Dynamic Identification of the Franka Emika Panda Robot With Retrieval of Feasible Parameters Using Penalty-Based Optimization – Supplementary material (revised version: October 15$^{\text{th}}$, 2019) –

Claudio Gaz[1] Marco Cognetti[2] Alexander Oliva[3] Paolo Robuffo Giordano[2] Alessandro De Luca[1]

***Abstract*— In this document, we report a number of supplementary materials to our RA-L paper [1].**

I. Joint friction parameters

Table I provides the numerical values of the parameters obtained with the fitting procedure described in Sec. IV.B of [1]. These parameters refer to the equations

$$\tau_{f,j} = \frac{\varphi_{1,j}}{1+e^{-\varphi_{2,j}(\dot{q}_j+\varphi_{3,j})}} - \frac{\varphi_{1,j}}{1+e^{-\varphi_{2,j}\varphi_{3,j}}}, \quad j \in [1,\ldots,7],$$

which model the friction acting at the robot joints.

TABLE I

Estimated joint friction parameters.

	Joint 1	Joint 2	Joint 3	Joint 4	Joint 5	Joint 6	Joint 7	units
φ_1	5.4615e-01	0.87224	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	3.9533e-02	2.5882e-02	-4.6070e-02	3.6194e-02	2.6226e-02	-2.1047e-02	3.5526e-03	rad/s

II. Comparison of dynamic coefficients

The manufacturer does not provide any information on the dynamic parameters of the Panda robot, and we have no ground truth values at our disposal for the dynamic coefficients (a.k.a., *base parameters*) that have to be estimated.

Nevertheless, we have compared the dynamic coefficients obtained with the reverse engineering approach in [1] with those of a classical identification method [2] that uses measurements of the joint positions and joint torques only. In this case, static positioning is no longer sufficient and the robot must execute sufficiently exciting trajectories in order to allow a reliable estimation of the dynamic coefficients. In particular, following [3], we designed the reference trajectories for the joints $j \in \{1, 7\}$ as

$$q_j(t) = \sum_{l=1}^{L} \frac{a_{l,j}}{l\omega_f} \sin\left(l\omega_f\, t\right) - \frac{b_{l,j}}{l\omega_f} \cos\left(l\omega_f\, t\right) + q_{0,j},$$

where $L = 5$ is the number of harmonics, $\omega_f = 0.15\pi$, and the coefficients $a_{l,j}$, $b_{l,j}$ and $q_{0,j}$ are reported in Table II. These trajectories have been commanded to the robot joints during 21 s, collecting a total of 20544 samples of positions and torques per joint.

TABLE II

Trajectory parameters for model identification of the Franka Emika Panda robot.

Joint j	1	2	3	4	5	6	7
$a_{1,j}$	-0.2031	-0.0699	0.3076	0.1269	-0.2773	0.2100	-0.2273
$a_{2,j}$	0.1295	-0.5380	0.5864	-0.0253	-0.0857	-0.1194	-0.5636
$a_{3,j}$	-0.0090	-0.2015	-0.4813	-0.2405	0.4810	-0.0950	-0.2099
$a_{4,j}$	0.2319	-0.5535	-0.1228	0.2178	0.5311	-0.0964	0.5725
$a_{5,j}$	-0.7598	0.2352	-0.5273	0.6984	-0.1639	-0.0399	-0.3748
$b_{1,j}$	-0.1136	-0.2437	0.3898	0.0401	0.5491	0.0692	0.2023
$b_{2,j}$	0.6600	0.1820	-0.0268	-0.1503	0.1046	-0.6946	-0.2570
$b_{3,j}$	-0.1858	-0.2390	0.0179	0.2359	0.0407	-0.4442	-0.5010
$b_{4,j}$	0.1867	-0.2647	0.4767	-0.2022	-0.0432	-0.5749	0.0774
$b_{5,j}$	0.2357	0.4252	0.2726	0.3693	0.2448	-0.0698	-0.4833
$q_{0,j}$	-0.5850	-0.1744	-0.3373	-1.8767	-1.0631	1.7917	-0.7284

Since we noticed already that joint friction is not negligible for the robot under study, we added in the robot dynamic model suitable friction functions to each joint $j \in \{1, 7\}$, which are linear in the three parameters $f_{v,j}$, $f_{c,j}$, and $f_{o,j}$ – see eq. (8) in [1]:

$$\tau_{f,j}(\dot{q}_j) = f_{v,j}\dot{q}_j + f_{c,j}\,\mathrm{sign}(\dot{q}_j) + f_{o,j}.$$

These functions may suffer from sudden discontinuities in the neighbourhood of $\dot{q}_j = 0$, $j = 1, \ldots, 7$, but are adopted here because of their linearity in the defining parameters.

The results of the identification process is summarized in Table III. The first column provides the symbolic expressions of the dynamic coefficients. The second and third columns show the values of the coefficients $\hat{\boldsymbol{\pi}}_{\mathrm{R-E}}$ identified using the reverse engineering approach and, respectively, their standard deviations $\%\sigma(\hat{\boldsymbol{\pi}}_{\mathrm{R-E}})$. The regressor condition number is 48.9, while the relative error percentage is 0.0315%. The fifth and sixth columns show the values of the coefficients $\hat{\boldsymbol{\pi}}_{\mathrm{CLS}}$ identified through the classical method with exciting trajectories and, respectively, their standard deviations $\%\sigma(\hat{\boldsymbol{\pi}}_{\mathrm{CLS}})$. In this case, the regressor condition number is 121.88, while the relative error percentage is 1.41%. Finally, the fourth column reports the numerical value of the dynamic coefficients $\boldsymbol{\pi}(\hat{\boldsymbol{p}})$ computed by substituting the values of the dynamic parameters retrieved with our framework [1] in their symbolic expressions. It can be noted that these values are slightly different from the original ones, showing that the ordinary least squares solutions $\hat{\boldsymbol{\pi}}_{\mathrm{R-E}}$ and $\hat{\boldsymbol{\pi}}_{\mathrm{CLS}}$ are usually not physically consistent [3].

[1] Dipartimento di Ingegneria Informatica, Automatica e Gestionale, Sapienza Università di Roma, via Ariosto 25, 00185 Roma, Italy. E-mail: {gaz,deluca}@diag.uniroma1.it.

[2] CNRS, Univ Rennes, Inria, IRISA, Rennes, France, E-mail: {marco.cognetti,prg}@irisa.fr.

[3] Inria, Univ Rennes, CNRS, IRISA, Rennes, France, E-mail: alexander.oliva@inria.fr.

TABLE III

Dynamic coefficients identified using our reverse engineering approach ($\hat{\boldsymbol{\pi}}_{\mathrm{R-E}}$) and using a classical approach ($\hat{\boldsymbol{\pi}}_{\mathrm{CLS}}$), with their standard deviations (in percentage). The coefficients $\boldsymbol{\pi}(\hat{\boldsymbol{p}})$ are computed from the retrieved dynamic parameters $\hat{\boldsymbol{p}}$. Dynamic coefficients $\hat{\boldsymbol{\pi}}_{\mathrm{OSI}}$ are obtained from a one-step identification procedure.

Dynamic coefficients	$\hat{\boldsymbol{\pi}}_{\mathrm{R-E}}$	$\%\sigma(\hat{\boldsymbol{\pi}}_{\mathrm{R-E}})$	$\boldsymbol{\pi}(\hat{\boldsymbol{p}})$	$\hat{\boldsymbol{\pi}}_{\mathrm{CLS}}$	$\%\sigma(\hat{\boldsymbol{\pi}}_{\mathrm{CLS}})$	$\hat{\boldsymbol{\pi}}_{\mathrm{OSI}}$
$J_{2yy} + J_{1zz}$	0.0292	0.3735	0.0373	0.0204	2.3609	0.0132
$J_{2xx}-J_{2yy}+J_{3yy}+0.0998m_3+0.632c_{3z}m_3+0.1067(m_3+m_4+m_5+m_6+m_7)$	0.9818	0.0176	0.9896	0.9188	0.1202	0.9485
J_{2xy}	-0.0052	1.5625	-0.0040	-0.0033	11.0392	0.0014
J_{2xz}	0.0284	0.2698	0.0103	0.0042	14.3570	7.4640×10^{-4}
J_{2yz}	-0.0035	2.3330	0.0008	-0.0155	2.3054	-0.0113
$J_{3yy} + J_{2zz} + 0.09985\ m_3 + 0.632\ c_{3z}\ m_3 + 0.1067\ (m_4 + m_5 + m_6 + m_7)$	1.0428	0.0091	1.0358	0.9346	0.0973	0.9491
$J_{3xx} - J_{3yy} + J_{4yy} - 0.0068\ m_4$	0.0106	1.2664	0.0116	0.0149	5.8149	-0.0164
$J_{3xy} + 0.0825\ c_{4z}\ m_4$	-0.0002	30.3622	----	0.0127	3.5669	-0.0020
J_{3xz}	-0.0104	0.4874	-0.0055	-0.0061	5.2129	0.0021
J_{3yz}	-0.0048	1.2379	-0.0044	-0.0138	2.3907	-0.0055
$J_{4yy} + J_{3zz} + 0.0068\ m_4 + 0.0136\ m_5 + 0.0136\ m_6 + 0.0136\ m_7$	0.1169	0.0691	0.1245	0.0825	0.5394	0.0922
$J_{4xx} - J_{4yy} + J_{5yy} + 0.768\ c_{5z}\ m_5 + 0.1406\ (m_5 + m_6 + m_7)$	0.5324	0.0202	0.5408	0.5462	0.0951	0.5381
$J_{4xy} + 0.0317\ m_5 + 0.0317\ m_6 + 0.0317\ m_7 + 0.0825\ c_{5z}\ m_5$	0.1501	0.0274	0.1388	0.1202	0.1665	0.1255
J_{xz4}	0.0048	0.9079	0.0039	0.0040	6.5285	-0.0020
J_{4yz}	-0.0027	1.6177	-0.0016	0.0078	3.4723	1.8047×10^{-4}
$J_{5yy} + J_{4zz} + 0.1543\ m_5 + 0.1543\ m_6 + 0.1543\ m_7 + 0.768\ c_{5z}\ m_5$	0.6369	0.0103	0.6325	0.6196	0.0829	0.6089
$J_{5xx} - J_{5yy} + J_{6yy} + 0.0077\ m_7$	0.0322	0.2739	-0.0075	0.0114	3.9131	0.0137
J_{5xy}	-0.0037	0.9578	-0.0015	0.0036	5.7718	2.3992×10^{-4}
J_{5xz}	-0.0061	0.4932	-0.0046	-0.0102	1.6764	-0.0117
J_{5yz}	0.0077	0.4345	0.0022	0.0080	2.7133	0.0077
$J_{6yy} + J_{5zz} + 0.0077\ m_7$	0.0181	0.3169	0.0276	0.0216	1.6395	0.0210
$J_{6xx} - J_{6yy} + J_{7yy} - 0.0077\ m_7$	-0.0065	1.0591	-0.0009	-0.0157	1.8290	-0.0097
$J_{6xy} + 0.088\ c_{7z}\ m_7$	0.0054	0.5331	0.0055	0.0014	8.9602	0.0032
J_{6xz}	-1.2901×10^{-5}	218.6430	----	0.0047	3.8644	0.0039
J_{6yz}	0.0008	3.3084	-0.0001	0.0006	27.1286	-1.0794×10^{-4}
$J_{7yy} + J_{6zz} + 0.0077\ m_7$	0.0250	0.1773	0.0304	0.0145	1.8227	0.0204
$J_{7xx} - J_{7yy}$	-0.0018	2.7378	0.0024	0.0024	7.3234	2.6087×10^{-4}
J_{7xy}	0.0012	2.0320	-0.0004	0.0049	1.8700	0.0011
J_{7xz}	0.0018	1.2299	-0.0017	0.0023	4.8889	0.0013
J_{7yz}	-0.0006	3.6756	-0.0005	-0.0027	4.7174	-0.0016
J_{7zz}	0.0013	2.6129	0.0049	-0.0043	4.4680	0.0022
$c_{2x}\ m_2$	-0.0054	0.1551	-0.0020	0.0288	1.8926	0.0280
$c_{2y}\ m_2 - 0.316\ m_4 - 0.316\ m_5 - 0.316\ m_6 - 0.316\ m_7 - 0.316\ m_3 - c_{3z}\ m_3$	-3.1026	0.0003	-3.1043	-3.1363	0.0105	-3.1353
$0.0825\ m_4 + 0.0825\ m_5 + 0.0825\ m_6 + 0.0825\ m_7 + c_{3x}\ m_3$	0.6874	0.0014	0.6842	0.6803	0.0481	0.6814
$c_{3y}\ m_3 - c_{4z}\ m_4$	0.0234	0.0353	0.0282	-0.0121	1.2201	-0.0103
$c_{4x}\ m_4 - 0.0825\ m_6 - 0.0825\ m_7 - 0.0825\ m_5$	-0.4888	0.0016	-0.4901	-0.4726	0.0368	-0.4742
$0.384\ m_5 + 0.384\ m_6 + 0.384\ m_7 + c_{4y}\ m_4 + c_{5z}\ m_5$	1.7185	0.0005	1.7207	1.7517	0.0193	1.7525
$c_{5x}\ m_5$	-0.0100	0.0644	-0.0147	-0.0005	26.5699	0.0015
$c_{5y}\ m_5 - c_{6z}\ m_6$	0.0771	0.0074	0.0824	0.0817	0.1362	0.0815
$0.088\ m_7 + c_{6x}\ m_6$	0.1657	0.0034	0.1650	0.1637	0.0744	0.1639
$c_{6y}\ m_6 - c_{7z}\ m_7$	-0.0677	0.0083	-0.0688	-0.0623	0.1527	-0.0613
$c_{7x}\ m_7$	0.0062	0.0744	0.0077	0.0013	6.5781	6.3142×10^{-4}
$c_{7y}\ m_7$	-0.0005	0.9490	-0.0031	0.0048	1.6783	0.0033
$f_{v,1}$				0.0665	1.9658	0.0628
$f_{v,2}$				0.1987	0.9615	0.2088
$f_{v,3}$				0.0399	3.4971	0.0361
$f_{v,4}$				0.2257	0.9124	0.2174
$f_{v,5}$				0.1023	1.5868	0.1021
$f_{v,6}$				-0.0132	12.2607	1.6128×10^{-4}
$f_{v,7}$				0.0638	2.1388	0.0632
$f_{c,1}$				0.2450	0.5889	0.2549
$f_{c,2}$				0.1523	0.9602	0.1413
$f_{c,3}$				0.1827	0.7324	0.1879
$f_{c,4}$				0.3591	0.4152	0.3625
$f_{c,5}$				0.2669	0.4668	0.2728
$f_{c,6}$				0.1658	0.8077	0.1529
$f_{c,7}$				0.2109	0.7055	0.2097
$f_{o,1}$				-0.1073	0.6932	-0.1069
$f_{o,2}$				-0.1566	2.9549	-0.1601
$f_{o,3}$				-0.0686	1.1566	-0.0718
$f_{o,4}$				-0.2522	0.9563	-0.2562
$f_{o,5}$				0.0045	19.4633	0.0079
$f_{o,6}$				0.0910	0.8390	0.0935
$f_{o,7}$				-0.0127	6.2628	-0.0070

On the other hand, in order to estimate a set of dynamic coefficients which are physically consistent, the following *one-step* approach can be used. It is possible, indeed, to change the term $\phi(\boldsymbol{p}_k)$ of the cost function in the Parameters Retrieval algorithm (29) in [1], as:

$$\phi(\boldsymbol{p}_k) = \left\|\overline{\boldsymbol{Y}}\boldsymbol{\pi}(\boldsymbol{p}_k) - \overline{\boldsymbol{\tau}}\right\|^2 ,$$

where $\overline{\boldsymbol{Y}}$ is the stacked regressor, $\overline{\boldsymbol{\tau}}$ is the stacked measurements vector and $\boldsymbol{\pi}(\boldsymbol{p}_k)$ is the coefficients vector computed from the current parameters vector $\boldsymbol{p}_k$. The drawback of the use of this function ϕ (with respect to the one reported in [1]) is that it is computed from the cumbersome stacked regressor, thus affecting the computational time. The identified dynamic coefficients $\hat{\boldsymbol{\pi}}_{\text{OSI}}$ obtained from the previously described one-step identification procedure are reported in the seventh column of Tab. III.

Note that the fourth column of Table III has two missing entries for the eight and twenty-fourth dynamic coefficients: these small coefficients are in fact discarded due to their large standard deviations (see the corresponding second column in the table).

In conclusion, the similarity between the results obtained with the classical method and with our approach indirectly confirms the validity of the set of dynamic coefficients in Table III for the Panda robot.

III. Comparison of Dynamic Parameters

The dynamic parameters of the Panda robot, i.e., the mass, center of mass, and inertia tensor of each link, have been retrieved by solving the nonlinear optimization problem presented in Sec. V of [1] as well as by using the LMI-SDP framework presented in [3], based on the Python code available at https://github.com/cdsousa/wam7_dyn_ident. This code has been slightly modified in order to include the triangular inequalities on the inertia tensors, according to [4]. The results are reported in the two Tables VIII and IX.

In Tab. VIII, the value of c_{1z} is reported as "$*$" (*don't care*) since it does not have any influence on the dynamics (i.e., it does not appear in the E-L model, and can take any value).

The used lower (LB) and upper (UB) bounds and the final value $\hat{\boldsymbol{p}}$ obtained with our optimization algorithm are shown in the second to fourth columns. The final parameters $\boldsymbol{\delta}$ (including 21 friction parameters) obtained with the LMI-SDP method are reported in the fifth columns. In particular, these latter values were retrieved from the dynamic coefficients $\boldsymbol{\beta}^{\star e}$ by means of eq. (49) in [3], using the same bounds (properly manipulated) that we adopted for our algorithm. The resulting $\boldsymbol{\delta}$ set was then slightly manipulated, in order to obtain for each link the center of mass from the its first moment of inertia and the barycentric inertia tensor from the inertia tensors w.r.t. the link frame.

Figure 1 shows a comparison between the torques measured by the joint torque sensors of the Panda robot during a validation trajectory and the estimated torques generated from a Newton-Euler (N-E) routine using the $\hat{\boldsymbol{p}}$ parameters and the $\boldsymbol{\delta}$ parameters. In order to further validate the parameter sets $\hat{\boldsymbol{p}}$ and $\boldsymbol{\delta}$, we also designed 10 new validation trajectories (each lasting for 10 seconds), performing the same joint torque comparisons and then computing the mean square error (MSE) for each trajectory. The concatenated joint trajectories are reported in Fig. 2, the concatenated joint torques in Fig. 3, and the total MSE in Table IV. These results confirm that both estimates are reliable and consistent.

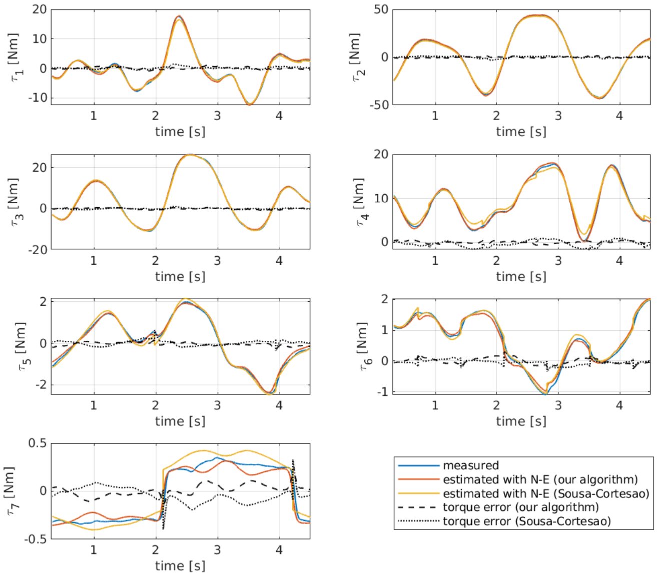

Fig. 1. Comparison between measured joint torques during a validation trajectory (blue solid lines) and estimated torques generated from a N-E routine using the $\hat{\boldsymbol{p}}$ parameters from our parameter retrieval method in [1] (solid red lines) and the parameters $\boldsymbol{\delta}$ from the LMI-SDP framework in [3] (solid yellow lines). The torque errors are reported as well.

TABLE IV
Mean Square Error (MSE) [(Nm)^2] of torque predictions using the parameters $\hat{\boldsymbol{p}}$ of our method [1] and the parameters $\boldsymbol{\delta}$ of LMI-SDP approach [3] in validation experiments of Fig. 2.

	Joint 1	**Joint 2**	**Joint 3**	**Joint 4**	**Joint 5**	**Joint 6**	**Joint 7**
MSE $\hat{\boldsymbol{p}}$	0.4068	1.0282	0.5951	0.4231	0.1749	0.0403	0.0267
MSE $\boldsymbol{\delta}$	0.4955	1.0155	0.2751	0.6627	0.0828	0.0508	0.0284

IV. Validation in V-REP

In order to validate the set of identified parameters reported in Tables VIII and IX, we built a robot model in V-REP [5], as shown in Figure 4. We adjusted the parameters for each link, in order to be compatible with the V-REP interface (e.g., for each link, the inertia tensor in V-REP has to be expressed w.r.t. the given CoM reference frame).

The validation process follows the idea that if the identification of the robot dynamic parameters is satisfactory, the measured torques on the real robot, apart from friction which is not simulated in V-REP, should match the ones provided

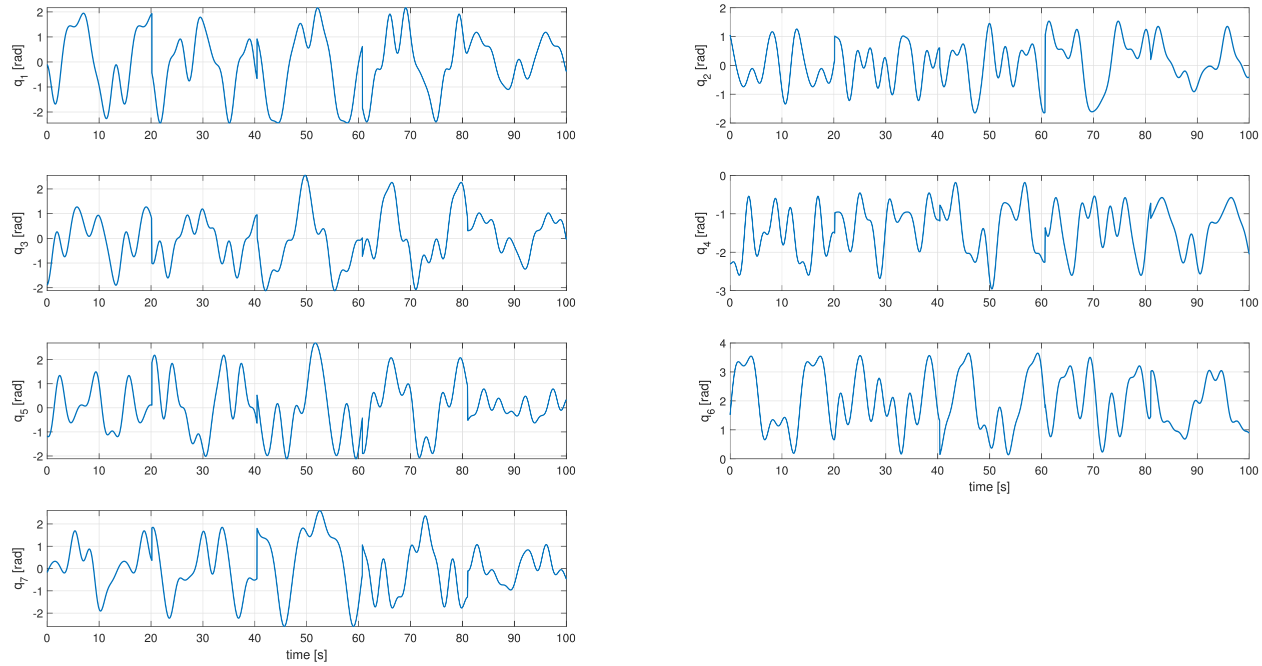

Fig. 2. Joint positions recorded during 10 validation trajectories (10 seconds each). The associated joint torques are reported in Fig. 3.

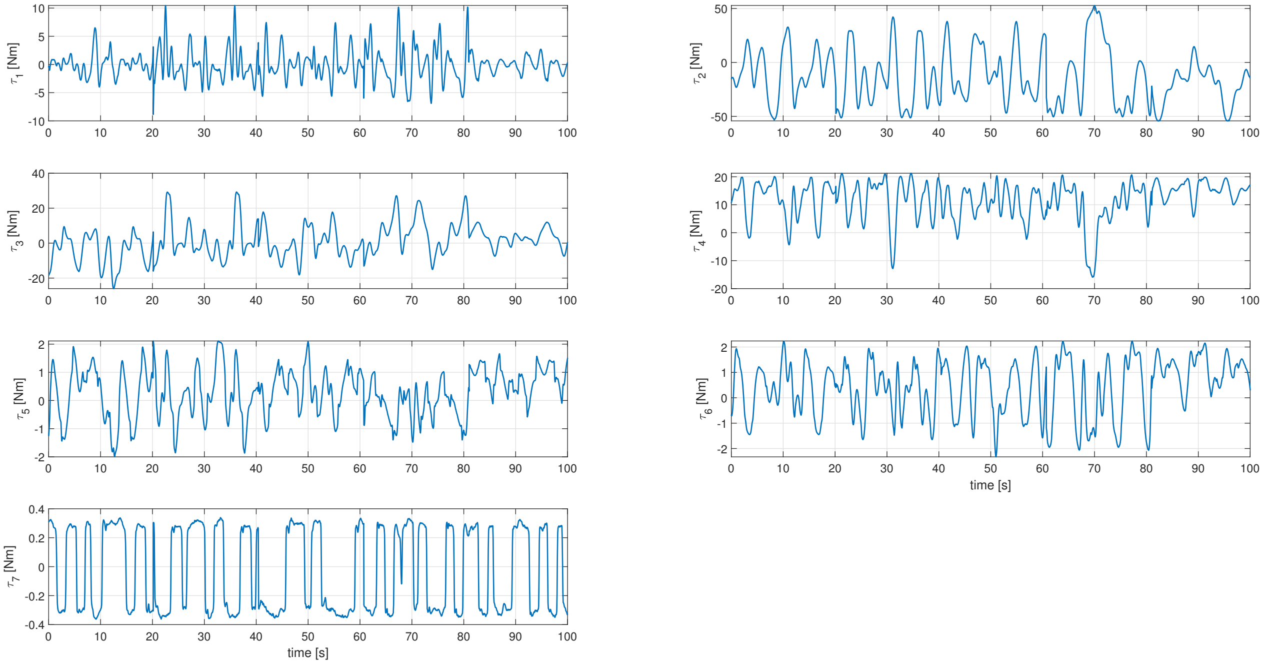

Fig. 3. Joint torques recorded during the 10 validation trajectories reported in Fig. 2.

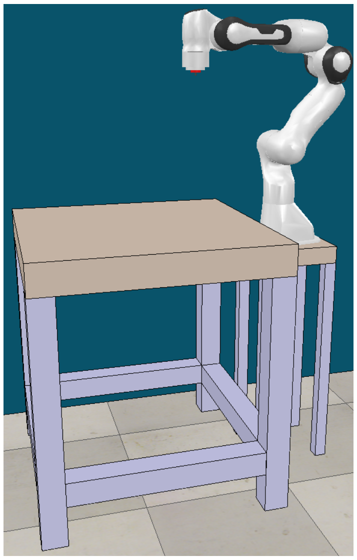

Fig. 4. The Franka Emika Panda robot in V-REP.

by the physics engines of the simulator when performing the same trajectory.

In particular, V-REP provides four different free licence physics engines. A performance comparison of these engines is reported in Figs. 5 and 6. Both the real and the simulated robot are commanded through sinusoidal joint velocity inputs (spanning their maximum available range). Figure 5 shows, for each joint, a filtered version of the measured torques of the real robot (solid blue lines) vs. those provided by the different physics engines (light blue: Bullet 2.78, orange: Bullet 2.83, yellow: ODE, and purple: Newton). As it can be seen, the simulated torques for all engines have a good overlap with the measured ones. Figure 6 shows the mean and the standard deviation of the torque errors for each joint. This error is defined as the difference between the measured torque on the real robot and the one generated from the simulation. We can notice that the mean torque errors are close to zero while the peaks of the standard deviations are due to measurement noise on the real robot torque readings and to numerical inaccuracies of the physics engines. In our experiments, all engines showed a good numerical performance, as confirmed by the values in Table V, and looking at the results in joint space, ODE (although it is the most noisy among the engines) and Bullet 2.83 seem to be the best performing engines in V-REP.

TABLE V

MEAN VALUES OF THE TORQUE ERRORS AT THE ROBOT JOINTS FOR EACH PHYSICS ENGINE AVAILABLE IN V-REP.

	Joint 1	**Joint 2**	**Joint 3**	**Joint 4**	**Joint 5**	**Joint 6**	**Joint 7**	**units**
Bullet 2.78	0.0554	1.7616	-0.7363	-0.0111	0.0735	0.0970	-0.0372	N·m
Bullet 2.83	0.0508	0.5735	-0.2617	-0.0844	0.0698	0.0628	-0.0386	N·m
ODE	0.0474	0.1782	-0.0986	-0.1493	0.0795	0.0397	-0.0386	N·m
Newton	0.1091	-0.8096	-0.1440	-1.4133	0.0989	0.0027	-0.0255	N·m

In order to confirm the quality of the proposed solution, we analyzed the performance also in Cartesian space. To this aim, we compared the 3D pose (as obtained from V-REP) of the end-effector of the simulated robot with the corresponding pose obtained by reading the joint positions from the real robot and passing them to the direct kinematics module. We defined then a position and an orientation error (following the Roll-Pitch-Yaw convention), whose mean values are reported in Fig, 7 for all physics engines available in V-REP. This comparison is shown also in the video accompanying [1].

The results support once again the quality of the estimated dynamic parameters, since the mean position error is below 1.5 cm and the mean orientation error is below 3 degrees for all the engines. Particularly good results are obtained with the ODE engine, with mean errors of less than 5 mm in position and below 1.2 deg in orientation. Moreover, the maximum errors for the worst performing physics engine (Bullet 2.78) are 9.13 cm in position and 3.14 deg in orientation.

V. USE OF OUR METHOD WITH NONLINEAR AND CONDITIONAL CONSTRAINTS

In order to retrieve a feasible set of dynamic parameters $\hat{\boldsymbol{p}}$ for the Panda robot, only linear constraints were used, since the non-convex shapes of the links would allow their centers of mass to be located even outside the link themselves. We present here a simple example of retrieval of dynamic parameters where the use of nonlinear constraints would be preferable, and which fits then in the framework of our method [1].

Consider the spatial 2R robot with orthogonal joint axes in Fig. 8. Its (standard) Denavit-Hartenberg kinematic parameters are reported in Table VI. The robot body consists of two links: link 1 is a cylinder of height $l_1 = 1.5$ m and radius $r_1 = 0.15$ m; link 2 has a total length of $l_2 = 0.8$ m and is composed by a cylinder of length $l_2 - l_2'$ and radius $r_2 = 0.08$ m, connected at the end with a truncated cone of length $l_2' = 0.2$ m and with radius $r_3 = 0.02$ m for its smaller base.

TABLE VI

DH TABLE OF THE SPATIAL 2R ROBOT.

i	a_i	α_i	d_i	θ_i
1	0	$\pi/2$	l_1	q_1
2	l_2	0	0	q_2

The assumed ground truth values of the dynamic parameters $\boldsymbol{p}_{\text{real}}$ are given in the fourth column of Table VII.

The symbolic form of the dynamic coefficients $\boldsymbol{\pi}_{2\text{R}} \in \mathbb{R}^8$ of this robot is:

$$\boldsymbol{\pi}_{2\text{R}} = \begin{pmatrix} J_{1yy} + J_{2yy} - 0.64 m_2 \\ J_{2xx} - J_{2yy} + 0.64 m_2 \\ J_{2xy} \\ J_{2xz} - 0.8 c_{2z} m_2 \\ J_{2yz} \\ J_{2zz} - 0.8 m_2 \\ 0.8 m_2 + c_{2x} m_2 \\ c_{2y} m_2 \end{pmatrix}.$$

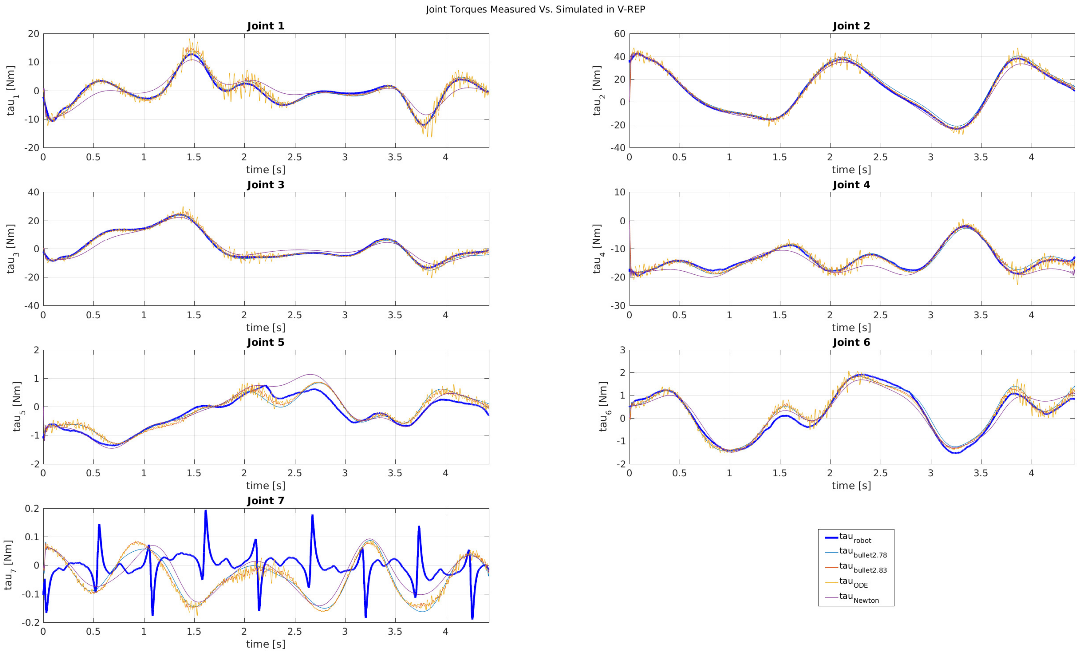

Fig. 5. Comparison between the torques generated by the real robot during an experiment (blue) and those obtained from the simulated Panda robot in V-REP using different physics engines (Bullet 2.78, Bullet 2.83, ODE, Newton).

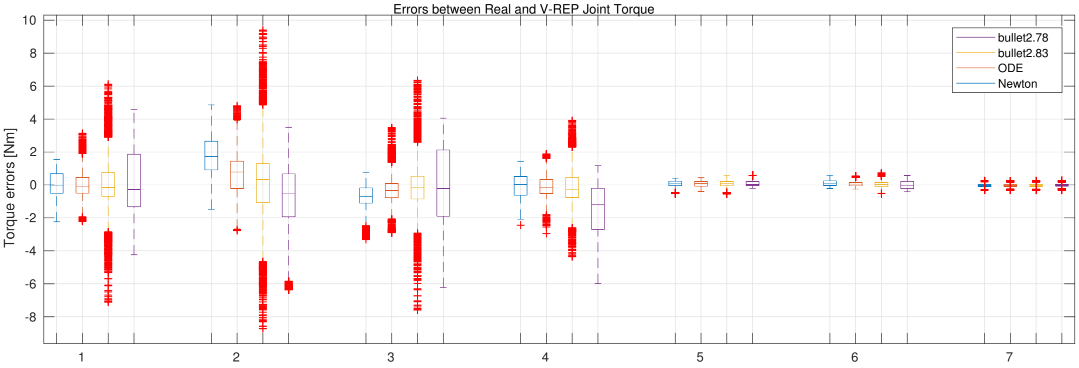

Fig. 6. Torque errors (w.r.t. the real robot) of the physics engines Bullet 2.78, Bullet 2.83, ODE, and Newton in V-REP during an experiment with the Panda robot for each of its 7 joints. The boxes indicate mean values of the errors while the bars denote their standard deviations.

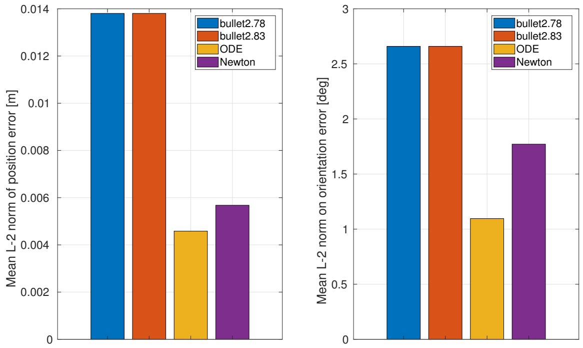

Fig. 7. The position (left) and orientation (right) mean error for the different physics engines available in V-REP. While all the physics engines show a good performance, ODE seems to outperform the others.

In order to estimate the values of the dynamic coefficients $\boldsymbol{\pi}_{2\mathrm{R}}$, a numerical simulation is performed using $\boldsymbol{p}_{\mathbf{real}}$ and the joint torques are recorded during an exciting motion. Then the dynamic coefficients are properly estimated with an ordinary least squares method.

At this stage, the algorithm for the retrieval of the dynamic parameters $\hat{\boldsymbol{p}}$ is launched, using the bounds on physical feasibility reported in Table VII, upper and lower bounds on the total mass, $1 \leq m_1 + m_2 \leq 15$, and linear constraints on the inertia tensors (exploiting the triangular inequality). When using box constraints on the position of the center of mass of each of the two links, the solution would be searched in an extra volume (i.e., a parallelepiped) which is $V_{\mathrm{link1}} = 4/\pi \simeq 1.27$ times larger than the volume of the cylindric link 1 and $V_{\mathrm{link2}} = 4/\pi\,(1 + 2l_2'/l_2) \simeq 2$ times larger than the one of link 2. On the other hand, considering that the distance of the center of mass from the major link axis must be less than its radius, the previous (approximate)

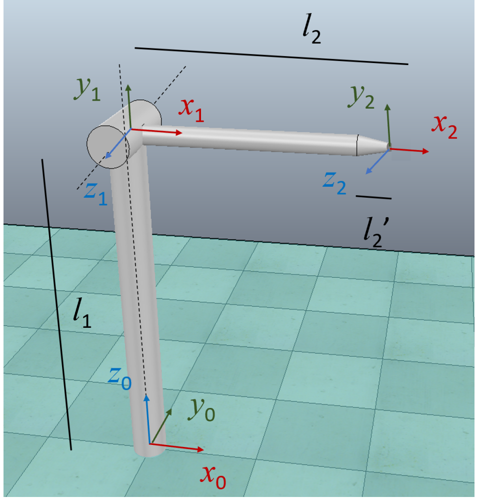

Fig. 8. A spatial 2R robot with joints along axes z_0 and z_1.

TABLE VII
LOWER AND UPPER BOUNDS FOR THE DYNAMIC PARAMETERS OF THE 2R ROBOT IN FIG. 8, WITH REAL $\boldsymbol{p}_{\text{real}}$ AND ESTIMATED $\hat{\boldsymbol{p}}$ VALUES.

parameter	LB	UB	$\boldsymbol{p}_{\text{real}}$	$\hat{\boldsymbol{p}}$	units
m_1	0	10	8	4.5788	kg
m_2	0	5	3	3.7828	kg
c_{1x}	$-\infty$	∞	0.01	-0.016124	m
c_{1y}	-1.42	0.08	-0.75	-0.74408	m
c_{1z}	$-\infty$	∞	-0.02	-3.8e-04	m
c_{2x}	-0.8	0	-0.1	-0.24487	m
c_{2y}	$-\infty$	∞	0	6.16e-06	m
c_{2z}	$-\infty$	∞	0	-3.66e-03	m
I_{1xx}	0	1	0.03	0.61296	kg·m^2
I_{1xy}	-1	1	0	0.071738	kg·m^2
I_{1xz}	-1	1	0	-0.0099797	kg·m^2
I_{1yy}	0	1	0.03	0.022982	kg·m^2
I_{1yz}	-1	1	0	0.031154	kg·m^2
I_{1zz}	0	1	0.01	0.6172	kg·m^2
I_{2xx}	0	1	0.01	0.019819	kg·m^2
I_{2xy}	-1	1	0	-4.525e-06	kg·m^2
I_{2xz}	-1	1	0	-0.0077412	kg·m^2
I_{2yy}	0	1	0.04	0.35386	kg·m^2
I_{2yz}	-1	1	0	-5.9342e-05	kg·m^2
I_{2zz}	0	1	0.04	0.34412	kg·m^2

box constraints on the center of mass of the two links can be replaced by the following (exact) nonlinear and conditional (if-else) constraints g_{CoM_1} and g_{CoM_2}, to be used within our core optimization **Algorithm 1** in [1]:

$$\alpha_1 = r_1 - \sqrt{c_{1y}^2 + c_{1z}^2},$$

$$\alpha_2 = \begin{cases} r_2 - \sqrt{c_{2y}^2 + c_{2z}^2}, & \text{if } c_{2x} < -l_2' \\ \left(r_2 + \frac{r_3 - r_2}{l_2'}(l_2' + c_{2x})\right) - \sqrt{c_{2y}^2 + c_{2z}^2}, & \text{otherwise,} \end{cases}$$

$$g_{\text{CoM}_1} = -\min\{0, \alpha_1\}, \quad g_{\text{CoM}_2} = -\min\{0, \alpha_2\}.$$

Taking advantage of the knowledge of the link shapes, these nonlinear constraints ensure that each center of mass will lie inside the corresponding link shape but that no feasible position is being excluded, thus yielding a complete solution. A validation test was finally performed, comparing the joint torques on sinusoidal trajectories that are computed using the real parameters with the joint torques estimated by means of a N-E routine fed with the parameters $\hat{\boldsymbol{p}}$ reported in the fifth column of Table VII. The retrieved solution generates just the same dynamics as obtained with the real parameters.

Another interesting quantity that can be estimated only when using the N-E inverse dynamics algorithm is the wrench acting at the robot joint level, i.e., the exchanged forces/moments between two successive links connected by a joint. To this aim, it is very important to retrieve a set of dynamic parameters that is as close as possible to the real one. To show this, we relaxed the constraints in order to obtain a second solution set of dynamic parameters $\hat{\boldsymbol{p}}'$, which, however, contains also non-feasible parameters (i.e., parameters that have no physical meaning).

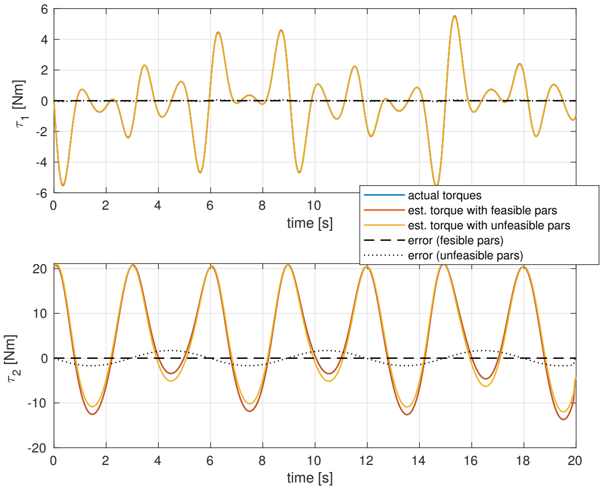

Fig. 9. Validation of estimated parameters in Table VII. A N-E routine fed with the feasible parameters $\hat{\boldsymbol{p}}$ and with some unfeasible parameters $\hat{\boldsymbol{p}}'$ returns similar estimated torques (respectively, red and yellow lines) along the motion. When comparing these estimates with the actual torques recorded along a sinusoidal joint trajectory (blue lines, almost overlapped by the red line), we notice that both sets provide a good torque estimation.

In Fig. 9, the yellow lines represent the estimated joint torques during a validation experiment coming from a N-E routine that is being fed with the unfeasible parameters $\hat{\boldsymbol{p}}'$. Despite of this, reliable motion torque estimations can still be appreciated. On the contrary, a strong inconsistency

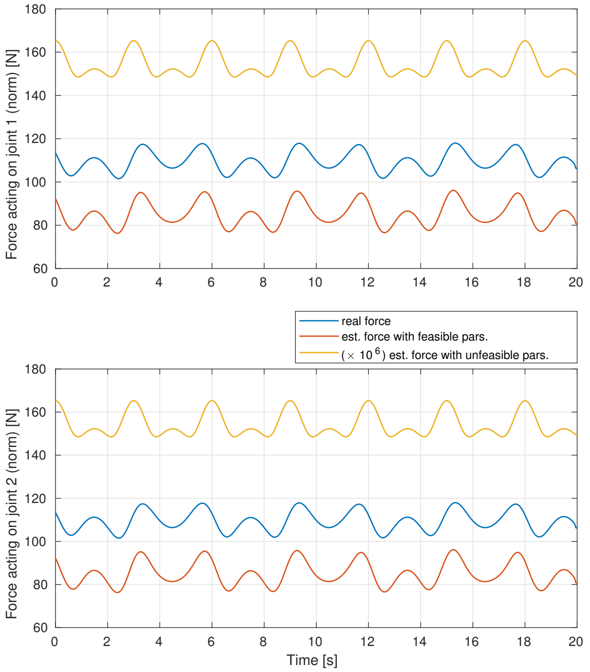

Fig. 10. Internal forces acting on the joints of the simulated 2R robot during a sinusoidal joint trajectory. Estimates of the internal forces obtained from a N-E routine are significantly closer to the real internal forces (blue lines) if feasible (i.e., physically consistent) dynamic parameters are provided to the routine, while estimates obtained from an unfeasible set $\hat{\boldsymbol{p}}'$ can be even 10^6 times larger (!) than real internal forces (note that the yellow line was scaled for the ease of reading).

may be observed when estimating the internal forces acting on the joints, obtained as a byproduct of the standard N-E algorithm. As shown by the values of these estimated forces in Fig. 10, it can be seen that estimations obtained from the feasible parameters $\hat{\boldsymbol{p}}$ (red lines) are close to the real forces (blue lines), while estimations retrieved from the unfeasible parameters $\hat{\boldsymbol{p}}'$ (yellow lines) are almost 10^6 times larger than the real values "felt" at the joints. This confirms that a physically-consistent solution which is close to the real one is mandatory for estimating the joint wrenches.

References

[1] C. Gaz, M. Cognetti, A. Oliva, P. Robuffo Giordano, and A. De Luca, "Dynamic identification of the Franka Emika Panda robot with retrieval of feasible parameters using penalty-based optimization," *IEEE Robotics and Automation Lett.*, 2019.

[2] M. Gautier and W. Khalil, "Direct calculation of minimum set of inertial parameters of serial robots," *IEEE Trans. on Robotics and Automation*, vol. 6, no. 3, pp. 368–373, 1990.

[3] C. Sousa and R. Cortesão, "Physical feasibility of robot base inertial parameter identification: A linear matrix inequality approach," *Int. J. of Robotics Research*, vol. 33, no. 6, pp. 931–944, 2014. [Online]. Available: https://doi.org/10.1177/0278364913514870

[4] ——, "Inertia tensor properties in robot dynamics identification: A linear matrix inequality approach," *IEEE/ASME Transactions on Mechatronics*, vol. PP, pp. 1–1, 01 2019.

[5] Coppelia Robotics. (2015) V-REP virtual robot experimentation platform. [Online]. Available: http://www.coppeliarobotics.com

TABLE VIII

Lower and upper bounds for masses and centers of mass of the Panda robot and their corresponding estimated values using our approach ($\hat{\boldsymbol{p}}$) and the one in [3] ($\boldsymbol{\delta}$).

parameter	**LB**	**UB**	$\hat{p}$	$\boldsymbol{\delta}$ **(from [3])**	**units**
m_1	0	10	4.970684	2.0643e-05	kg
m_2	0	10	0.646926	9.2017	kg
m_3	0	10	3.228604	1.4575	kg
m_4	0	10	3.587895	4.5856	kg
m_5	0	10	1.225946	0.5231	kg
m_6	0	10	1.666555	2.0604	kg
m_7	0	10	7.35522e-01	0.1718	kg
c_{1x}	-0.05	0.05	3.875e-03	5.202e-08	m
c_{1y}	-0.05	0.05	2.081e-03	5.202e-08	m
c_{1z}	-0.4	0.05	*	-0.1750	m
c_{2x}	-0.05	0.05	-3.141e-03	0.0015	m
c_{2y}	-0.15	0.05	-2.872e-02	-0.0578	m
c_{2z}	-0.05	0.05	3.495e-03	-0.0384	m
c_{3x}	-0.05	0.15	2.7518e-02	0.1096	m
c_{3y}	-0.05	0.05	3.9252e-02	0.05	m
c_{3z}	-0.1	0.05	-6.6502e-02	-0.1	m
c_{4x}	-0.15	0.05	-5.317e-02	-0.0485	m
c_{4y}	-0.05	0.15	1.04419e-01	0.15	m
c_{4z}	-0.05	0.05	2.7454e-02	0.0147	m
c_{5x}	-0.05	0.05	-1.1953e-02	-0.004	m
c_{5y}	-0.05	0.05	4.1065e-02	0.045	m
c_{5z}	-0.05	0.05	-3.8437e-02	-0.05	m
c_{6x}	-0.05	0.15	6.0149e-02	0.0732	m
c_{6y}	-0.05	0.05	-1.4117e-02	-0.0251	m
c_{6z}	-0.05	0.05	-1.0517e-02	-0.0276	m
c_{7x}	-0.05	0.05	1.0517e-02	-0.0326	m
c_{7y}	-0.05	0.05	-4.252e-03	0.0087	m
c_{7z}	0.04	0.15	6.1597e-02	0.0396	m

TABLE IX

Lower and upper bounds for the inertia tensor elements of the Panda robot and their corresponding estimated values using our approach ($\hat{\boldsymbol{p}}$) and the one in [3] ($\boldsymbol{\delta}$).

parameter	**LB**	**UB**	$\hat{p}$	$\boldsymbol{\delta}$ **(from [3])**	**units**
I_{1xx}	0	1	7.0337e-01	0.6	kg·m^2
I_{1xy}	-1	1	-1.3900e-04	-3.2497e-14	kg·m^2
I_{1xz}	-1	1	6.7720e-03	-6.4280e-09	kg·m^2
I_{1yy}	0	1	7.0661e-01	0.6	kg·m^2
I_{1yz}	-1	1	1.9169e-02	-6.4280e-09	kg·m^2
I_{1zz}	0	1	9.1170e-03	2.0353e-06	kg·m^2
I_{2xx}	0	1	7.9620e-03	0.028	kg·m^2
I_{2xy}	-1	1	-3.9250e-03	5.5650e-05	kg·m^2
I_{2xz}	-1	1	1.0254e-02	0.0056	kg·m^2
I_{2yy}	0	1	2.8110e-02	0.0291	kg·m^2
I_{2yz}	-1	1	7.0400e-04	-2.7636e-04	kg·m^2
I_{2zz}	0	1	2.5995e-02	0.0011	kg·m^2
I_{3xx}	0	1	3.7242e-02	2.0154e-06	kg·m^2
I_{3xy}	-1	1	-4.7610e-03	-5.9144e-09	kg·m^2
I_{3xz}	-1	1	-1.1396e-02	1.1523e-08	kg·m^2
I_{3yy}	0	1	3.6155e-02	2.0243e-06	kg·m^2
I_{3yz}	-1	1	-1.2805e-02	6.0956e-09	kg·m^2
I_{3zz}	0	1	1.0830e-02	2.0164e-06	kg·m^2
I_{4xx}	0	1	2.5853e-02	2.0242e-06	kg·m^2
I_{4xy}	-1	1	7.7960e-03	7.1405e-09	kg·m^2
I_{4xz}	-1	1	-1.3320e-03	8.8917e-10	kg·m^2
I_{4yy}	0	1	1.9552e-02	2.0036e-06	kg·m^2
I_{4yz}	-1	1	8.6410e-03	-2.2039e-09	kg·m^2
I_{4zz}	0	1	2.8323e-02	2.0260e-06	kg·m^2
I_{5xx}	0	1	3.5549e-02	2.0057e-06	kg·m^2
I_{5xy}	-1	1	-2.1170e-03	1.6453e-10	kg·m^2
I_{5xz}	-1	1	-4.0370e-03	-3.6824e-10	kg·m^2
I_{5yy}	0	1	2.9474e-02	2.0037e-06	kg·m^2
I_{5yz}	-1	1	2.2900e-04	2.3080e-09	kg·m^2
I_{5zz}	0	1	8.6270e-03	2.0028e-06	kg·m^2
I_{6xx}	0	1	1.9640e-03	2.0026e-06	kg·m^2
I_{6xy}	-1	1	1.0900e-04	1.5823e-09	kg·m^2
I_{6xz}	-1	1	-1.1580e-03	2.0530e-09	kg·m^2
I_{6yy}	0	1	4.3540e-03	2.0070e-06	kg·m^2
I_{6yz}	-1	1	3.4100e-04	-4.9387e-10	kg·m^2
I_{6zz}	0	1	5.4330e-03	2.0073e-06	kg·m^2
I_{7xx}	0	1	1.2516e-02	2.0026e-06	kg·m^2
I_{7xy}	-1	1	-4.2800e-04	2.6159e-10	kg·m^2
I_{7xz}	-1	1	-1.1960e-03	1.4055e-09	kg·m^2
I_{7yy}	0	1	1.0027e-02	2.0037e-06	kg·m^2
I_{7yz}	-1	1	-7.4100e-04	-4.2047e-10	kg·m^2
I_{7zz}	0	1	4.8150e-03	2.0020e-06	kg·m^2