

# Non-linear Parameter Identification for Humanoid Robot Components

Parastoo Dastango<sup>1</sup>, Alex Ramirez-Serrano<sup>2</sup>

Department of Mechanical and Manufacturing Engineering, University of Calgary  
2500 University Drive N.W., Calgary, Alberta, Canada  
<sup>1</sup>parastoo.dastango@ucalgary.ca; <sup>2</sup>aramirez@ucalgary.ca

**Abstract** - This paper describes a kinematic calibration method applied to a humanoid robot arm having seven degrees of freedom. The methodology is developed and used to determine the parameters of robotic artifacts having a non-linear model. The parameter identification process can be used to identify diverse parameters such as the distance between the joints, moments of inertia, and others. The proposed methodology comprises four main steps: First, the kinematic model of the manipulator is determined using the Denavit-Hartenberg convention. Then, measurement values of the robotic artifact of interest are collected via a Multibody Dynamics simulation software. Subsequently, the desired system parameters are identified using a nonlinear identification algorithm. In this third step, the Gauss-Newton method is employed to linearize the non-linear model of the system, and the least squares method is used to find the best fit for the parameters. As the last step, the obtained identified model (model with identified parameters) is validated by comparing the behaviour of the manufactured system with its simulation behaviour. For this, sine wave signals are used as inputs to the actuators comprising the robot. The results show that the behaviour of both methods perfectly match which verifies that the identified parameters are correctly estimated and highlights the value of model calibration in the control of humanoid robots.

**Keywords:** Humanoids, System Identification, Kinematic Calibration, Robot Manipulators, Parameter Model Identification

## 1. Introduction

Robotic manipulators are being used in a variety of applications such as assembly lines [1], underwater welding [2], collaborative robots (cobots), and assistance for people with disabilities [3]. Among the extensive applications of robotic arms, their use in humanoid robots comprising two legs and two arms enabling their use, at least in theory, in the same environments as humans (e.g., manufacturing floors), and perform cooperative tasks with people (e.g., assembly operations and inspection of parts). In this context, their physical interactions with people must be safe and to some extent predictable. Therefore, the movement accuracy of humanoids and cobots is of vital importance. However, one of the factors which causes inaccuracy in a robot's operation are the unavoidable differences between the geometric characteristics of the robot's kinematic model (i.e., robot's design) and the manufactured characteristics (dimensions). Although these errors tend to be relatively small, they might become significant when the robot's linkages are arranged in a given geometrical configuration such as in series which cause the errors to accumulate and ultimately affect the motion accuracy of the robot such as its walking, object manipulations abilities, or its capability to safely interact with humans. Therefore, small errors during designing or manufacturing the components comprising the robot (e.g., linkages, and gear boxes) will cause a considerable error in placing/moving the robot's end-effector, feet, and other of its body components (e.g., sensors). As a result, a negative impact on the model-based controlling system of the robot(s) will be result [4].

The rest of this paper is organized as follows: Section 2 provides a brief literature review. Section 3 provides a description of the proposed approach. Then Section 4 presented the process of generating the kinematic model, followed by description of the non-linear system identification approach and the concept of kinematic calibration in Section 5. In Section 6, the results of the calibrated model are presented, and compared with the results of the Multibody Dynamics simulation software. Finally, conclusions are provided in Section 7.

## 2. Literature Review

One tool that has been used to deal with the problems identified in Section 1 is system or model identification, a practical solution to improve the mathematical model of systems for which the exact value of its parameters are inaccurate. [4]. In some cases, systems, including redundant robotic systems or systems whose characteristics change over time, are so complex that deriving an accurate mathematical model directly from conventional rules of physics is computationally intensive and

costly. In such cases, system identification uses the systems' inputs and outputs data to estimate the needed parameters including geometrical dimensions, moments of inertia, and other not easy to obtain parameters such as coefficients of friction.

There are few cases where the system identification process can be implemented using only raw data from measurements as the inputs that guide the method. However, due to the fact that the performance of any given system is a coupling effect between the inputs and the system parameters, in most cases, the system identification process is implemented using some technical theoretical knowledge about the system itself in addition to the inputs and outputs data. That is, the type and order of the model should be estimated to find which method of the system identification should be implemented. For instance, neural network [5] and formal mathematical algorithms for complex non-linear highly coupled systems, and linear time-invariant models [6], respectively are typically used each having their pros and cons. Although formal algorithms produce highly accurate models they are limited to simple systems and the techniques (e.g., machine learning, neural networks) used for non-linear systems are typically black-box only approximations. Unfortunately, non-linear systems represent most, if not all, of real-life systems.

There is a vast literature on system identification and its application to robotics. Humanoid system identification has been mostly focus on enhancing the stability of robot during walking. The authors in [7] consider the dynamics of walking robot errors associated with the idea of simplifying the multi-body robot to an inverted pendulum. They resolve the associated complexities by utilizing the system identification method to the stability model of the robot during walking. Others apply a non-linear system identification approach on a manipulator of a humanoid robot to estimate the contact model [8]. Beyond humanoid robot, there are several studies which focus on the kinematic calibration of arm manipulators including the kinematic calibration on parallel manipulators [e.g., 9]. Unlike parallel manipulators, serial manipulators tend to have a higher level of error due to the accumulative effect of errors in its kinematic chain, thus extensive work has been devoted to the kinematic calibration of serial manipulators, [e.g., 10 and 11]. The calibration process for any given system, specially robotics, is highly dependent on the available equipment for measurement. As a result, the process tends to be expensive and time-consuming [e.g., 12].

### 3. Proposed Approach

In this work, the non-linear model identification of a 7 degrees of freedom (DoF) serial manipulator (Figure 1) is investigated. In what follows, it is assumed that the arm and its corresponding dimensions shown in Figure 1 represents the left arm of a life-size humanoid robot which needs to be accurately modelled so that an effective controller can be designed to move the arm and humanoid with high precision. Thus, we aim to calibrate the designed arm in such a way that the desired output of the arm (i.e., the position and motion trajectory of its end-effector) matches the desired trajectory output as computed from the arm's kinematic model. In what follows, it is assumed that due to errors in manufacturing and assembling, the physical dimensions of the real robot arm do not match the CAD design and measuring the needed parameters (e.g., moments of inertia) is difficult. For this, a parameter identification method is implemented to determine the lengths of the linkages comprising the arm (i.e., the distance between each pair of joints forming the robot arm). In what follows we refer to the designed link lengths and their identified (real) values as  $d_{initial}$  and  $d_{identified}$ , respectively. Such parameter was chosen because the  $d_{initial}$  values tend to be highly prone to errors.

The typical four main steps in every kinematic model calibration are: i) Modelling, ii) Measurement, iii) Parameter identification, and iv) Compensation [10]. For the modelling step, the kinematics of the manipulator are derived using typical robot forward kinematics principles and Denavit-Hartenberg (D-H) convention. For the measurement step, the CAD model of the robot is imported and simulated into a Multibody Dynamics simulation software from where precise motion and reaction forces are obtained. From such simulating environment all working environment conditions (e.g., collisions, reaction forces, etc.), and the output data sets (motion trajectory) are obtained. Subsequently, such information is imported to the parameter identification algorithm (Step 3). The main idea behind this proposed method of parameter identification is to minimize the difference between the current position of the end-effector and the desired location that is obtained from the robot's kinematic model. In the area of linear systems, the least square method is mostly employed for the identification process. In non-linear systems, however, using an ordinary least square method is not applicable;

therefore, in the proposed approach, the Gauss-Newton method is first used to linearize the theoretical kinematic model [13]. After obtaining the identified parameter values,  $d_{identified}$ , the obtained parameters replace the theoretical parameter values. Herein, we refer to this model as the “new model”. The new model is then validated by comparing the trajectory outputs of obtained from the simulator and the new kinematic model.

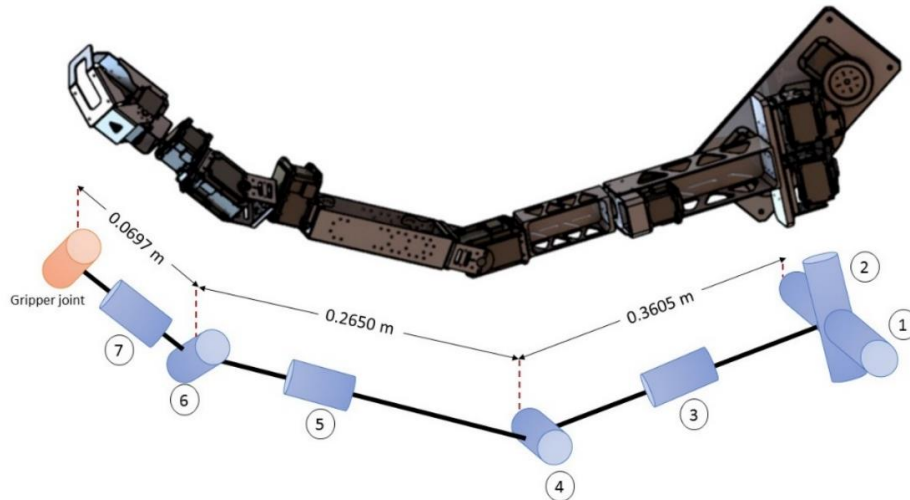


Fig. 1: 7-DoF robotic manipulator used in the case study.

Unlike previous developments, in this work, the theoretical manipulator is simulated in a Multibody Dynamics simulator running a physical engine from where the measurement data is estimated by simulating all inertial and gravitational forces which exist in the real-environment where the robot will operate.. The first advantage of this method is being significantly cost effective when compared to other methods that requires specialized hardware to be mounted on the robot. Secondly, measurement equipment such as Coordinate Measuring Machines or laser trackers have their own sources of errors which cause inconsistency on the output data of measurements, and consequently, on the calibrated model. However, a well-simulated model in the simulation environment does not contain such kind of errors. Lastly, calibrating a model in a simulation environment, and before the robot is manufactured, gives insightful tips about how to modify/enhance the mechanical design of the robot in order to reduce the number of errors when the manipulator is manufactured.

#### 4. Modelling: Kinematic Model

In this section, the kinematic model of the robot arm shown in Figure 1 is presented. As a multi-body system, the arm comprises a set of seven rigid bodies (i.e., links) joined by seven interconnection elements driven by actuators. The interconnection elements are revolute joints that constrain the relative motion of the corresponding pairs of links.

##### 4.1. Forward Kinematics

The forward kinematics analysis representing the structural formulation that is used to calculate the configuration of the end-effector as a function of the joint variables is formulated via the joint variables (the angles between the links). To perform the kinematic analysis, a local coordinate frame of  $O_i x_i y_i z_i$  rigidly attached to each of the seven joints,  $i^{th}$  link, is used. In order to express the motion of each link in the inertia frame of reference  $(O_0 x_0 y_0 z_0)$  the homogenous transformation matrices that expresses the position and orientation of  $O_i x_i y_i z_i$  with respect to  $O_j x_j y_j z_j$  is herein denoted by  $T_j^i$  and defined as per Equation (1):

$$T_j^i = \begin{cases} \begin{bmatrix} R_j^i & O_j^i \\ 0 & 1 \end{bmatrix} & \text{if } i < j \\ I & \text{if } i = j \\ (T_j^i)^{-1} & \text{if } j > i \end{cases} \quad (1)$$

where  $R_j^i$  denotes the orientation of frame of reference  $O_j x_j y_j z_j$  relative to  $O_i x_i y_i z_i$  computed by the rotational parts of the T-matrices as per Equation (2):

$$R_j^i = R_{i+1}^i \dots R_j^{j-1} \quad (2)$$

The coordinate vectors  $O_j^i$  are then computed via Equation (3):

$$O_j^i = O_{j-1}^i + R_{j-1}^i O_j^{j-1} \quad (3)$$

#### 4.2. Denavit-Hartenberg Representation

To formulate the cumulative effect of the entire set of joint angles for the 7-link manipulator the classical Denavit-Hartenberg representation is used. In such a convention, the homogeneous transformation  $T_i^{i-1}$ , represented as a product of the four basic transformations,  $Rot_{z,\theta_i}$ ,  $Trans_{z,d_i}$ ,  $Trans_{x,a_i}$  and  $Rot_{x,\alpha_i}$  is used which resulted in Equation (4):

$$T_i^{i-1} = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) & 0 & a_{i-1} \\ \sin(\theta_i)\cos(\alpha_{i-1}) & \cos(\theta_i)\cos(\alpha_{i-1}) & -\sin(\alpha_{i-1}) & -\sin(\alpha_{i-1})d_i \\ \sin(\theta_i)\sin(\alpha_{i-1}) & \cos(\theta_i)\sin(\alpha_{i-1}) & \cos(\alpha_{i-1}) & \cos(\alpha_{i-1})d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

where  $a_i$ ,  $\alpha_i$ , and  $d_i$  are the arm constants representing the link length, link twist, and the link offset, respectively while  $\theta_i$  denotes the angle of joint “ $i$ ”. In this case study, as mentioned in Section 3, the identification of  $d_i$  parameter ( $d_{identified}$ ) are to be determined. In Eqn. (4) the parameter  $a_i$  is different from the distance between the joints. Thus, the reader should be aware that the proposed algorithm identifies the  $d_i$  parameters not  $a_i$  (Table 1).

With the obtained set of  $T_i^{i-1}$  matrices for all the links comprising the arm, the transformation matrix to transforms the coordinate of the end-effector to the inertia coordinate is calculated, Equation (5). Table 1 shows the values of D-H parameters for each link, while Figure 2 presents the skeleton of the arm where the location of each of the eight local coordinate frames (including the inertial frame of reference from where the position of the end effector is computed) attached to their links of reference is shown. As shown in Figure 2, to simplify the mathematical formulation, using the D-H convention two or three local frames are positioned at the same origin. Therefore, from the value of the seven parameters that we aim to identify, at least in theory, only two of them (Links 3 and 5) have non-zero values (Table 1).

$$T_7^0 = T_1^0 \times T_2^1 \times T_3^2 \times T_4^3 \times T_5^4 \times T_6^5 \times T_7^6 \quad (5)$$

Table 1: D-H parameters values for the 7-DoF manipulator.

<b>i</b>	<b><math>\theta_i</math></b>	<b><math>d_i</math></b>	<b><math>a_{i-1}</math></b>	<b><math>\alpha_{i-1}</math></b>
1	$\theta_1$	0	0	0
2	$\theta_2$	0	0	90
3	$\theta_3$	0.3605	0	90

4	$\theta_4$	0	0	90
5	$\theta_5$	0.265	0	-90
6	$\theta_6$	0	0	-90
7	$\theta_7$	0	0	90

The resulting matrix,  $T_7^0$ , represents the kinematic model of the robot arm which is ready for the identification process.

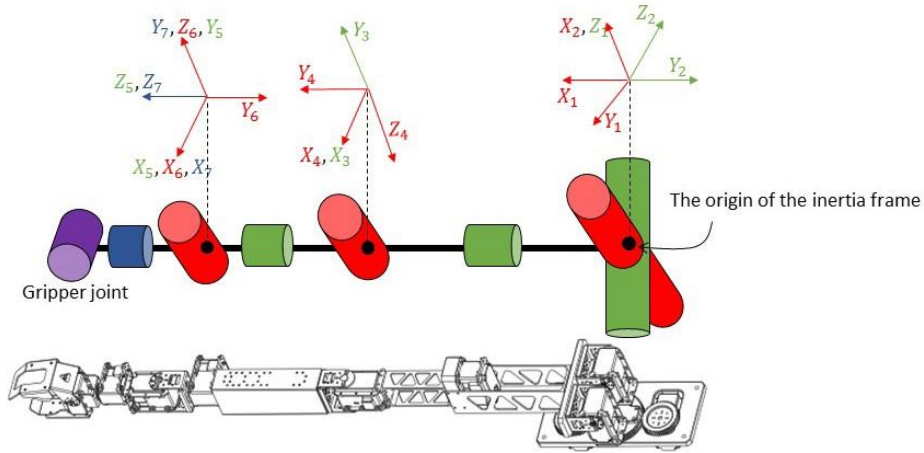


Fig. 2.H representation-The location of local frames regarding D

## 5. Non-linear Model Identification and Kinematic Calibration

Linear and non-linear systems are herein represented as per Equations (6) and (7), respectively:

$$\text{Linear model: } y^l = A^l \varphi \quad (6)$$

$$\text{Non-linear model: } y^l = f(x^l, \varphi) \quad (7)$$

where  $x^l = \{x_1^l, x_2^l, \dots, x_n^l\}$  denotes the input variables vector,  $y^l = \{y_1^l, y_2^l, \dots, y_M^l\}$  denotes the output variables vector, and  $\varphi = \{\varphi_1, \varphi_2, \dots, \varphi_{N_{par}}\}$  denotes the parameters vector where  $N_{par}$  is the number of parameters that it is desired/needed to be identified. In linear systems (Eqn. (6)), the matrix  $A^l$  (known as regressor matrix) is a function of the input variables  $x^l$ , and the least squares approach is utilized to obtain the needed parameters vector  $\varphi$  [13]. Nonlinear models, however, contain a non-linear function of  $f = \{f_1, f_2, \dots, f_M\}$ , so the ordinary least squares method cannot be implemented due to the fact that there is no regressor matrix for such models. One method to overcome this problem is the Gauss-Newton method [14]. This method is herein used to take an initial value of the parameters that one needs to identify and iteratively updates that the parameters by finding a correction value in each iteration. The advantage of this method are diverse: it is fast, and a quadratic convergence to the desired parameters exists; however, it is important to note that the proposed method fails to converge to the actual parameter values when the initial estimation of the parameters differs by a large factor from the actual parameter values. The method also fails when the non-linearity of the system is high. During the kinematic calibration of the proposed methodology, however, the initial values for parameters are known, and the non-linearity of the system is mild (contains only sine and cosine functions). For solving non-linear systems with the Gauss-Newton method, the model is linearized using Taylor series expansion. Considering the parameter estimation at iteration of  $k$ , the linearized model is written as follows:

$$y^l \approx f^l(\varphi^k) + A^l \Delta\varphi \quad (8)$$

where

$$A^l = \frac{\partial f^l}{\partial \varphi} \text{ (Jacobian matrix at } \varphi^k) \quad (9)$$

Therefore:

$$\Delta y^l = A^l \Delta\varphi \quad (10)$$

As a result, the parameter estimation process is transformed into an ordinary least square problem, where  $\Delta\varphi$  is defined as:

$$\Delta\varphi = (A^T A)^{-1} A^T \Delta y \quad (11)$$

Accordingly,  $\varphi^k$  is updated in every iteration per Equation (12). The pseudocode of the proposed methodology is shown in Table 2.

$$\varphi^{k+1} = \varphi^k + \Delta\varphi \quad (12)$$

Table 2: Pseudocode of the proposed method for obtaining the values of the linkage's length parameters.

Inputs to algorithm	Algorithm
Details of 14 poses including: <ul style="list-style-type: none"> <li>- Joint angles (<math>\theta_i</math>): <math>14 \times 7</math> joint angles (14 poses of 7-DoF arm)</li> <li>- link lengths (<math>a_i</math>): <math>\forall i \in \{1, \dots, 7\}</math></li> <li>- link twists (<math>\alpha_i</math>): <math>\forall i \in \{1, \dots, 7\}</math></li> <li>- link offsets (<math>d_i^0</math>): <math>\forall i \in \{1, \dots, 7\}</math></li> </ul> end-effector positions ( $p_{sim}$ ): $14 \times 7$ values (x, y, and z position of 14 poses), imported from simulation software	<b>Step 0:</b> $k = 0, d_i = d_i^0$ <b>Step 1:</b> Calculate $p_{model}$ position of end-effector using D-H convention and transformation matrices for 14 poses. <b>Step 2:</b> Calculate the error in the position of end-effector using imported data and $d_i$ for 14 poses. $error_i = p_{sim} - p_{model}, \forall i \in \{1, \dots, 7\}$ <b>Step 3:</b> Take the derivative of $p_{model}$ where it is variable of $d_i$ , and define the matrix regressor of A, for 14 poses:
<b>Outputs of algorithm</b> <ul style="list-style-type: none"> <li>- Matrix <math>d_{par}</math> which is a <math>1 \times 7</math> matrix including identified values for <math>d_i, \forall i \in \{1, \dots, 7\}</math>.</li> </ul>	$A_i = \frac{\partial p_{model}}{\partial d_i}, \forall i \in \{1, \dots, 7\}; A = \begin{bmatrix} A_1 \\ \vdots \\ A_7 \end{bmatrix}$ <b>Step 5:</b> Obtain the correction in distance using the least square method and matrix A for 14 poses.
<b>Parameters used in the algorithm:</b> <ul style="list-style-type: none"> <li>- <math>k</math>: iteration number.</li> <li>- <math>d_i^k</math>: the length of link offset for <math>i</math> link at iteration <math>k</math>.</li> <li>- <math>p_{model}</math>: the position of end-effector in the model.</li> <li>- <math>p_{sim}</math>: the position of end-effector in the simulation.</li> </ul>	$\Delta d = (A^T A)^{-1} A^T error$ <b>Step 6:</b> Correct the value of $d_i$ in each step until it converges to the final values of $d_i$ . These values are the new identified values of $d_i$ for 14 poses. $d_i^{k+1} = d_i^k + \Delta d, \forall i \in \{1, \dots, 7\}$ $d_i = d_i^{k+1}$ <b>Step 7:</b> If $\Delta d < \varepsilon$ , stop. If not, return to Step 1. ( $\varepsilon$ is the desired error)

## 6. Simulations and Results

In order to verify the proposed methodology for parameter identification on non-linear robotic models the methodology was applied to the kinematic model of the robot arm show in Figures 1 and 2. For this the robot arm was simulated in a multi-physics software where each of the seven joints,  $\theta_i$  for  $i = 1$  to 7, comprising the robot arm was commanded to move following a sinusoidal function,  $\theta_i = \sin(time)$ . That is, the same function was applied to each of the seven joint actuators. The actuator for the gripper was not commanded to move as such joint has no effect on the

positioning of the end-effector and thus does not affect the parameter estimation process. In fact, when setting up the simulations, this aspect was checked by commanding the gripper joint to move and comparing the results.

The robot arm was initially positioned hanging from its base in a pendulum-like position (see time 0 insert in Fig. 3) and and commanded to move. The model was run for 10 seconds using a simulation step size of 0.1 sec. The arm performed 1.6 complete cycles of the commanded trajectory which was enough time to complete the parameter estimation process. Table 3 shows the values of the initial link parameters ( $d_{initial}$ ) and their estimated (real) values ( $d_{identified}$ ).

Table 3: The initial and identified values for the link parameters.

	Link 1 [m]	Link 2 [m]	Link 3 [m]	Link 4 [m]	Link 5 [m]	Link 6 [m]	Link 7 [m]
$d_{initial}$	0	0	0.3605	0	0.265	0	0
$d_{identified}$	0.0028	0.0002	0.3636	0.0015	0.2650	0.0015	0.0005

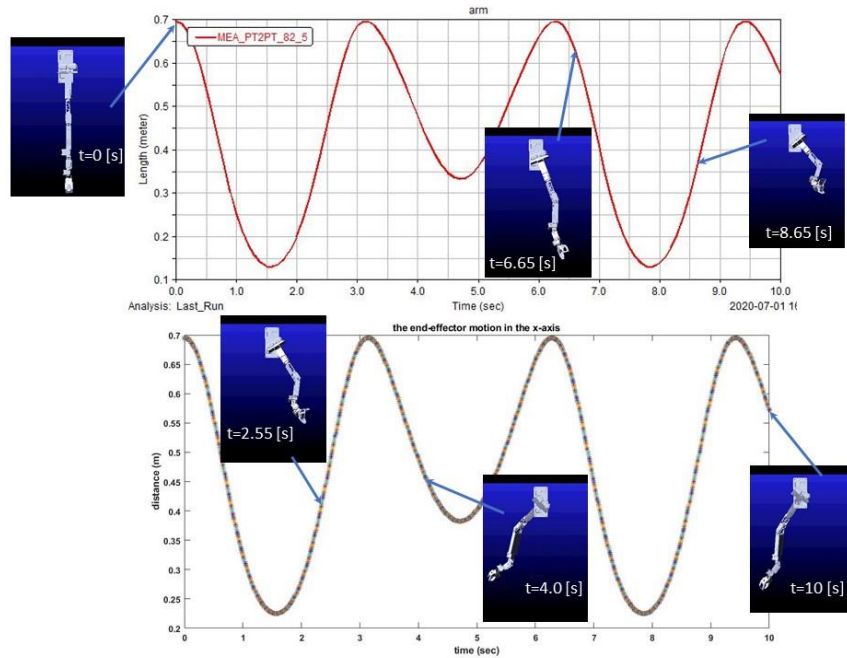
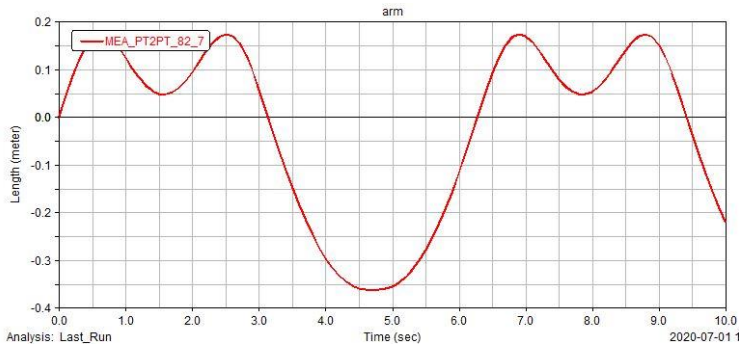
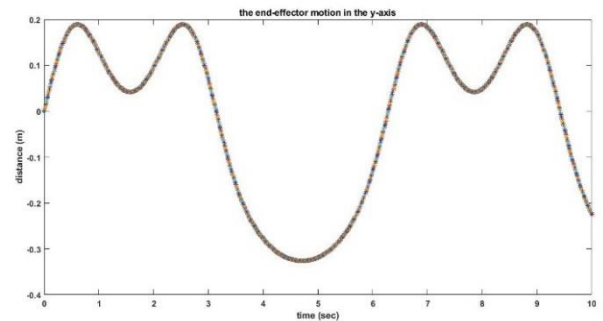


Fig. 3: Results for the end-effector's position in "x" axis: Top) Multi-body dynamic simulation, Bottom) Identified model.

During the tests, 14 robot arm poses were generated and imported into the simulation parameter calibration environment. Note that it was important to choose poses having no singularity conditions. After generating these 14 poses, the input data (i.e., joint angles) and output data (i.e., end-effector position in x, y, and z axes) was imported to the identification algorithm. Table 3 shows the initial and the identified values of the distance parameters,  $d$ . The results show that both outputs follow the same path with reasonable accuracy in the three axes of x, y, and z (as shown in Figures 3, 4, and 5, respectively). The slight difference between these two paths is due to the minimal errors which exist in the identified model (even though least squares method minimizes the square of errors, these errors cannot be zero in real case scenarios).

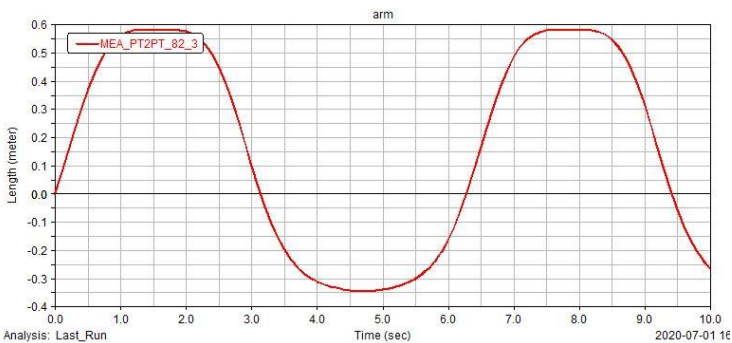


a) Multi-body dynamic simulation.

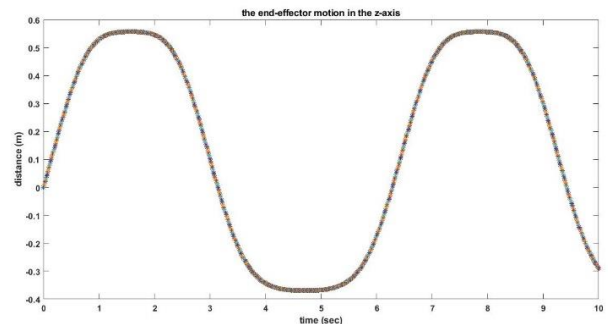


b) Identified model.

Fig. 4: Results for the end-effector position's in "y" axis.



a) Multi-body dynamic simulation.



b) Identified model.

Fig. 5: Results for the end-effector's position in "z" axis.



## 7. Conclusions

In this paper, the kinematic calibration of a 7-Dof humanoids' manipulator has been investigated. The main objective of the proposed parameter estimation algorithm was to address the errors which are created during the design and manufacturing aspects of robot arms. The results show that the method is successful in identifying kinematic parameters using the Gauss-Newton approach as part of the solution method in combination with the simulation environment as the measurement tool. This study corroborates that the model calibration is a powerful tool to address the problem of prevalent, inevitable errors in simulation environments. However, further work is required to test the performance of the method in real-world robots having higher complexities or being highly redundant. Under such robots the proposed approach might not converge to the solutions in one simulation run and might require several runs to deal with the redundancy. Another area for further progress is the application of the proposed method on identifying other parameters in the arm such as inertial parameters, sensors gain, and others. Although this paper focused on identifying the distance between joints, identifying other kinematic as well as time-varying parameters could be of interest.

## References

- [1] B. Hamner, S. Koterba, J. Shi, R. Simmons, and S. Singh, "An autonomous mobile manipulator for assembly tasks," *Autonomous Robots* 28, no. 1, pp. 131–149, 2010.
- [2] M. W. D. D. M. Lane and A. C. C. LEwards, "Hybrid position / force control of a hydraulic underwater manipulator," *IEE Proceedings-Control Theory and Applications*, vol 143, no. 2 pp. 145–151, 1996.
- [3] F. C. Samavati, M. Iranikhah, P. Dastango, M. Reza, and S. Alashti, "Mechanical Basic and Detailed Design for the Redundant Arm SAAM Applied on a Domestic Service Robot," *IEEE Artificial Intelligence and Robotics*, pp. 77–83, 2017.
- [4] V. Verdult, "Nonlinear System Identification : A State-Space Approach," 2002.
- [5] S. Haykin, "Neural networks, a comprehensive foundation second edition by Prentice-hall," pp. 2–4, 1999.
- [6] B. L. Van Overschee, P., & De Moor, *Subspace identification for linear systems: Theory—Implementation—Applications*. Springer Science & Business Media, 2012.
- [7] W. Suleiman, F. Kanehiro, K. Miura, and E. Yoshida, "Enhancing Zero Moment Point-Based Control Model : System Identification Approach Enhancing Zero Moment Point-Based Control Model:," *Advanced Robotics*, vol 25, no. 3-4, 2012.
- [8] S. Kolev and E. Todorov, "Physically consistent state estimation and system identification for contacts," *IEEE-RAS 15th International Conference on Humanoid Robots*, pp. 1036-1043. IEEE, pp. 1036–1043, 2015.
- [9] S. Besnard and W. Khalil, "Identifiable Parameters for Parallel Robots Kinematic Calibration," *IEEE International Conference on Robotics and Automation*, vol. 3, pp. 2859-2866, pp. 2859–2866, 2001.
- [10] A. Joubair and I. A. Bonev, "Kinematic calibration of a six-axis serial robot using distance and sphere constraints," *Journal of Advanced Manufacturing Technology*, vol 77, no. 1-4, pp. 515–523, 2015.
- [11] S. Hayati and M. Mirmirani, "Improving the Absolute Positioning Accuracy of Robot Manipulators," *Journal of Robotic Systems*, vol. 2, no. 4, pp. 397–413, 1985.
- [12] A. Nubiola, M. Slamani, A. Joubair, and I. A. Bonev, "Comparison of two calibration methods for a small industrial robot based on an optical CMM and a laser tracker," *Robotica*, vol. 32, no. 2014, pp. 447–466, 2020.
- [13] C. W. Wampler and J. M. Hollerbach, "An Implicit Loop Method for Kinematic Calibration and Its Application to Closed-Chain Mechanisms," *IEEE Transactions on Robotics and Automation*, vol. 11, no. 5, 1995.
- [14] J. P. Norton, *An introduction to identification*. Courier Corporation, 2009.