Development of a Franka Emika Cobot Simulator Platform (CSP) Dedicated to Medical Applications

Chapter · May 2021

DOI: 10.1007/978-3-030-75259-0_11

4 authors:

Amir Trabelsi
Université de Poitiers

Juan Sebastián Sandoval Arévalo
Université de Poitiers

Ghiss Moncef
Ecole Nationale d'Ingénieurs de Souss

med amine Laribi
Université de Poitiers

Development of a Franka Emika Cobot Simulator Platform (CSP) Dedicated to Medical Applications

Amir Trabelsi[1,2](✉), Juan Sandoval[2], Moncef Ghiss[1], and Med Amine Laribi[2]

[1] National Engineering School of Sousse, LMS, University of Sousse, BP 264 Sousse Erriadh, 4023 Sousse, Tunisia
{trabelsi.amir,moncef.ghiss}@eniso.u-sousse.tn

[2] Department of GMSC, Pprime Institute CNRS, ENSMA, University of Poitiers, UPR 3346, Poitiers, France
{juan.sandoval,med.amine.laribi}@univ-poitiers.fr

Abstract. Collaborative robots are gaining an important role in several medical applications. This kind of robot is suitable to physically interact with the medical staff in a secure shared workspace. Therefore, collaborative robots accompany medical staff and help them to improve their working conditions. Nonetheless, before deploying these robots, it is essential to validate their behavior for several simulated scenarios. The proposed study aims to build a simulation platform for a Franka Emika robot, recently employed as a medical assistant. This platform implements the kinematic and dynamic robot's model and allows studying its behavior in interaction with an external environment.

Keywords: Collaborative robots · Franka Emika · Robot dynamics · Medical robotics

1 Introduction

Collaborative robots, known as "cobots", are conceived to combine their safety, repeatability, and precision capabilities with human expertise and skills to perform complex tasks in a shared workspace. Thanks to these features, cobots are gaining great success, particularly in the medical field. In fact, several solutions currently exist to assist and help doctors to perform their medical gestures under improved working conditions, also reducing the execution time while gaining precision.

This is the case of recent works carried out by researchers of the CoBRA team at the Pprime Institute of the University of Poitiers, who propose the use of Franka Emika, a 7 degree of freedom (DoF) torque-controlled cobot, as a robotic assistant for medical applications. Some of the proposed applications concern minimally invasive surgery [1], Doppler sonography [2], or robotized craniotomy [3, 4].

Despite the experimental advances presented in [1–4], a simulation platform for the proposed robot is needed, allowing better planning of the medical tasks while examining

S. Zeghloul et al. (Eds.): RAAD 2021, MMS 102, pp. 95–103, 2021.
https://doi.org/10.1007/978-3-030-75259-0_11

the behavior of the cobot. The focus of this paper is to present a developed Franka Emika Cobot Simulation Platform (CSP) qualified to simulate the kinematic and dynamic behavior of the cobot in different medical applications. The simulator has been developed under Matlab Simulink and integrates realistic inertial parameters and a close estimation of joint frictions.

This paper details the steps followed to build the Franka Emika Cobot Simulator Platform (CSP). Section 2 presents the numerical modeling and the creation of a simulation platform by developing a kinematic model of a 7-DoF serial robot. Section 3 deals with the Franka Emika dynamic model, including the estimated inertial parameters of the robot links, as well as the joint friction parameters. Finally, a conclusion is provided about the obtained results.

2 Kinematic Modeling

The development of the kinematic model of the 7-DoF Franka Emika cobot begins with the identification of its geometric parameters based on the constructor's technical specifications [5] and the development of a mathematical model to control the actions of the articulated mechanical system.

Figure 1 illustrates the kinematic chain of the robot. It highlights the geometric model by developing the modified Denavit-Hartenberg (D-H) parameterization. This model allows the expression of the end-effector of the Franka robot's position and orientation. The D-H parameters of the Franka cobot are given in Table 1.

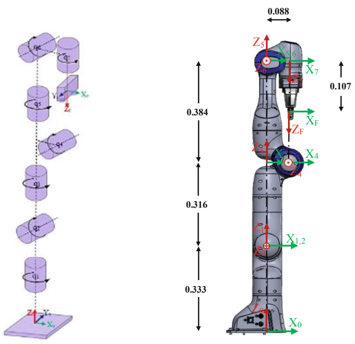

Fig. 1. Kinematic chain of the Franka Emika cobot.

To simulate the Franka Emika kinematic model, we use the Simscape Multibody toolbox of Matlab/Simulink. This latter presents a powerful tool for modeling rigid body mechanics. It is suitable for modeling the dynamics and kinematics of considerably complicated systems [6]. Firstly, we model the robot's bodies using Computer-Aided Design (CAD) software. In our case, we use SolidWorks software to upload the CAD files of the cobot and define a coordinate system for each cobot's link. Afterward, we redesign the surface shapes as volume parts and extract them as step files. Finally, we develop the mechanical cobot's chain in Simulink, as shown in Fig. 2.

Table 1. D-H parameters of the Franka Emika cobot.

Joint	q (rad)	(m)	(m)	(rad)
1	q_1	0.3330	0	0
2	q_2	0	0	$-\pi/2$
3	q_3	0.3160	0	$\pi/2$
4	q_4	0	0.0825	$\pi/2$
5	q_5	0.3840	−0.0825	$-\pi/2$
6	q_6	0	0	$\pi/2$
7	q_7	0	0.0880	$\pi/2$
Flange	0	0.1070	0	0

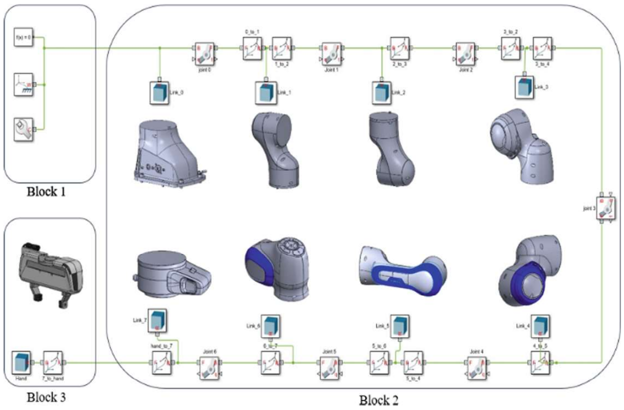

Fig. 2. Block diagram of the Franka Emika cobot in Simscape with the corresponding links.

To validate the simulated kinematic model, we compare its workspace to the one given by the constructor [7]. To do this, we proceed with the creation of an algorithm to generate the overall possible joint configurations without reaching the joint limits. The reachable positions for the end-effector are depicted in Fig. 3.

To properly exploit the robot simulated model, control inputs and feedback blocks are included in the model. As seen in Fig. 4, Box 1 presents the control inputs of the joint positions, Box 2 measures the current joint velocities whereas Box 3 implements a physical sensor measuring the current end-effector pose. Those values are exported to the main script implementing the control law.

After having validated the robot's motion in joint-space, we proceed to compute the Jacobian matrix allowing the mapping of the cartesian-space. In fact, the Jacobian matrix $\mathbf{J}(q) \in \Re^{6\times 7}$ defines the relationship between joint-space velocities $\dot{\mathbf{q}}_i \in \Re^7$ and the cartesian-space velocities $\mathbf{v} \in \Re^6$ as follows:

$$\mathbf{v} = \begin{bmatrix} \dot{\mathbf{p}} \\ w \end{bmatrix} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}}_i \tag{1}$$

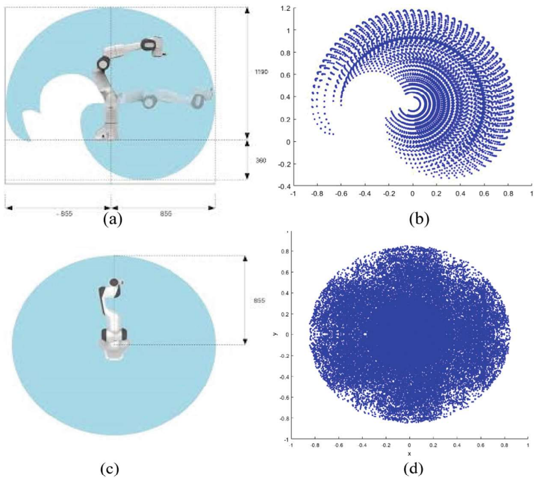

Fig. 3. Franka Emika workspace. Side-view (a) Given by the constructor (b) Obtained by simulation. Top-view (c) Given by the constructor (d) Obtained by simulation.

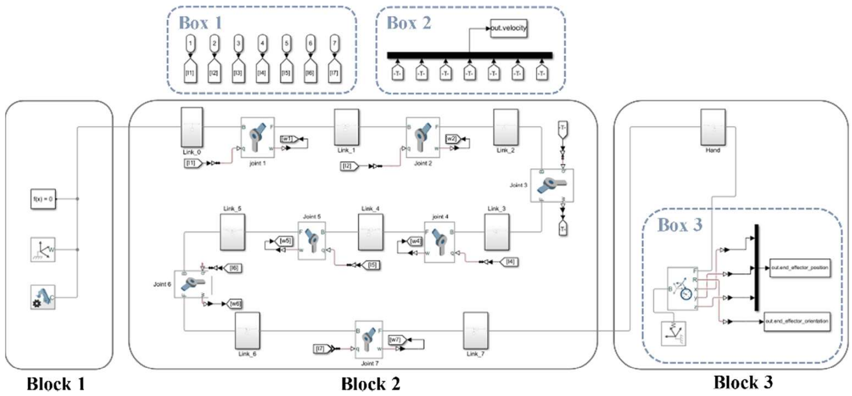

Fig. 4. Kinematic model implementation in Simulink.

where the cartesian-space velocity vector is composed by the end-effector linear $\dot{\mathbf{p}} \in \Re^3$ and angular $\mathbf{w} \in \Re^3$ velocities.

In addition to expressing the Jacobian analytically, we can express it through a geometric approach. In fact, analytical computing is complicated. The analytical method is developed by solving a sequence of linear equations deriving the geometric model. For this purpose, a different form of Jacobian expression, called the geometric Jacobian, was developed [8]. It is expressed as follows,

$$\mathbf{J} = \begin{bmatrix} \mathbf{J}_v \\ \mathbf{J}_w \end{bmatrix} = \begin{bmatrix} \mathbf{J}_{v1} & \cdots & \mathbf{J}_{v7} \\ \mathbf{J}_{w1} & \cdots & \mathbf{J}_{w7} \end{bmatrix} \tag{2}$$

where the first term $\mathbf{J}_v$ expresses the effect on linear velocity ($\dot{\mathbf{p}}$) and the second term $\mathbf{J}_w$ expresses the effect on the rotational velocity ($\boldsymbol{\omega}$). Since all the joints are revolute, $\mathbf{J}_{vi}$ and $\mathbf{J}_{wi}$ are computed as follows,

$$\begin{bmatrix} \mathbf{J}_{vi} \\ \mathbf{J}_{wi} \end{bmatrix} = \begin{bmatrix} {}^0\mathbf{z}_{i-1} * \left({}^0\mathbf{p}_7 - {}^0\mathbf{p}_{i-1}\right) \\ {}^0\mathbf{z}_{i-1} \end{bmatrix} \tag{3}$$

where ${}^0\mathbf{p}_7$ is the position of the end effector with respect to the base frame, ${}^0\mathbf{p}_{i-1}$ is the position of each revolute joint with respect to the base frame, and ${}^0\mathbf{z}_{i-1}$ is the joint axis vector of each revolute joint.

3 Dynamic Modeling

The implementation of a realistic dynamic model of the cobot is an essential feature to simulate the cobot behavior as close to reality as possible. Actually, this dynamic model is used to define the torque-control laws, for instance, to compensate the gravitational forces, thus, an inaccurate calculation of the dynamic model could cause unexpected results. As the Franka Emika robot is torque-controlled, we deal with the inverse dynamic model, where the current joint accelerations $\ddot{\mathbf{q}}$, velocities $\dot{\mathbf{q}}$, and positions $\mathbf{q}$ can be measured through the Box 2 presented in Fig. 4. The joint torques are then computed as follows,

$$\boldsymbol{\tau} = F^{-1}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) \tag{4}$$

The Newton-Euler method is used to solve the joint-space inverse dynamics. This algorithm calculates the applied and necessary joint torque variables to produce a given set of joints accelerations. The computation can be carried out in the four steps presented below [9].

The first step is to calculate the velocity $\mathbf{V}_i$ and acceleration $\mathbf{a}_i$ of each joint starting with the velocity and acceleration known from the fixed base. The velocity of any part of a kinematic tree can be recursively determined as the sum of the velocity of its previous joint and the velocity of the joint that connects it.

$$\mathbf{V}_i = \mathbf{V}_{i-1} + \mathbf{S}_i\dot{\mathbf{q}}_i \tag{5}$$

The recurrence relation for accelerations is obtained by differentiating this equation, resulting in,

$$\mathbf{a}_i = \mathbf{a}_{i-1} + \mathbf{S}_i\ddot{\mathbf{q}}_i + \dot{\mathbf{S}}_i\dot{\mathbf{q}}_i \tag{6}$$

The second step is to calculate the net force acting on the body (i). This force is related to the acceleration of body i by the equation of motion,

$$\mathbf{f}_i^a = \mathbf{I}\mathbf{a}_i + \mathbf{V}_i * \mathbf{I}_i\mathbf{V}_i \tag{7}$$

Where $\mathbf{I}_i$ is the inertia matrix of the i$^{\text{th}}$ link and $\mathbf{f}_i^a$ is the net force acting on the body (i).

The third step consists of calculating the force transmitted from link (i-1) to link (i) through the joint (i). Equation (8) gives the following recursive formula for the calculation of joint forces, where $\mathbf{f}_i^e$ is the external force acting on the body (i).

$$\mathbf{f}_i = \mathbf{f}_i^a - \mathbf{f}_i^e + \mathbf{f}_{i+1} \tag{8}$$

And the last step is to compute the torques $\boldsymbol{\tau}_i$, which are given by:

$$\boldsymbol{\tau}_i = \mathbf{S}_i^T \mathbf{f}_i \tag{9}$$

The implementation of the dynamic model involves the knowledge of the inertial parameters of the rigid body links of the cobot. Each rigid body is defined by ten inertial parameters: the mass, the coordinates of the center of mass, and the parameters of the symmetric inertial matrix. Moreover, since the requested parameters are typically not available, no information is given by the manufacturer about the dynamic parameters of the Franka Emika bodies, those parameters had to be estimated. Several research works show different methods allowing to identify them. For this work, we were based on two models for the Franka cobot: the inertial identification using Penalty-Based Optimization [10] and the classical inertial identification method (GAZEBO model) [11].

In order to evaluate the proximity of these models to the behavior of the real robot, we have implemented the inertial parameters resulted from both methods in the simulator and compared the torque signals with the ones produced by the real robot for the predefined joint motion trajectory shown in Fig. 5a. As illustrated by this figure, only the last three joints are excited. Moreover, the robot configuration illustrated in Fig. 5b. shows that joints 2, 3, 4, and 5 present the highest lever arm and support the heaviest loads. Therefore, we considered these joints for our comparison study.

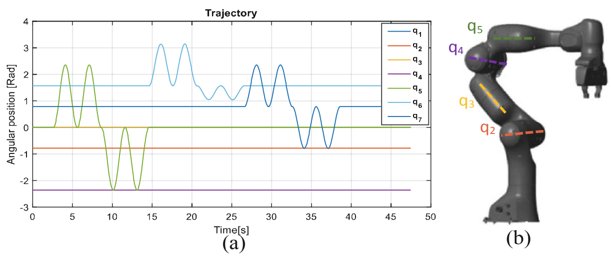

Fig. 5. The desired trajectory: (a) angular position versus time, (b) initial configuration.

The comparison between the simulated joint torques, obtained by both the identification and the GAZEBO methods, and the real joint torques is illustrated in Fig. 6. The obtained results allow concluding that the dynamic parameters obtained by the identification method are closer to the real case than the GAZEBO model. The offset shown in the simulation results, around 0.4 Nm compared to the real case, is due to the model uncertainties, particularly the unmodeled joint friction.

As proved above, the lack of knowledge of friction torques certainly alter the simulation results and disrupt the control laws. Therefore, we proceed to add the joint friction torques to the simulator to improve its performance.

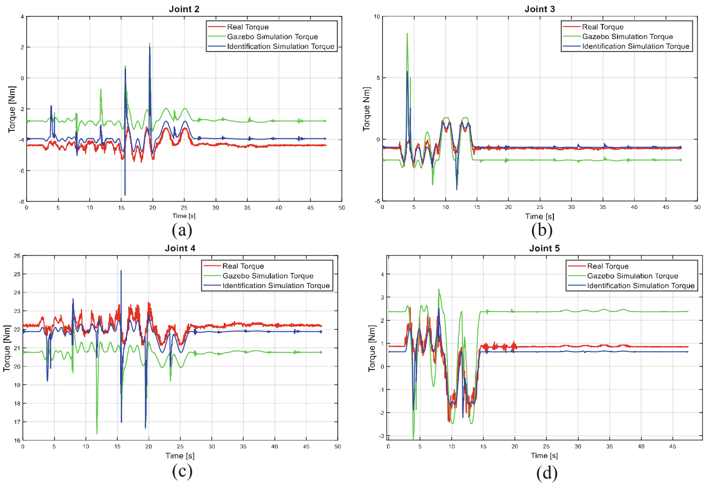

Fig. 6. Joint torques comparisons for (a) joint 2, (b) joint 3, (c) joint 4, and (d) joint 5.

Authors in [8] proposed a new method for the estimation of the joint friction torque vector $\boldsymbol{\tau}_f$. Joints positions and torques were obtained from the real robot following planned trajectories while carrying various payloads. The recorded data were concatenated to calculate the global least square estimation for the new friction parameters as illustrated in Table 2.

Table 2. Estimated joint friction parameters [10].

	Joint 1	Joint 2	Joint 3	Joint 4	Joint 5	Joint 6	Joint 7	Unit
φ_1	5.4615e−01	8.7224e+03	6.4068e−01	1.2794e +00	8.3904e−01	3.0301e−01	5.6489e−01	N.m
φ_2	5.1181e+00	9.0657e+00	1.0136e+01	5.5903e +00	8.3469e+00	1.7133e+01	1.0336e+01	s/rad
φ_3	1.0336e+01	2.5882e−02	−4.6070e−2	3.6194e−02	2.6226e−02	−2.1047e−2	3.5526e−03	rad/s

According to the proposed method in [8], the joints friction torque vector can be written as follows,

$$\boldsymbol{\tau}_{f,i} = \frac{\varphi_{1,i}}{1 + e^{-\varphi_{2,i}(\dot{q}_i + \varphi_{3,i})}} + \frac{\varphi_{1,i}}{1 + e^{-\varphi_{2,i} \cdot \varphi_{3,i}}} \tag{10}$$

The aforementioned joint friction model has been implemented in the proposed simulator. We generate several joint trajectories to estimate the joint frictions and verify the simulation model behavior. As expected, we observe that the friction is proportional to the joint velocity, as described in Eq. (10). For a first planned trajectory (Fig. 7), we excite the last three joints successively and estimate the different joint frictions in the time. Then, we try a more complex trajectory (Fig. 8) reaching the limits of the joint velocities and we notice that the friction torques do not exceed 0.8 Nm.

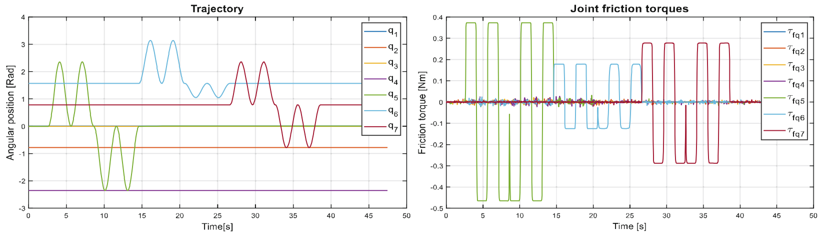

Fig. 7. Estimated joint friction torques for the first planned trajectory.

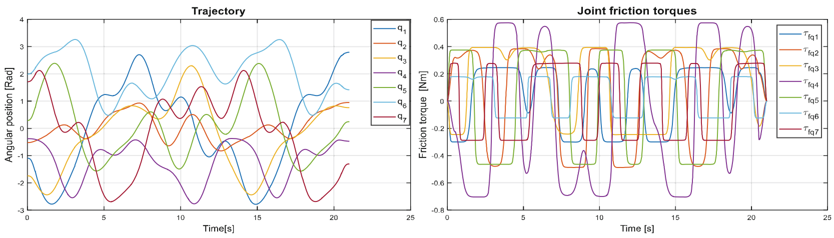

Fig. 8. Estimated joint friction torques for the second planned trajectory.

4 Conclusion

The aim of this research work was to depict the development of a Franka Emika simulation platform conceived to simulate medical applications. To achieve this goal, we first determined the kinematic model by developing modified D-H parameters and then computing the Geometric Jacobian matrix. To simulate the dynamic behavior of the robot, we developed the Inverse Dynamic Model using Newton Euler's recursive algorithm. Both models were implemented in Matlab Simulink to build a simulation platform using the Simscape Multibody Toolbox. Finally, we identified the joint's friction torques and validated our simulator by using several experimental trajectories.

The simulator is currently being used by researchers of the CoBRA Team at the Pprime Institute to validate several medical applications, as illustrated in Fig. 9 for the case of Doppler sonography.

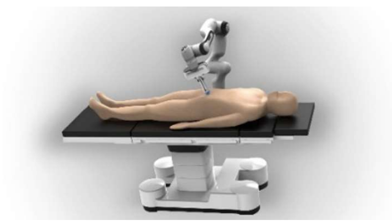

Fig. 9. Franka Emika cobot simulator platform.

References

1. Sandoval, J., Laribi, M.A., Zeghloul, S.: Autonomous robot-assistant camera holder for minimally invasive surgery. Mech. Mach. Sci. **78**, 465–472 (2020). https://doi.org/10.1007/978-3-030-30036-4_42
2. Sandoval, J., Laribi, M.A., Zeghloul, S., Arsicault, M.: On the design of a safe human-friendly teleoperated system for doppler sonography. Robotics (2019). https://doi.org/10.3390/robotics8020029
3. Essomba, T., Sandoval, J., Laribi, M.A.: Kinematic and force experiments on cadavers for the speci fi cation of a tele-operated craniotomy robot. In: Berns K., Görges D. (eds.) Advances in Service and Industrial Robotics. RAAD 2019. Advances in Intelligent Systems and Computing, vol. 980, pp. 447–454 (2019). Springer, Cham. https://doi.org/10.1007/978-3-030-19648-6_51
4. Zhan, Y., Duan, X.G.: Craniotomy robot system based on human-machine parallel collaboration. In: 2016 IEEE International Conference Mechatronics and Automation, ICMA 2016, pp. 1119–1124. IEEE (2016). https://doi.org/10.1109/ICMA.2016.7558719
5. Robot and interface specifications (2017). https://frankaemika.github.io/docs/control_parameters.html
6. Yang, Y., Cong, D.C., Han, J.W.: Elevator feel and centering unit simulation based on SimMechanics. Appl. Mech. Mater. **10-12**, 424-429 (2008). https://doi.org/10.4028/www.scientific.net/AMM.10-12.424
7. Performance S. Franka Emika PANDA - DATASHEET (2019). www.franka.de
8. Yazdanpanah, R.: Geometric Jacobians derivation and kinematic singularity analysis for smokie robot manipulator & the Barrett WAM. In: International Conference Robot Mechatronics (2017)
9. Featherstone, R.: Rigid Body Dynamics Algorithms. (2008). www.springer.com
10. Gaz, C., Cognetti, M., Oliva, A., Giordano, P.R., de Luca, A.: Dynamic identification of the Franka Emika Panda robot with retrieval of feasible parameters using penalty-based optimization. IEEE Robot. Autom. Lett. **4**(4), 4147–4154 (2019). https://doi.org/10.1109/LRA.2019.2931248
11. Gautier, M., Khalil, W.: Direct calculation of minimum set of inertial parameters of serial robots 11. IEEE Trans. Robot. Autom. **6**, 368–373 (1990). https://doi.org/10.1109/70.56655