Modelica Model for the youBot Manipulator

Rhama Dwiputra Alexey Zakharov Roustiam Chakirov Erwin Prassler
Bonn-Rhein-Sieg University of Applied Sciences, Department of Computer Science
Grantham-Allee 20, 53757 Sankt Augustin

Abstract

In robotics, a robot model plays an important role in education and research. This paper presents the development of a Modelica model for the youBot manipulator. Whereas other robotics tools focus on the robot interaction with its environment, Modelica multi domain capability allows the modeling of the manipulator controllers and motors. The model was developed with a new Modelica library for the manipulator components to provide modularity, reusability and abstraction. To ensure the model accuracy and analyze the complexity reduction in the manipulator model, an evaluation through comparison with the actual system has been performed. The evaluation shows promising result and provides possible future work for the model improvement. The Modelica model of the youBot manipulator is available freely for used as teaching tool or further experimentation.

Keywords: Control; Manipulator; Modelica; youBot

1 Introduction

Models and simulation tools are crucial in robotic research. Although there have been major improvements in the electronic and mechanical field in recent years, robot is still a rare and expensive equipment. The use of model and simulation overcomes this scarcity problem and accelerates the improvement of the actual robot. Robot models allow researchers and university students to access different robots simultaneously. Furthermore, experimentation with models is cost and time efficient due to its ability to be automated, conditioned and accelerated.

youBot[1] is a robot platform designed to serve as reference platform for industry, research and education. youBot is publicly available and the standard youBot configuration consists of an omnidirectional mobile platform and a five DOFs manipulator with a two finger gripper. In practice, the youBot has been modified with an additional laser scanner or 3D camera sensor. Due to its frequent use as a test subject for educational purpose or investigation of new methods, a model for the youBot is highly advantageous.

In this paper, the development of a Modelica[1] model for the youBot manipulator is presented. Modelica is a non-propriety, object-oriented, description language for multi domain modeling which is maintained by the non-profit Modelica association. This makes Modelica suitable for use in education and research (similar to the purpose of the youBot). Although it is non-propriety, the Modelica language is also used in commercial modeling software. The Modelica library for youBot manipulator were also developed using the commercial software Dymola[2]. Rather than creating the complete system in one model, the manipulator is divided into several smaller components model. The components model are stored in a new Modelica library and categorized in different packages based on its functionality.

It is important to note that a robot model is a representation of the actual system to a certain extent and the benefit of having a robot model only holds true when the model is accurate. Otherwise, using the robot model can become a problem rather than a solution [2]. Therefore, the development of the manipulator model were followed by an evaluation through comparison with the actual system. Through the evaluation, the influence of idealistic models, approximation and estimation value in the model were analyzed.

This paper is constructed as follows. After this introductory section (Section 1), Modelica related robotic research will be presented in Section 2. Section 3 presents the specification of the youBot manipulator and Section 4 describe the Modelica Library for the youBot manipulator. Afterward, section 5 presents the evaluation of the manipulator model. Finally, section 6 summarizes the work and provides possible future work.

[1] www.modelica.org
[2] www.3ds.com/products-services/catia/portfolio/dymola

2 State of The Art

Robotic tools with a model of the youBot are Virtual Robot Experimentation Platform or VREP [3], We-bots [4] and Gazebo [5]. These tools focus on simulating the robot interaction with its surrounding environment (navigation, object manipulation) and have its limitation when simulating the robot internal components (mechanical, electrical, and control system). Modeling the robot internal component require multi-domain capability such as provided by Modelica description language. Modelica has been used for modeling spider robotic arm [6], 6-axis industrial robot[7, 8, 9], 3 DOFs parallel Gantry-Tau robot[10], 5 DOFs manipulator[11] and mobile platform[12, 13].

In most cases, a robot model in Modelica is being used for investigating manipulator dynamics (design, optimization and motion control). Such research requires iteration of manipulator motion and adjustment to the controller which can have damaging effect when executed in a real robot. [8] performed optimization through iteration to find a compromise between acceleration, velocity and energy consumption and [9] solved the minimum time optimization problem. [7] derives the inverse dynamic model of a manipulator using algorithms for differential-algebraic equation available in the Dymola software. Dymola is a commercial modelling software based on Modelica. Dymola was also used in [11] to design a picking manipulator for agriculture purpose. [10] develops method for kinematic calibration with the Modelica model of parallel Gantry-Tau robot. Aside in optimization of motion control, Modelica robot models have also been used for telemanipulation [6], robot communication [13] and teaching tools [12].

3 The youBot Manipulator

The specification of the manipulator was acquired from the following sources:

- official youBot website[1],
- email communication with the official distributor of the youBot manipulator[2] and
- discussion with researcher from BRICS[3] who was involved in the development of the software for youBot manipulator.

[1]http://youbot-store.com/
[2]info@locomotec.com
[3]http://www.best-of-robotics.org/

This section consist of two subsections, *kinematic chains* and *control system*. Due to the nature of the robot which is actively being developed, the description presented is subject to changes.

3.1 Kinematic chain

The youBot manipulator is a serial chain manipulator with five revolute joints (shown in Figure 1). The manipulator is equipped with a two finger gripper as its end effector and each finger weights 0.001 kg. When compared to the overall manipulator body, the gripper and its motion has insignificant influence to the system dynamic. Therefore, the gripper is modeled only for the visualization purpose.

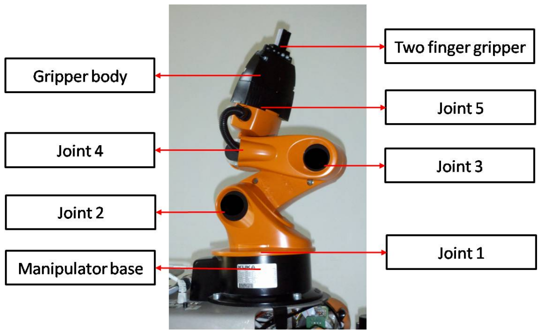


Figure 1: The youBot manipulator

The manipulator height is 65.5 cm (when fully extended), weights 6.3 kg and has a payload of 0.5 kg. Each joint is actuated by brushless dc motor and gearbox with different specification. The kinematic chains, joint range and dynamic property of the manipulator are presented in appendix A.

3.2 Control System

The control system accommodates position, velocity and current control in each joint. The control system consists of: 1. three cascade *proportional-integral-derivative* or PID controller, 2. a velocity ramp or v-ramp generator and 3. a space vector pulse width modulation (SVPWM). There are two mode available for joint position control, *PID* and *v-ramp* mode. In PID mode, the joint velocity is calculated in the PID controller whereas in v-ramp mode the joint velocity is generated with trapezoidal velocity profile characteristic. In this paper, the developed model is based on the joint position control in PID mode. Figure 2 shows the overview of the manipulator's controller. When in v-ramp mode, the PID position in Figure 2 is replaced with the v-ramp generator. θ is the joint angle, v is the

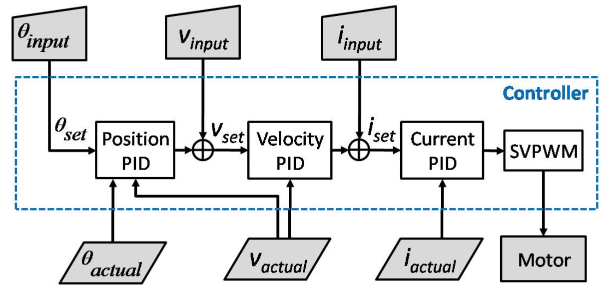


Figure 2: Controller overview

joint velocity and i is the motor current. The *set* variables (θ_{set}, v_{set}, i_{set}) are the input values for the PID, the *actual* variables are the values from the system's sensor and the *input* variables are the user defined values. When controlling the joint angle, a user provide the θ_{input} for the controller and the velocity PID receive the output of position PID as its v_{set}. When controlling the joint velocity, a user provide the v_{input} for the controller which directly forwarded as v_{set} to the velocity PID (the output of the position PID in such cases will be ignored). SVPWM's output sets the voltage in the joint's motor.

The PIDs for position, velocity and current have similar architecture. As a representative of the PID controllers, Figure 3 shows the flowchart of the velocity PID. e is the error or the difference between the set

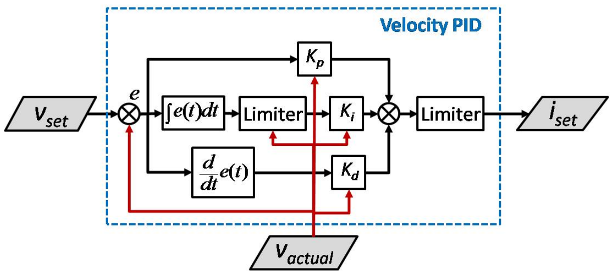


Figure 3: Velocity PID overview

value and the actual value. K_p, K_i, K_d are the gain parameters for proportional, integral and derivative component. v_{output} is forwarded to current PID as i_{set}. As observed in Figure 3, the velocity PID controller is similar to the text book PID as follows:

$$C_t = K_p e(t) + K_i \int_{t-\Delta t}^{t} e(t)dt + K_d \frac{d}{dt} e(t) \quad (1)$$

Where C is the controller output and Δt is the PID period. However, the gain parameter in the velocity PID adjust itself based on the motor velocity as follows:

$$k = \begin{cases} k_2 & \text{if } |v| \geq a \\ k_1 + (\frac{|v|}{a} * (k_2 - k_1)) & \text{if } |v| < a \end{cases} \quad (2)$$

Where k is the gain parameters (K_p, K_i or K_d in Equation 1), k_1 is the lower boundary of the gain parameter, and k_2 is the upper boundary of the gain parameter value, v is the motor velocity and a is the threshold value. Similar structure is also implemented in the PID for position control. Therefore, the position PID and velocity PID will be referred as the non-linear PID controller. The non-linear PID enables the user to set different control behavior for low and high velocity. Similar to the gain parameters, limiters in the position and velocity controller have non-linear characteristic where the limit value is defined by the motor velocity.

4 The *youBot* Modelica Library

The library were developed using "divide and conquer" principle with emphasize on modularity, reusability and abstraction. This approach provides modularity in the manipulator model which enables user to exchange components and perform component based experimentation. Additionally, a template model will be provided for sub-systems and parts which are frequently used in the manipulator model (reusability). In such cases, the model will have adjustable parameter sets to be configured based on its implementation. Finally, the manipulator model is developed in different abstraction layers based on its complexity (Figure 4). The lower layer provides a more detailed information in each component and the upper layer provides the general overview of the system. Through inheritance, the selected parameter sets of the component in the lower level can be configured from the upper level.

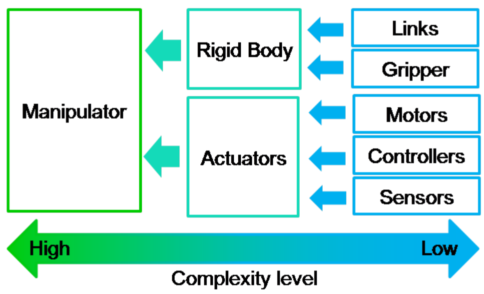


Figure 4: Complexity based abstraction layer

In every modeling process, simplification and estimation is unavoidable mainly due to the following reasons:

- *Computational load.* The model's level of complexity should consider its influence in the model accuracy and its computational load.

- *Limited information*. Specifically in the technology domain, information is the key point for competing in the global market. In such cases, manufactures choose not to provide the complete information of their product to avoid copycat.

The use of simplification and estimation will be presented in the description of each package. Furthermore, the influence of the model's complexity reduction will be analyzed in Section 5. The youBot Modelica library consists of the following four packages which are*Controller* package, *Axis* package, *Body* package and *System* package. The library is developed with the use of several packages in MSL such as Modelica.Blocks.Math for standard mathematical functions and Modelica.Mechanics for 3-dimensional mechanical system. This paper follows the same Modelica convention in describing the models. The initial of a model or a package uses capitalized letter. When necessary, the model is defined with its package name. The model Modelica.Blocks.Interfaces.RealInput refers to the model *RealInput* which is inside the package *Interfaces*. The Interfaces package is inside the *Blocks* package and the Blocks package is inside the Modelica Standard Library or MSL. Finally, an instance of a model is written in small letters aside from a few exceptional cases (i.e. T for temperature variable).

4.1 Controller Package

The Controller package consists of the components for the manipulator control system. The Controller package is divided into three packages which are the *Components* package, the *PIDs* package and the *Modes* package. The Controller.Components package consists of models which have low complexity level. Figure 5 show the models in the Component package. As in Modelica, the instance's name of a model is placed on the upper part of the symbols in blue color. The model *Limiter2* and *Gain2* (Figure 5b and 5a) perform the calculation for non-linear PID controller (Equation 2). The model *DisDer* (Figure 5c) produces the derivative value of a specific time period from a discretized continuous input. The model *P2V* (Figure 5d) converts PWM rate to voltage rate which will be used to replace the SVPWM in the controller (Figure 2).

The Controller.PIDs package consists of three different PID models which are the *Position*, *Velocity* and the *Current* model. As the name suggested, the models are the PID controllers for position, velocity and current in the youBot manipulator (Figure 2). As a representative, Figure 6 shows the Velocity model from the PIDs package. *v_set*, *v_actual* and *i_set* represent v_{set}, v_{actual} and i_{set} in Figure 3 respectively. The additional component *N* in the Velocity model is to produces the current output in mA to mimic the readings in the actual system.

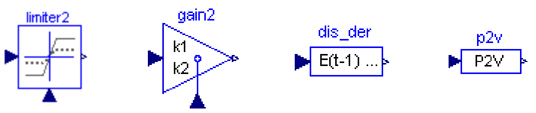
Finally, the Controller.Modes package is for different types of control mode. Currently, the available model in the Modes package is the *Position* model (Figure 7). The output *V_set* represent the voltage value which will be connected to the motor's power supply unit. Using the same approach, the model for controlling the joint velocity or the motor current can be developed.

4.2 Axis Package

The Axis package is for different models of the manipulator's actuator. The Axis package consist of the *Position* model (Figure 8). *DCPM* is the brushless DC motor model, *PSU* is the power supply unit model, *R* is the gearbox model, *F* is the friction model, *controller* is the Controller.Mode.Position model (Figure 7), *sens_v* is the joint's velocity sensor, *sens_theta* is the joint's position sensor and *joint* is the connection to the manipulator's joint. The model for DC motor, power supply unit and friction uses the ideal system model from the MSL. The *controller* output (*V_set* in Figure 7) is connected to the PSU and its input is extended for the Axis.Position input as *theta_set*. The output of *sens_v*, *sens_theta* and the value of DCPM's current is connected to the control system (*theta_actual*, *v_actual* and *i_actual* in Figure 7). The adjustable parameter sets in the Axis.Position model includes gearbox ratio, controller parameter and motor specification.

4.3 Body package

The Body package consist of models for the manipulator's kinematic link. The Body package has three models which are *Gripper*, *Link* and *Manipulator*. The Gripper model is the rigid body model of the youBot two finger gripper. The Link model is the rigid body model for one manipulator link (Figure 9a). *RF_rot* is the link's reference frame rotation, *frame_a* is the connection to the previous link, *frame_b* is the connection to the next link, *joint* is the link's rotating part which is connected to the actuator through *motor*, and the *body* represent the rigid body of the link. The additional component *RF_vis* is to provide user with the visualization of the link's reference frame. The Manipu-



(a) Limiter2 (b) Gain2 (c) DisDer (d) P2V

Figure 5: The Controller.Component Package models

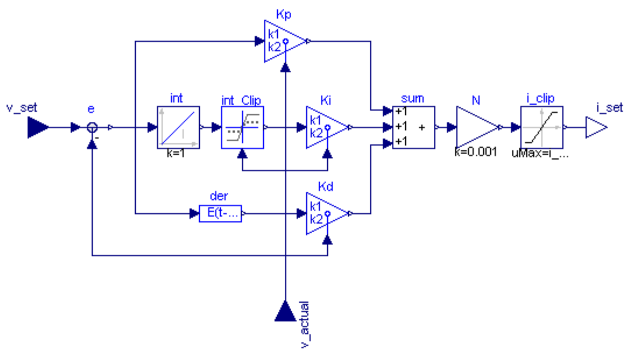


Figure 6: The Controller.PIDs.Velocity model

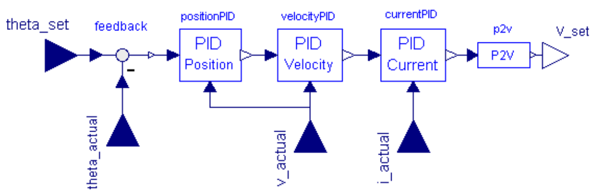


Figure 7: The Controller.Mode.Position model

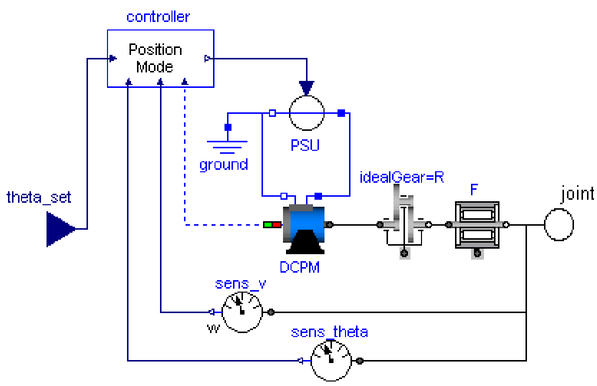
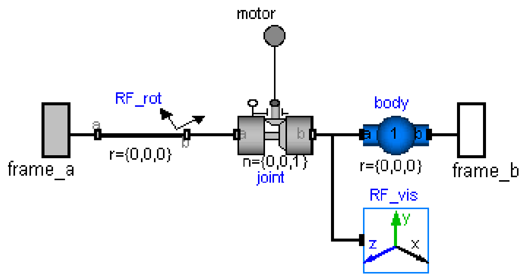
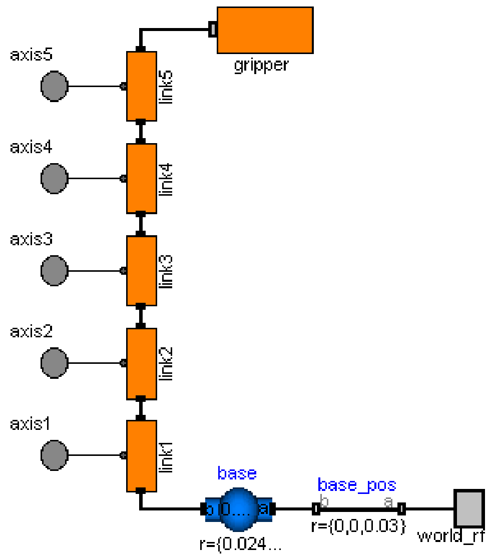


Figure 8: The Axis.Position model



(a) The Body.Link model



(b) The Body.Manipulator model

Figure 9: The Body package models

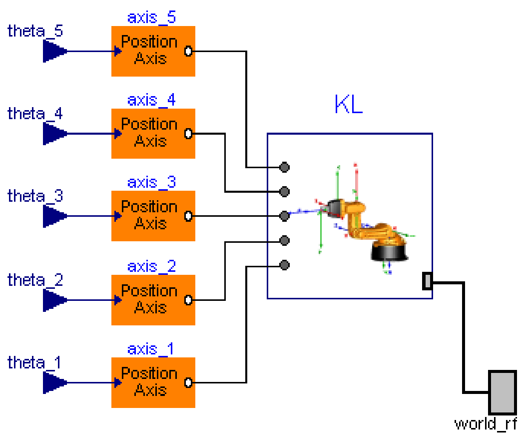


Figure 10: The System.Position model

lator model is the rigid body model of the youBot Manipulator's kinematic link (Figure 9b). The *base_pos* and *base* represent the base of the manipulator and its position in the world reference frame respectively. The Manipulator model has five connectors ($axis1$ to $axis5$) for each joint and one connector for the world frame of reference ($world\_rf$).

4.4 System package

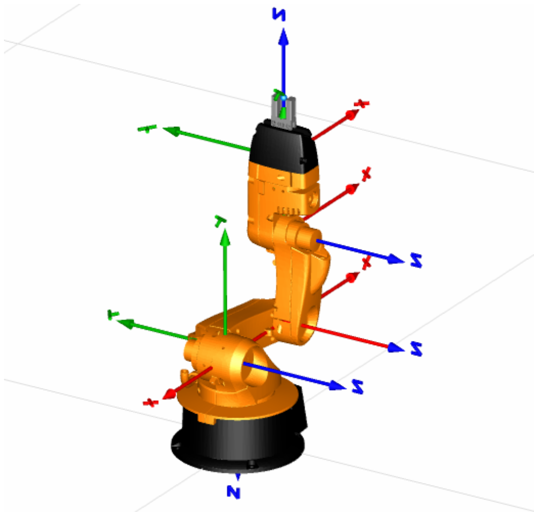
The System package is for the manipulator complete and ready-for-use models. The System package has two models which are, the *Dummy* model and the *Position* model. The Position model is the model of youBot manipulator's kinematic link and its actuators. Figure 10 shows the Position model from the System package. *KL* is the rigid body model of the youBot manipulator's kinematic link (Figure 9b) whereas *theta_5* represent the user defined joint angle and *axis_5* is the actuator (motor and control system) for joint 5. In the Dummy model, the kinematic link is connected to a dummy actuators (Modelica.Mechanics.Rotational.Speed) where the user can set the velocity of each joint directly. The Dummy model is used for comparison evaluation in Section 5.

5 Result

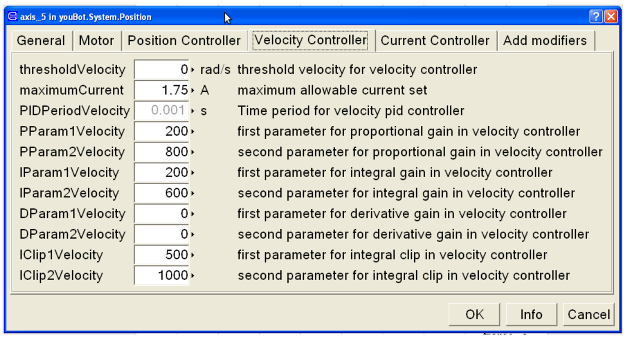
Figure 11a shows the Dymola's visualization of the manipulator model (System.Position) and Figure 11b shows an example of the parameters set configuration for the velocity controller in the fifth joint of the manipulator. The parameter name in the model are consistent with the existing driver and firmware. The ability of changing the parameters showcase an advantage of having the manipulator model in Modelica. The configuration process in the actual system is time consuming and experimentation with the control parameters can have damaging effect to the motor. Additionally, changing the motor and the controller requires disassembling the manipulator.

5.1 Evaluation

A comparison with an actual system was performed after the development of the manipulator model. The evaluation is a preliminary test which purpose is to provide an informed estimation for the model accuracy and to isolate the major components which requires further development. As a preliminary test, the comparison aspect is limited to the joint position



(a) Model visualization



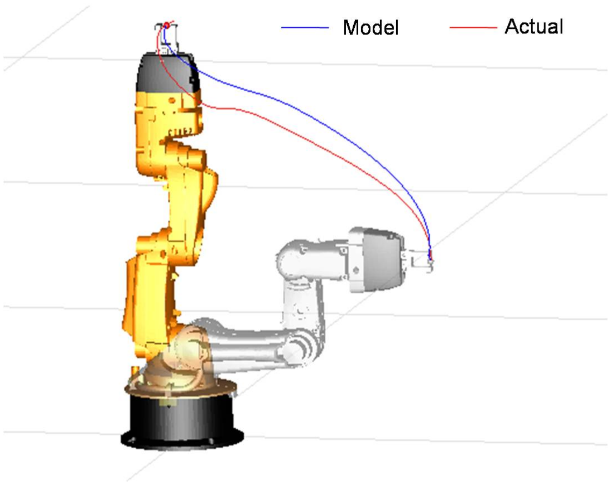
(b) Parameter adjustment

Figure 11: The youBot manipulator model

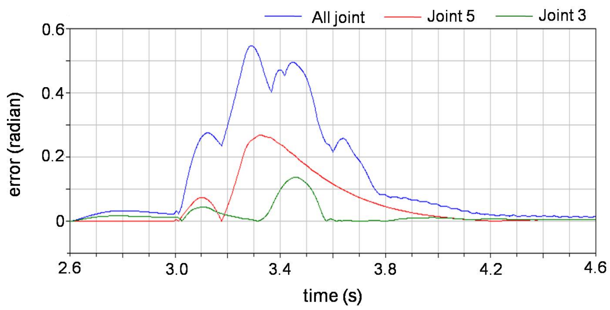
and the joint velocity throughout a point-to-point motion. The comparison were performed using Dymola. For the actual system, the joint velocity is recorded while performing the motion (the sensor measurement is assumed to be accurate). Afterward, the recorded joint velocity is used as input for the System.Dummy model. In the same setting, a manipulator model (System.Position) is also performing the same motion. The motion involves all joint to move 90° to evaluate the model accuracy as a complete system. Such motion was chosen so that the resulting error will be the accumulation of every approximation and simplification in the model.

5.2 Analysis

Figure 12a shows the path measured in the actual system and the path generated by the model. The gray colored youBot manipulator is the starting pose of the motion. The path generated by the model is smoother that the actual system as a result of the idealistic con-

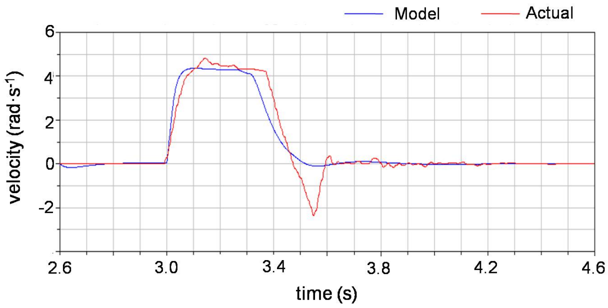


(a) Path deviation visualization

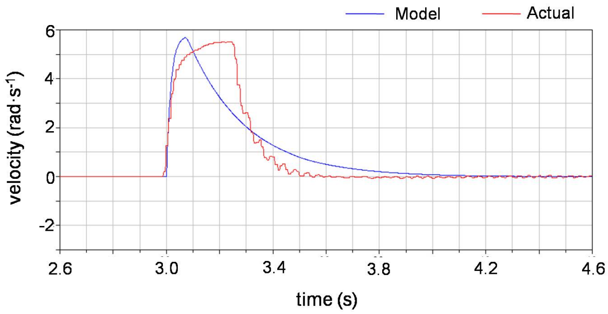


(b) Joint angle difference

Figure 12: Evaluation result



(a) Joint 3



(b) Joint 5

Figure 13: Joint velocity comparison

dition in the simulation. The absolute difference in joint angle (Figure 12b) peaked with the value of 0.55 radian. The difference in each joint depends on the maximum velocity parameter (v_{max}) in the controller. As shown in Figure 12b, joint 3 ($v_{max} = 4.19$ rad $\cdot$ s^{-1}) has a considerably lower peak than joint 5 ($v_{max} = 5.90$ rad $\cdot$ s^{-1}). The difference in joint angle peaked at two points. The time stamps shows that the two peak points are at the start of a velocity change (from stop to move and slowing down from a constant velocity). The joint velocity in the actual system is less stable than in the simulation (Figure 13a). This is the result of the motor vibration which is excluded from the manipulator model. Finally, the ideal motor model results in the deviation on higher velocity as shown in Figure 13b. Similar phenomena in joint velocity and joint position were also found in other joints. From this result, it is concluded that the motor model is the major factor of the deviation. The friction parameter, SVPWM model and the inertia estimation tensor are the other aspects which contribute to the deviation. However, the investigation of these components will result in a less significant increase of the model accuracy.

6 Conclusion

In this paper, the development of the Modelica model of the youBot manipulator is presented. The Modelica library for the manipulator model provides the user with modularity, reusability and abstraction. The model accuracy has been evaluated through comparison with the actual system. The evaluation result shows that the model reflect the actual system within a reasonable deviation. The analysis shows that the motor model is the major contributing aspect in the deviation between the model and the actual system. Possible improvement for the developed Modelica library is the development of a more accurate motor model and a more comprehensive test for the manipulator components (controller components and rigid body). For future works, the manipulator model is planned to be used for hardware-in-the-loop experiment. The development or design of other manipulator model is also possible through the reusability of the components in the Modelica library. The library is already available publicly[1] to be used as teaching tools or experiment involving manipulator dynamics, load identification, fault analysis and motion control.

[1] www.youbot-store.com

References

[1] R. Bischoff, U. Huggenberger, and E. Prassler, "Kuka youbot - a mobile manipulator for research and education," in *IEEE Int. Conf. on Robotics and Automat. (ICRA)*, pp. 1 –4, may 2011.

[2] P. Fritzson, *Principles of Object-Oriented Modeling and Simulation With Modelica 2.1*. IEEE Press, 2004.

[3] M. Freese, S. Singh, F. Ozaki, and N. Matsuhira, "Virtual robot experimentation platform v-rep: a versatile 3d robot simulator," in *Proc. of the Second Int. Conf. on Simul., Model., and Program. for Auton. Robots*, SIMPAR'10, (Berlin, Heidelberg), pp. 51–62, Springer-Verlag, 2010.

[4] O. Michel, "Webots: Professional mobile robot simulation," *J. of Adv. Robotics Syst.*, vol. 1, no. 1, pp. 39–42, 2004.

[5] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *IEEE/RSJ Int. Conf on Intell Robots and Syst.*, pp. 2149–2154, 2004.

[6] G. Ferretti, M. Gritti, G. Magnani, and P. Rocco, "A remote user interface to modelica robot models," in *Proc. of the 3rd Int. Modelica Conf.*, pp. 231–240, 2003.

[7] M. Thuemmel, M. Otter, and J. Bals, "Control of robots with elastic joints based on automatic generation of inverse dynamics models," in *Proc. of 2001 IEEE/RSJ Int. Conf. on Intell. Robots and Syst*, pp. 925–930, 2001.

[8] A. Kazi, Merk, M. G. Otter, and H. Fan, "Design optimisation of industrial robots using the modelica multi-physics modeling language," in *Proc. 33rd ISR Int. Symp. Robot*, pp. 347 –352, 2002.

[9] M. Hast, J. Åkesson, and A. Robertsson, "Optimal robot control using modelica and optimica," in *Proc. of the 7th Int. Modelica Conf.*, Modelica Association, Sept. 2009.

[10] I. Dressler, J. Schiffer, and A. Robertsson, "Modeling and control of a parallel robot using modelica," in *Proc. 7th Int. Modelica Conf.*, (Como, Italy), Sep 2009.

[11] Y. Chen, S. Jin, X. Zou, D. Xu, and W. Cai, "Study of modeling and simulating for picking manipulator based on modelica," in *Proc. of the 2nd Int. Conf. on Intell. Robotics and Appl.*, (Berlin, Heidelberg), pp. 1211–1216, Springer-Verlag, 2009.

[12] J. Åkesson, U. Nordstroem, and H. Elmqvist, "Dymola and modelica_embeddedsystems in teaching - experiences from a project course," in *Proc. of the 7th Modelica Conf.*, 2009.

[13] U. Pohlmann, S. Dziwok, J. Suck, B. Wolf, C. Loh, and M. Tichy, "A modelica library for real-time coordination modeling," in *Proc. of the 9th Int. Modelica Conf.*, 2012.

A Manipulator Specification

Table 1: Kinematic chain

	Parent frame	Translation (*cm*)			Rotation (degree)		
		x	y	z	x	y	z
Joint 1	Base	2.4	0	11.5	180°	0°	0°
Joint 2	Joint 1	3.3	0	0	90°	0°	-90°
Joint 3	Joint 2	15.5	0	0	0°	0°	-90°
Joint 4	Joint 3	0	13.5	0	0°	0°	0°
Joint 5	Joint 4	0	11.36	0	-90°	0°	0°
Gripper	Joint 5	0	0	5.716	90°	0°	180°

Table 2: Joint range

	Joint 1	Joint 2	Joint 3	Joint 4	Joint 5
Joint range	-169° 169°	-65° 90°	-151° 146°	-102.5° 102.5°	-165° 165°

Table 3: Dynamic Properties

	Mass (*kg*)	Intertia Tensor Elements ($kg{\cdot}cm^2$)		
		I_{xx}	I_{yy}	I_{zz}
Link 1	1.39	29.525	60.091	58.821
Link 2	1.318	31.145	5.483	31.631
Link 3	0.821	17.2767	4.1967	18.468
Link 4	0.769	6.764	10.573	6.61
Link 5	0.678	1.934	1.602	0.689
Gripper	0.201	2.324	3.629	2.067