U_{motor}$  is the supply voltage applied to the motor [V],  $\omega_{motor}$  is the angular velocity of the motor [rad/s],  $C_{motor}$  and  $C_{res}$  are the motor and resistive torques [N.m], respectively. Besides, the parameters  $k_e$  [V.s/rad] and  $k_c$  [N.m/A] present the coefficient of the back electromotive force voltage  $e$  and the torque constant, respectively. These two parameters are equivalent and have the same value.  $f$  is the motor friction [N.m.s/rad], and  $J$  is the inertia moment of the motor [kg.m<sup>2</sup>]. The model parameters are given in Table 1 [10].

Table 1. Model parameters

Parameters	Values
$k_e$	0.037 V.s/rad
$k_c$	0.037 N.m/A
$f$	0.000015 N.m.s/rad
$J$	2.4e-6 kg.m <sup>2</sup>
$U$	3 V
$I$	0.4 A
$R$	4.5 Ω
$L$	1.6 mH
$C_{motor}$	0.37 N.m

We consider that the initial conditions are equal to zero for the sake of simplification to get the hardware design. The transfer function of the DC motor, which computes its angular velocity  $\omega_{motor}$  according to the supply voltage, is expressed in (19).

$$\omega_{motor}(p) = \frac{\frac{k_c}{k_c \cdot k_e + f \cdot R}}{\frac{L \cdot J}{k_c \cdot k_e + f \cdot R} p^2 + \frac{R(J + f)}{k_c \cdot k_e + f \cdot R} p + 1} U_{motor}(p) - \frac{\frac{R + L \cdot p}{k_c \cdot k_e + f \cdot R}}{\frac{L \cdot J}{k_c \cdot k_e + f \cdot R} p^2 + \frac{R(J + f)}{k_c \cdot k_e + f \cdot R} p + 1} C_{res}(p) \tag{19}$$

In our case, the transfer function obtained is expressed in (20).



$$\omega_{motor}(p) = \frac{4.894}{1.225e-4p^2 + 0.0246p + 1} U_{motor}(p) - \frac{24.4499 + 0.1222p}{1.225e-4p^2 + 0.0246p + 1} C_{res}(p) \quad (20)$$

The continuous transfer function is then discretized at the sampling time of 0.01 s and takes the form of (21).

$$\omega_{motor}(z) = \frac{1.068z + 0.5476}{z^2 + 0.8042z + 0.1342} U_{motor}(z) - \frac{9.107z - 1.037}{z^2 + 0.8042z + 0.1342} C_{res}(z) \quad (21)$$

From the previous discrete transfer function and applying the inverse z-transform, the motor angular velocity is expressed according to the supply voltage, as given in (22).

$$\omega_{motor}(k) = 4.866 u_{motor}(k-1) + 0.11 u_{motor}(k-2) + 0.00348 \omega_{motor}(k-1) \quad (22)$$

### 3.2. Pi controller

The equation that drives the PI controller is given by (23). The control signal  $u$  acts on the motor supply voltage.

$$u(k) = u(k-1) + 0.0051 err(k) + 0.0051 err(k-1) \quad (23)$$

The  $K_p$  and  $K_i$  parameters are obtained using the PI tuner tool. They are respectively equal to 0.0051 and 1.028.

### 3.3. Model of the complete system

To use the HDL Coder tool, we built the Simulink model of the system using the recursive equations of the motor and the PI controller. It is illustrated in Figure 9.

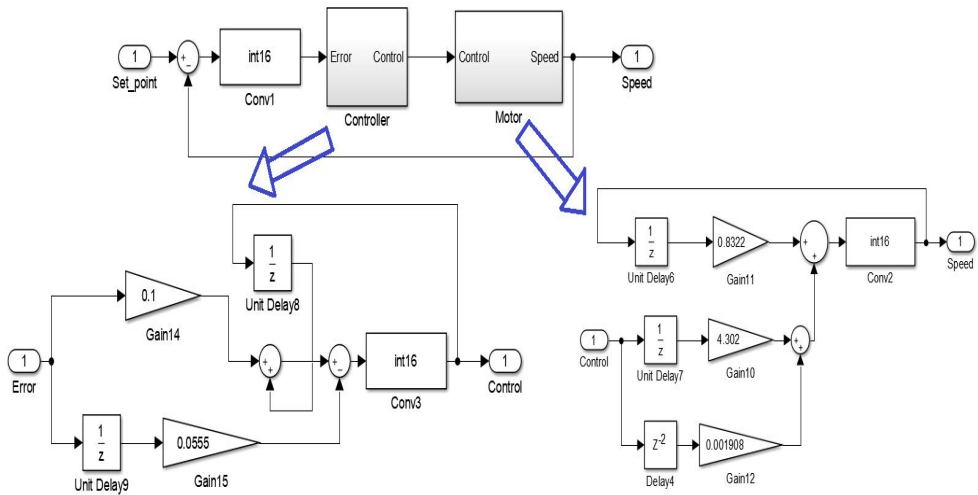


Figure 9. Complete model of the motor velocity control

The ‘Convert’ blocks allow converting the data type from double to integer 16 bits to get a synthesizable hardware code. Thus, we can implement it into the FPGA.

#### 4. Hardware design of the PI controller associated with the DC motor

In this section, we developed a synthesizable hardware design of the complete model of the motor and the PI controller, as shown in Figure 10.

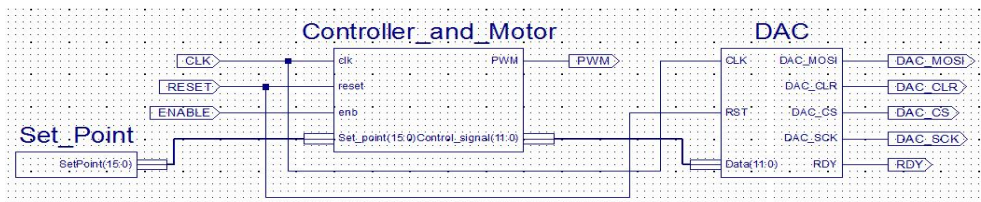


Figure 10. Hardware design of the motor and the PI controller

The model was obtained using the HDL Coder of Matlab and imported to the ISE environment. Even if the reset signal is inhibited, the system can be disabled when the ENABLE signal is low. This latter is not needed for the DAC. The digital control signal is converted into a voltage value  $V$  using (24).

$$V = \frac{data[11:0]}{4096} V_{ref} \tag{24}$$

where,  $V_{ref}$  is the supply voltage of the FPGA board and is equal to 3.3 V.

It is worth noting that  $V_{ref}$  corresponds to the maximum supply voltage of the motor. Besides, the DAC resolution is 12 bits. For this reason, the control signal has a width of 12 bits. However, the speed and the setpoint are 16-bit signals.

The digital data is transmitted to the DAC bit per bit (signal MOSI) using the SPI interface during 32 cycles according to the clock frequency (SCK) of 25 Mhz. The FPGA controls the DAC through these SPI signals, also including the CLR and CS signals. The CLR

signal must be high during the conversion. It allows activating the DAC peripheral. The CS signal selects it as a slave and must be low during the conversion. This latter is described by a finite state machine presented in Figure 11.

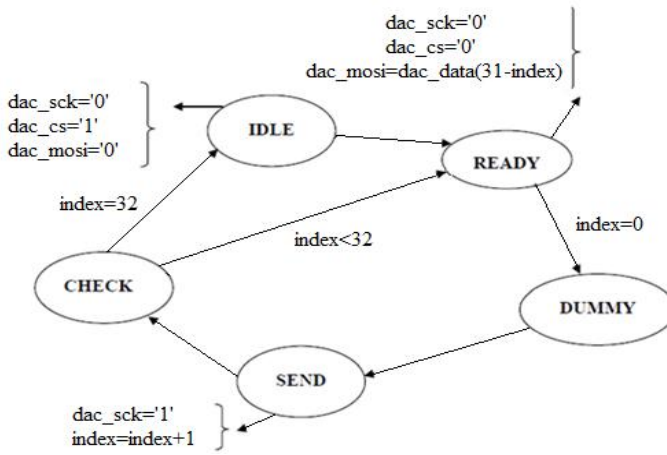


Figure 11. Finite state machine of the DAC block

## 5. Fault Diagnosis of the DC motor

The diagnosis is performed to detect the malfunctioning operation of the system and identify the element that causes it. Indeed, the maximum analog value delivered by the DAC is equal to the reference voltage  $V_{ref}$  of the FPGA board. Thus when the supply voltage becomes higher than  $V_{ref}$  due to a fault, we can not detect this malfunction. For this purpose, we handled defective cases that disturbed the motor velocity. In the first case, we applied a low and high supply voltage without damaging the motor. This fault describes the case when the controller does not operate properly. Thus, the motor velocity does not reach the required value. In The second case, we increased the motor friction. This fault reflects the motor overload.

To diagnose the system using the BG approach, we must write the residuals' equations at the level of 1-junctions where a flow sensor ( $D_f$ ) is added at each 1-junction, as shown in Figure 12. We assume that the motor torque is constant.

The residuals are used as fault indicators. Indeed, they should indicate a fault that has happened. Then, the previous BG model becomes in normal operation, as shown in Figure 13.

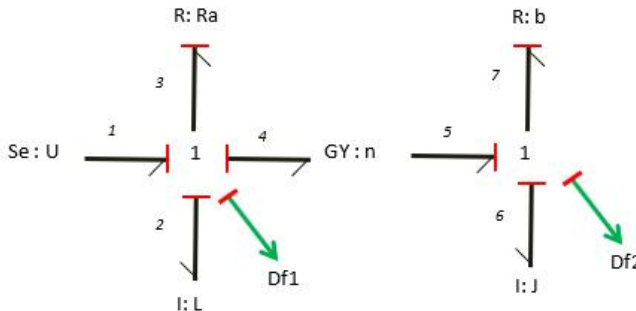


Figure 12. Bond graph model of the DC motor

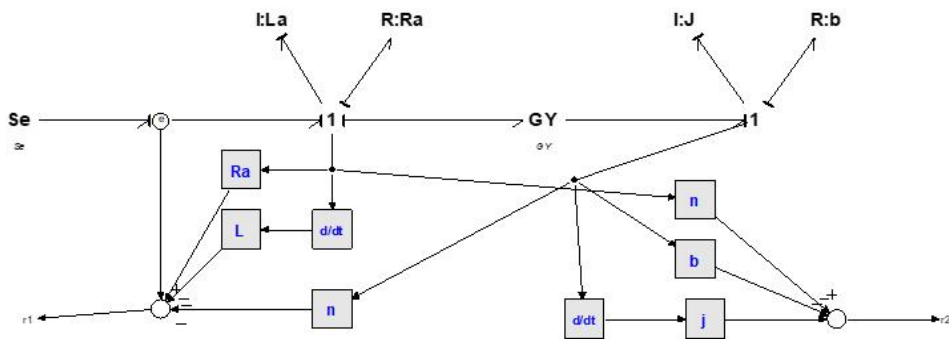


Figure 13. BG model of the DC motor in normal operation

From Figures 12 and 13, the residuals are determined at the level of each 1-junction, as indicated by (25).

$$r = \sum \text{effortsIn} - \sum \text{effortsOut} = 0 \tag{25}$$

The residual  $r_1$  is computed at the level of the first 1-junction according to (26).

$$r_1 = e_1 - e_2 - e_3 - e_4 = U - L \frac{dDf1}{dt} - Ra \cdot Df1 - n \cdot Df2 \tag{26}$$

The residual  $r_2$  is computed at the level of the second 1-junction according to (27).

$$r_2 = e_5 - e_6 - e_7 = n \cdot Df1 - J \frac{dDf2}{dt} - b \cdot Df2 \tag{27}$$

From (26) and (27), we found that the residual  $r_1$  is sensitive to the BG elements R:Ra and I:L. Therefore, they allow its localization. Likewise, the residual  $r_2$  is sensitive to the I elements I:J (Inertia) and R:b (resistive torque). Table 2 summarizes the dependence between the residuals and all the BG model elements.

Table 2. Dependence between the residuals and the BG elements.

	r1	r2
Ra = 4.5 V	1	1
L = 1.6 mH	1	0
b = 1.5e-5 Nm	0	1
J = 2.4e-6 kg.m <sup>2</sup>	0	1
n = 0.37 Nm	1	1
Mse:U = 3 V	1	0
Df1	1	1
Df2	1	1

## 6. Results and discussion

### 6.1. Simulation results of the DC motor model

The simulation was carried out for 3 s, for which the target angular velocity changed from 50 to 76 rpm. In this case, the motor reached the target velocity after about 1 s for a supply voltage of 2 to 3 V generated by the PI controller, as shown in Figure 14.

## 6.2. Simulation results of the fault diagnosis model

The simulation of the pseudo-BG model presented in Figure 13 was carried out during 0.1 s. In normal operation, there are no faults. Consequently, the residuals  $r_1$  and  $r_2$  should reach zero, as shown in Figure 15.

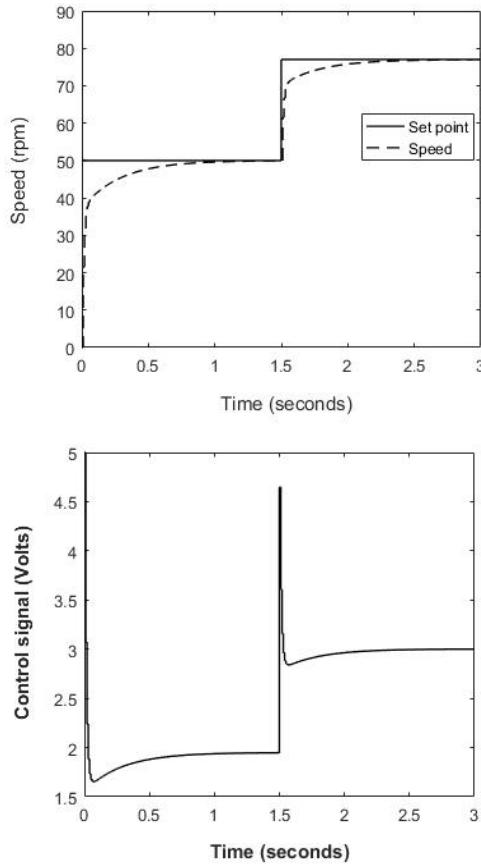


Figure 14. Variation of the motor velocity according to the control signal generated by the PI controller

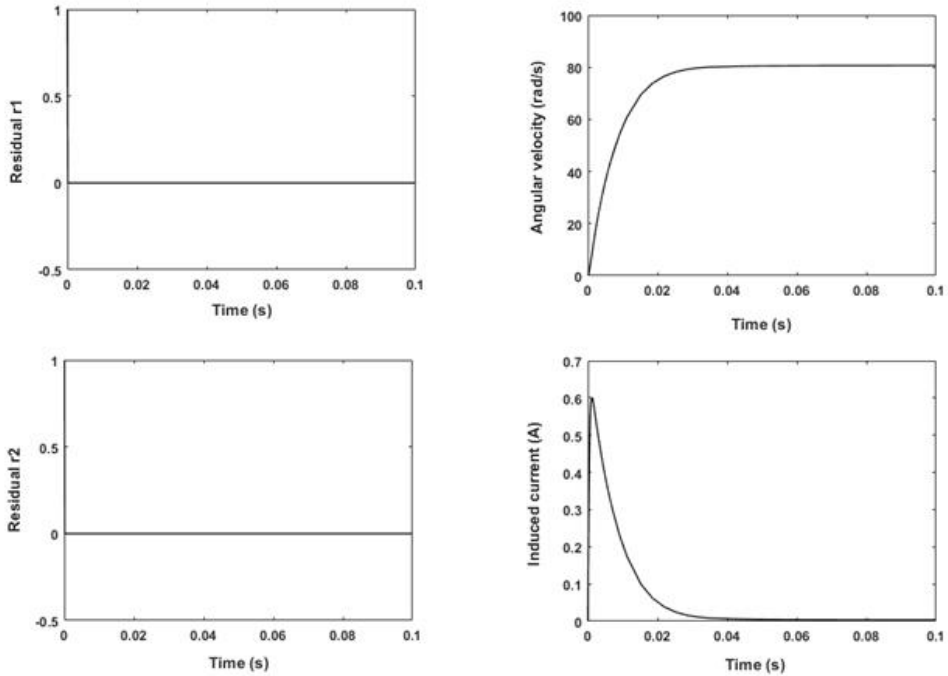


Figure 15. Profile of the residuals and behavior of the angular velocity and the induced current when the supply voltage is 3 V (normal operation)

In this case, the angular velocity reaches about 77 rad/s (735 rpm) when a supply voltage of 3 V is applied. This voltage is generated by the DAC to power the DC motor.

The two faults, applied to the DC motor and influenced the residuals, consist of a low/high supply voltage (1.5 and 4 V) and a friction torque rise. It makes the motor overloaded. Indeed, Figure 16 shows the influence of faults on the profile of residuals when the supply voltage rises from 3 to 4 V between 0.2 and 0.4 s, and the friction torque  $b$  also rises from 0.016 to 0.12 mNm between 0.5 and 0.6 s (high supply voltage and friction couple rise). It is worth noting that when the supply voltage exceeds 3.3 V, we cannot persuade the liable fault without the BG fault analysis, seeing that this value presents the maximum one that the DAC can generate. Hence, it is necessary to carry out a fault analysis to detect this malfunction.

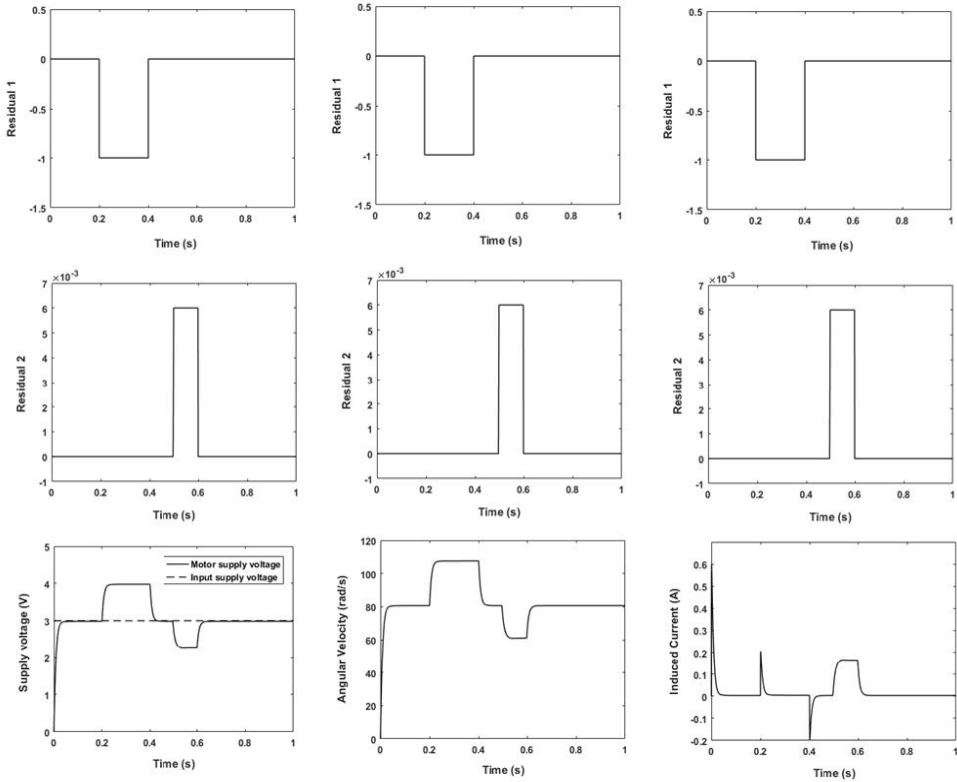


Figure 16. Profile of residuals and variation of the motor supply voltage, the induced current, and the angular velocity when the supply voltage and the friction torque increased to 4 V and 0.12 mNm, respectively, are applied to the system

We note that the rise of the motor supply voltage caused an increase of its velocity to about 110 rad/s. Consequently, the desired velocity (about 77 rad/s) can no longer be reached. However, when the residual r1 becomes zero, the motor normal operation resumes. After that, the second fault occurred, and thus the velocity decreased to about 60 rad/s, seeing that the motor became overloaded. Besides, its supply voltage dropped from 3 to about 2.5 V.

For the same friction torque, we decreased the supply voltage from 3 to 1.5 V. The obtained results are plotted in Figure 17. In this case, the angular velocity dropped to 40 rad/s, seeing that the motor supply voltage becomes insufficient to ensure its normal operation. The induced current is zero and causes a pic of  $\pm 0.3$  A when the supply voltage increases and decreases, respectively.

We conclude then that we can detect any malfunction of the DC motor and the element which causes it using the BG approach and through the profile of the residuals.

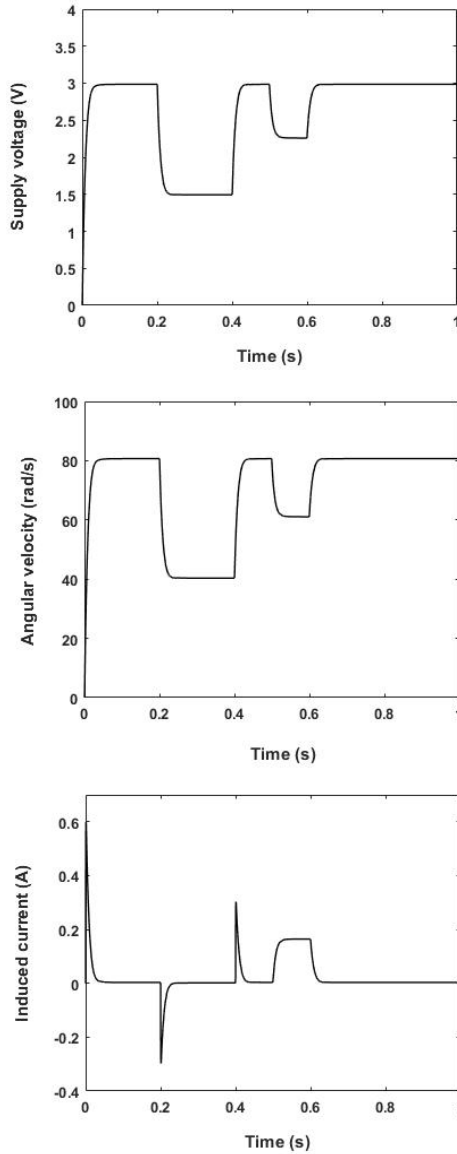


Figure 17. Variation of the motor supply voltage, the induced current, and the angular velocity when the supply voltage decreases to 1.5 V

### 6.3. Simulation results of the hardware design

The hardware design was developed using the Xilinx ISE software and implemented on an FPGA Spartan 3A/3AN board having a clock frequency equal to 50 MHz.

The design simulation was carried out using the ISIM tool for 5000 ns, as shown in Figure 18. The angular velocity of the motor reached 76 rad/s for a target velocity equal to 80 rad/s after about 240 ns. Therefore, the DC motor response is highly rapid. It only requires 12 clock cycles. On the other hand, the digital value of the corresponding control signal is established at 3720. Its analog value obtained by the DAC corresponds to 3 V. The



PWM signal is high as long as the control signal is non-zero, which keeps running the DC motor.

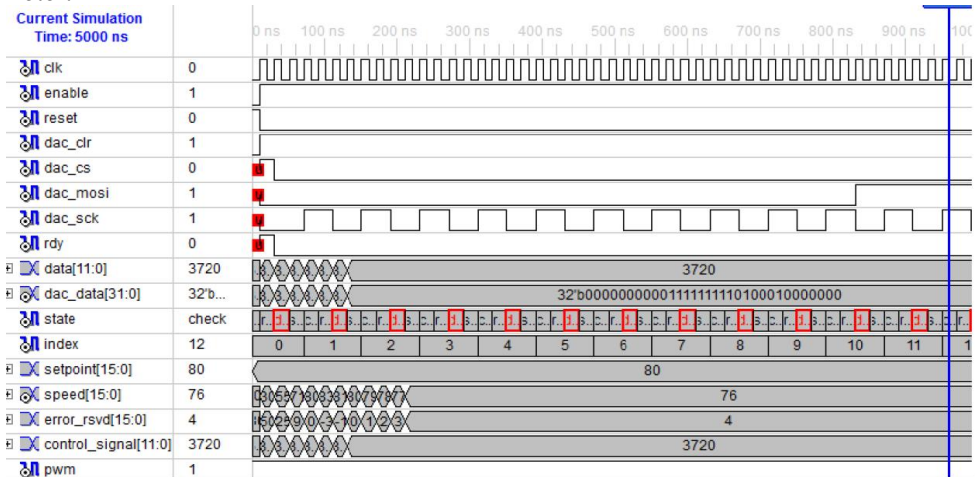


Figure 18. Simulation results of the hardware design

#### 6.4. Implementation results of the hardware design

The design implementation on the Xilinx Spartan 3A Starter kit board requires the execution of several processes. The first one is the synthesis process which generates the RTL pattern of the design.

After the synthesis, we executed the processes of translate, map, and route after fixing the localization of each port in the design. The hardware resources used are summarized in Table 3.

Table 3. Hardware resources used by the design

Logic Utilization	Used	Available	Utilization (%)
Number of Slices	99	5888	1
Number of Slice Flip Flops	76	11776	1
Number of 4 input LUTs	171	11776	1
Number of bonded IOBs	9	372	2
Number of MULT18x18 IOs	5	20	25
Number of GCLKs	1	24	4

Afterwards, the FPGA circuit is configured by the bitstream using the IMPACT tool. The implementation results are shown in Figure 19. The oscilloscope shows the converted average value of 2.98 V and its maximum value of 3V. The PWM signal status is highlighted by a led that is set high in this case.

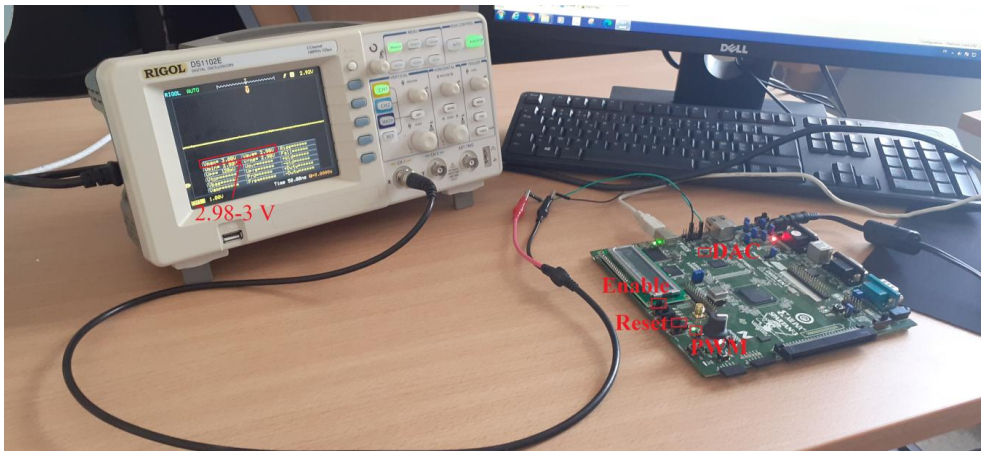


Figure 19. Implementation results of the control signal generation by the FPGA Spartan 3A/AN board

## 7. Conclusion

In the present paper, we developed a bond graph model of a 3 V DC motor used in small robotic prostheses. It allows diagnosing and detection its faults by identifying the elements that cause them using residuals. Despite the use of the DAC to deliver the motor supply voltage that does not exceed the reference voltage  $V_{ref}$  of the FPGA board (3.3 V), a fault may unknowingly cause an abnormal increase in this value. From here comes the importance of the bond graph analysis to detect this malfunction. On the other hand, we noted that FPGAs are suitable for designing small autonomous prostheses based on DC motors. Indeed, these reconfigurable circuits are characterized by high-density logic gates, low energy consumption, and rapid response. For this purpose, we proposed a hardware design developed using the HDL Coder Tool of Matlab. The simulation of the BG model was performed for two kinds of faults: an increase and decrease of both the supply voltage and the friction torque of the DC motor. The results show that the model allows the detection of the fault and the element that causes it. Likewise, the hardware model allows for reaching the desired velocity and generating the corresponding control signal that supplies the DC motor.

In future work, we aim to develop an observer of the system.

## References

- [1] W. Borutzky, *Bond Graph Modelling for Control, Fault Diagnosis and Failure Prognosis*, Springer Nature Switzerland AG, 1-325, 2021.
- [2] X. Wang, Q. Meng, Z. Zhang, J. Sun, J. Yang & H. Yu, Design and evaluation of a hybrid passive–active knee prosthesis on energy consumption, *Mechanical Sciences*, 11, 425–436, 2020.
- [3] M. I. Awad, A. A. Dehghani-Saniij, D. Moser & S. Zahedi, Motor electrical damping for back-drivable prosthetic knee, *In Proc. 11th France-Japan & 9th Europe-Asia Congress on Mechatronics (MECATRONICS) /17th International Conference on Research and Education in Mechatronics (REM)*, 348-353, France, 2016.
- [4] R. R. Torrealba, E. D. Fonseca-Rojas, Toward the Development of Knee Prostheses: Review of Current Active Devices, *Applied Mechanics Reviews*, 71(3), 1-22, 2019.
- [5] W-S. Kim, S-Y. Kim & Y-S. Lee, Development of a Powered Knee Prosthesis using a DC Motor, *Journal of Institute of Control, Robotics and Systems*, 20(2), 193-199, 2014.
- [6] B. Jo, Y-S. Lee, Design and Control of an Active Powered Knee Prosthesis System using Position-based Impedance Control, *In Proc. of the 3rd World Congress on Mechanical, Chemical, and Material Engineering (MCM'17)*, 1-8, Italy, 2017.

- [7] A. Telba, Motor Speed Control Using FPGA, *In Proc. of the World Congress on Engineering (WCE 2014)*, 1-4, U.K, 2014.
- [8] E. Julie Hepzibah, R. Korah, FPGA Implementation for Speed Monitoring and Speed Control of a DC Motor using Fuzzy Logic, *International Conference on Emerging Trends in Electrical Engineering and Energy Management (ICETEEEM-2012)*, 222-228, 2012.
- [9] A. Sellami, D. Mzoughi & A. Mami, Efficiency of the Bond Graph Approach for Robust Diagnosis Micro Electro Mechanic System (MEMS), *Journal of Circuits, Systems and Computers*, 30(5), 1-21, 2020.
- [10] A. Sallami, N. Zanzouri & M. Ksouri, Robust Fault Diagnosis Observer of Dynamical Systems Modelled by Bond Graph Approach, *IFAC Proceedings*, 45, 415-420, 2012.
- [11] C. B. Low, D. Wang, S. Arogeti, & M. Luo, Quantitative Hybrid Bond Graph-Based Fault Detection and Isolation, *IEEE Transactions on Automation Science and Engineering*, 7(3), 558–569, 2010.
- [12] A. Mahmoudi, E. Roshandel, S. Kahourzade, F. Vakilipoor & S. Drake, Bond graph model of line-start permanent-magnet synchronous motors, *Electrical Engineering*, 2022.
- [13] D. C. Karnopp, D. Margolis & R. Rosenberg, *Systems Dynamics: A Unified Approach*, 2nd edition. (John Wiley, New York, 1990).
- [14] M. Vergé, D. Jaume, Fault detection and bond graph modeling, *IFAC Symp. on Fault Detection, Supervision and Safety for Technical Processes*, 748–753, Finland, 2004.
- [15] B. Ould Bouamama, A. K. Samantaray, M. Staroswiecki, & G. Dauphin-Tanguy, Derivation of constraint relations from bond graph models for fault detection and isolation, *Proc. ICBGM'03, Simul. Series*, 104–109, Arizona, 2003.
- [16] C. H. Lo, Y. K. Wong & A. B. Rad, Model-based fault diagnosis in continuous dynamic systems, *ISA Transactions*, 43(3), 459–475, 2004.
- [17] A. K. Samantaray, S. K. Ghoshal, Bicausal bond graphs for supervision: From fault detection and isolation to fault accommodation, *Journal of Franklin Institute*, 345(1), 1–28, 2008.
- [18] W. Borutzky, Residual sinks in bond graph based numerical fault detection, *Proc. MS'08 of the 19th IASTED International Conference on Modelling and Simulation*, 163–171, Canada, 2008.
- [19] A. K. Samantaray, K. Medjaher, B. Ould Bouamama, M. Staroswiecki & G. Dauphin-Tanguy, Diagnostic bond graphs for online fault detection and isolation, *Simulation and Modelling Practice Theory*, 14(3), 237–262, 2006.
- [20] N. Chatti, B. Ould-Bouamama, A-L. Gehin & R. Merzouki, Signed bond graph for multiple faults diagnosis, *Engineering Applications of Artificial Intelligence*, 36, 134-147, 2014.
- [21] S. Benmoussa, R. Merzouki & B. Ould Bouamama, Plant Fault Diagnosis using Bond Graph Approach: Application to Intelligent Autonomous Vehicle, *IFAC Proceedings*, 45, 952-957, 2012.
- [22] A. Sallami, N. Zanzouri & M. Ksouri, Robust Diagnosis of a DC Motor by Bond Graph Approach, *American Journal of Engineering and Applied Sciences*, 9(2), 432-438, 2016.
- [23] G. Saoudi, R. El Harabi & M. N. Abdelkrim, Graphical Linear Observers for Fault Detection: the DC motor case study, *2013 10th International Multi-Conference on Systems, Signals & Devices (SSD)*, 1-8, Tunisia, 2013.