

Copyright Notice

©2021 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

Calibration of an Elastic Humanoid Upper Body and Efficient Compensation for Motion Planning

Johannes Tenhumberg and Berthold Bäuml

Abstract—High absolute accuracy is an essential prerequisite for a humanoid robot to autonomously and robustly perform manipulation tasks while avoiding obstacles. We present for the first time a kinematic model for a humanoid upper body incorporating joint and transversal elasticities. These elasticities lead to significant deformations due to the robot’s own weight, and the resulting model is implicitly defined via a torque equilibrium. We successfully calibrate this model for DLR’s humanoid Agile Justin, including all Denavit–Hartenberg parameters and elasticities. The calibration is formulated as a combined least-squares problem with priors and based on measurements of the end effector positions of both arms via an external tracking system. The absolute position error is massively reduced from 21 mm to 3.1 mm on average in the whole workspace. Using this complex and implicit kinematic model in motion planning is challenging. We show that for optimization-based path planning, integrating the iterative solution of the implicit model into the optimization loop leads to an elegant and highly efficient solution. For mildly elastic robots like Agile Justin, there is no performance impact, and even for a simulated highly flexible robot with 20 times higher elasticities, the runtime increases by only 30%.

I. INTRODUCTION

A prerequisite for a humanoid robot to autonomously and robustly perform tasks in the real world is an accurate kinematic model of itself. For grasping, the robot needs to precisely position its tool center point (TCP) relative to objects, and when planning collision-free motions in self-acquired 3D models of the environment [1] the extension of the whole body has to be predicted correctly.

Humanoid robots are complex mechatronical systems built from lightweight components, which often leads to a significant deviation from a straightforward, so-called, *geometric* kinematics and make it necessary to model *non-geometric* effects like elasticities. For our advanced humanoid robot DLR Agile Justin [2], e.g., the deviation from the geometric kinematics leads to a mean error of 21 mm and a worst-case error as large as 61 mm in the whole workspace rendering robust autonomous action almost impossible.

In this paper, we present a kinematic model incorporating joint elasticities and transversal elasticities – to our knowledge for the first time for a complete humanoid upper body. We provide a clear and concise derivation of the kinematic model from the torque equilibrium, explicitly state the resulting exact implicit equation, and show how it can be solved

The authors are with the Institute of Robotics and Mechatronics, German Aerospace Center (DLR), Münchenerstr. 20, Wessling, 82234, Germany {johannes.tenhumberg, berthold.baeuml}@dlr.de
This work was partly funded by the Bavarian Ministry of Economic Affairs, Regional Development and Energy, within the projects SMiLE (LABAY97) and SMiLE2gether (LABAY102).

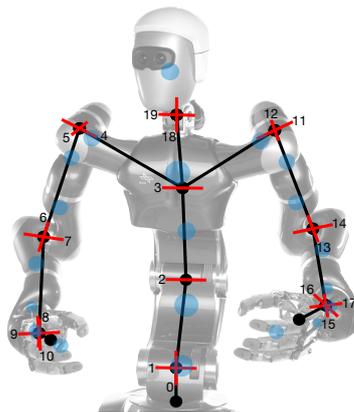


Fig. 1. Kinematic tree structure of the humanoid Agile Justin, showing its 19 degrees of freedom (red) and the mass model used for calibration (blue)

iteratively. We successfully calibrate all Denavit–Hartenberg (DH) parameters and elasticities of this model for the humanoid Agile Justin and discuss contributions of the components of the kinematic model to the error reduction. Finally, we present our elegant approach to efficiently use the complex and implicit model in optimization-based motion planning with almost no performance impact.

II. RELATED WORK

Roth et al. [3] provide an early overview on robot calibration, where they describe four critical aspects: the calibration model, the dataset, the identification of the model’s parameters, and the compensation, i.e., how to use the model in a robotic task. They also describe three different levels of calibration. The first level is only finding the joint offsets. The second is identifying all the robot’s geometric parameters. And the third level is the non-geometric calibration, including joint elasticities and gear backlash.

The first two calibration levels have been successfully applied to various robot arms [4, 5, 6]. In all these works, the focus is on algorithmic differences, what sets of DH parameters to use, and the comparison of the speed for calibration, but they ignore the speed of compensation.

Besides robotic arms, also more complex humanoid robots have been calibrated. Maier et al. [7] calibrate the joint offsets for the humanoid robot Nao and Stepanova et al. [8] all the DH parameters for the iCub robot. They also investigate how to combine different internal measurement chains to get a full-body calibration.

Nevertheless, there are cases for which a purely geometric model of the robot is not sufficient. A common source of non-geometric errors is the elasticities in the joints. If the robot is equipped with joint torque sensors, it is convenient to use their measurements in the kinematic model as done in Klodmann et al. [9] for the MIRO robot arm and by Besset et al. [10] for an LWR arm. But as the actual sensor readings are needed, this approach is only feasible for control but not for motion planning. Also, joint torque sensors can not measure any transversal effects.

A different approach for handling elasticities requires identifying a mass model of the robot and computing the torques at their static equilibrium for each pose. Caenen and Angue [11] showed how to incorporate torques into the DH-formalism by adding torque-dependent offsets to the rotational DH parameters. However, they completely neglected to find the torque equilibrium. Others [12, 13, 14] included an iterative search to find the equilibrium, but they only dealt with the joint elasticities and did not include transversal elasticities. Furthermore, they neglected the problem of efficient compensation of such an iterative model.

In the case of our elastic humanoid Agile Justin [2], an efficient compensation of the non-geometric model is crucial as we want to use it in an optimization-based path planner [1]. In previous work, we already automatically calibrated the multi-sensorial head [15, 16] and the IMUs in the head and the base [17]. Instead, in this paper, we provide an accurate calibration and efficient compensation for Agile Justin's full kinematic as needed for whole-body motion planning.

III. ROBOT MODEL

A. Geometric Model

The forward kinematics of a robot maps from the configuration space of generalized joint angles to the robots' physical pose in the cartesian workspace. This function is central to robotic path planning. E.g., grasping an object and checking for obstacle or self-collision all strongly depend on an accurate model of the robot's kinematics. A widely used representation of this kinematic model is formulated with the DH parameters. In this formulation, four values $\rho_i = [d_i, r_i, \alpha_i, \theta_i]$ describe the connection between two consecutive frames of the robot:

$${}^{i-1}T_i = \text{Rot}_x(\alpha_i) \cdot \text{Trans}_x(r_i) \cdot \text{Rot}_z(\theta_i) \cdot \text{Trans}_z(d_i) \quad (1)$$

The joints q_i are treated as offsets to θ_i or d_i in [2] depending on the type of joint. This minimal representation with two translational and two rotational parameters is enough to describe an arbitrary robot. However, a limit of this formulation shows up for parallel axes (see joints 1 to 3 in Fig. 1). In this case, it is impossible to represent a small variation at the next link with a small variation of the DH parameters. As [11] showed, a solution is to use an additional parameter β representing a rotation around the y-axis, leading to modified DH parameters $\rho_i = [d_i, r_i, \alpha_i, \beta_i, \theta_i]$:

$${}^{i-1}T_i = \text{Rot}_y(\beta_i) \cdot \text{Rot}_x(\alpha_i) \cdot \text{Trans}_x(r_i) \cdot \text{Rot}_z(\theta_i) \cdot \text{Trans}_z(d_i) \quad (2)$$

The frame of the TCP ($i = M$) relative to the robot's base ($i = 0$) is calculated by applying the transformations in series:

$${}^0T_M = {}^0T_1 \cdot {}^1T_2 \cdot \dots \cdot {}^{M-1}T_M \quad (3)$$

For more complex robots with a kinematic tree structure with multiple TCPs M_i , an equation like (3) holds for each branch. We understand the forward kinematics f to map from the robot configuration q not only to the position of the end effector(s) but to all frames of the robot.

$$F = [{}^0T_1, {}^0T_2, \dots, {}^0T_N] = f(q, \rho) \quad (4)$$

B. Non-geometric Model

However, only considering the geometric model falls short of describing the real robot. This paper focuses on torques as the primary source of non-geometric effects. In general, an acting torque will bend the robot and produce a slightly different position. As the DH parameters can describe an arbitrary robot, it is convenient to integrate the non-geometric effects into this formalism. Caenen and Angue [11] showed this idea by explicitly expressing the influence of torques onto the DH parameters. The simplest possibility is to add a torque-dependent linear to the geometric DH parameters ρ_0 :

$$\rho = \rho(\rho_0, \kappa, \tau) = \rho_0 + \kappa \tau \quad (5)$$

Note, that the effects of forces on the link lengths are neglected and the matrix $\kappa = [0, 0, \kappa^\alpha, \kappa^\beta, \kappa^\theta]$ describes only compliance around the respective axes, corresponding to the rotational DH parameters $[\alpha, \beta, \theta]$ and the acting torques $\tau = [\tau^x, \tau^y, \tau^z]$. As a consequence, also the forward kinematics now depends via the DH parameters ρ on the torques τ and the elasticity parameters κ :

$$F = f(q, \rho(\rho_0, \kappa, \tau)) \quad (6)$$

In the regime of no external forces and reasonably slow, quasi-static motions, the torques originate only from the robot's weight and its specific distribution. For a frame, the pair $v_j = [m_j, w_j]$ describes the mass m_j and its position w_j relative to this frame (i.e., ${}^0w_j = {}^0T_j w_j$). The torques produced by this mass due to the gravity vector g around the respective coordinate axes of another frame i with origin 0p_i can be calculated [11] by [11]:

$$\begin{aligned} \tau_{ij}^x &= (({}^0T_j w_j - {}^0p_{i-1}) \times m_j g) \cdot {}^0e_{i-1}^x \\ \tau_{ij}^y &= (({}^0T_j w_j - {}^0p_{i-1}) \times m_j g) \cdot {}^0e_{i-1}^y \\ \tau_{ij}^z &= (({}^0T_j w_j - {}^0p_i) \times m_j g) \cdot {}^0e_i^z \end{aligned} \quad (7)$$

To calculate the full torque τ_i acting on a link i , the contributions from all masses that act on the respective link have to be summed up, i.e., the torques from all masses which are higher up in the robot's kinematic tree.

¹Those torques act on the rotational axes parameterized by α_i , β_i , and θ_i , which are described in [2]. Because they make up a frame by sequential stacking, they do not all share the same frame of reference.

The torques $\tau = \tau(F, \mathbf{v})$ now depend on the weight distribution $\mathbf{v} = [v_1, v_2, \dots]$ of the robot and therefore on the frames F which are depending on the DH parameters ρ :

$$\rho = \rho(\rho_0, \kappa, \tau(F(q, \rho), \mathbf{v})). \quad (8)$$

This implicit equation for the DH parameters defines the equilibrium between the torques due to gravity and the torques due to flexion of the robot. We define the solution to this equation as the non-geometric DH parameters

$$\rho^* = \rho^*(q, \rho_0, \kappa, \mathbf{v}). \quad (9)$$

One possibility to solve (8) is by the following iteration:

$$\rho_n = (1 - \lambda)\rho_{n-1} + \lambda\rho(\rho_0, \kappa, \tau(F(q, \rho_{n-1}), \mathbf{v})) \quad (10)$$

$$\rho_\infty = \rho^* \quad (11)$$

Choosing an appropriate $\lambda \in [0, 1]$ for the weighted sum ensures the convergence of the iteration even for very soft (or strongly non-linear) robots². A suitable choice for the start of the iteration are the geometric DH parameters ρ_0 which can be interpreted as a robot in zero gravity or a robot with infinite stiffness. Finally, the non-geometric forward kinematics is then given by

$$F^* = f^*(q, \rho_0, \kappa, \mathbf{v}) := f(q, \rho^*(q, \rho_0, \kappa, \mathbf{v})). \quad (12)$$

IV. CALIBRATION

A. Measurement Model

We use an external camera system from Vicon, consisting of six 16Mpx cameras mounted on the ceiling and designed to track retro-reflective markers with high accuracy. We fixated two such markers on the robot's hands, the last link of the respective kinematic chains (see Fig. 2). To use the markers' positions y for calibrating the forward kinematics f , additional information is necessary. First, we need the robot's base relative to the camera system's world frame cT_0 . Second, the markers' position on the left and right end effector p_r and p_l must be known. It is not always possible to determine those frames beforehand; therefore, they become part of the calibration problem. Our measurement function h consists of the forward kinematics and the additional frames at the ends of the kinematic chain to close the measurement loop:

$$y = h(q, \Theta) = {}^cT_0 \cdot f^*(q, \rho_0, \kappa, \mathbf{v})_{r,l} \cdot [p_r, p_l] \quad (13)$$

The calibration parameters $\Theta = [\rho_0, \kappa, \mathbf{v}, c]$ are a combination of the DH parameters ρ , the compliances κ , the masses \mathbf{v} , and the parameters $c = [p_c, o_c, p_r, p_l]$ for the camera base-frame and marker positions to map the forward kinematics to the measurements. The camera frame cT_0 is here defined by its position p_c and its orientation o_c .

²The convergence can be shown by interpreting the DH parameters as generalized coordinates and the update rule as the discretized integration over time of a damped dynamical system (the robot moving due to gravity).

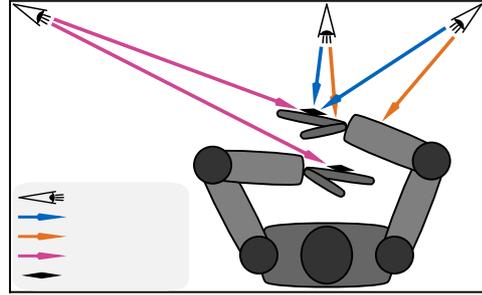


Fig. 2. A common measurement setup with cameras tracking markers on the end effectors; showcasing the problem of occlusion and suitable orientation of the marker for a given robot configuration.

B. Identification

The parameters Θ of the measurement function h can be identified using a dataset $D = \{(q^{(n)}, y^{(n)})\}_{n=1 \dots N}$ of pairs of the robot's configuration $q^{(n)}$ and the corresponding marker positions $y^{(n)}$. We formulate the identification as a single combined least-squares problem based on all measurements and all markers to minimize the task space error. To solve for the optimal parameters Θ^* , we use the maximum a posteriori (MAP) approach:

$$\min_{\Theta} \left[\sum_n \frac{1}{\sigma_m^2} |y^{(n)} - h(q^{(n)}, \Theta)|^2 + (\Theta - \Theta_p)^T \Lambda_p^{-1} (\Theta - \Theta_p) \right]$$

This approach uses a prior Gaussian distribution with mean Θ_p and a diagonal covariance matrix $\Lambda_p = \text{diag} \sigma_p^2$, where the vector σ_p describes the uncertainty of the different calibration parameters. For the measurement noise, we use the usual assumption of a Gaussian distribution with zero mean and standard deviation σ_m . The prior acts as regularization and guarantees the existence of a minimum, even in the presence of redundancies in the measurement/kinematic model. Setting the prior is not critical, but it should be set conservatively to plausible values. Because of the highly nonlinear measurement function, it might be necessary to use multistart for the initial guess to find the global minimum.

C. Configuration Selection and Efficient Sample Collection

We selected the measurement poses randomly by sampling from the configuration space to ensure good overall accuracy and not only close to some standard configurations. Nevertheless, the poses must be feasible for our experimental setup. Besides avoiding self-collision while measuring, the markers imposed additional constraints, as they must be well visible to the cameras. We used rejection sampling to ensure that at least four of the six cameras had a clear view of one marker. We checked for occlusion with simple ray tracing, a straight line from each camera to the marker's position. A sphere model of the robot, which we also use for collision checking, was used to test if the robot blocks any ray. Combining all those constraints, only 1/10000 configurations were feasible.

After determining a set of feasible configurations in simulation, we now need to measure those poses. The robot has

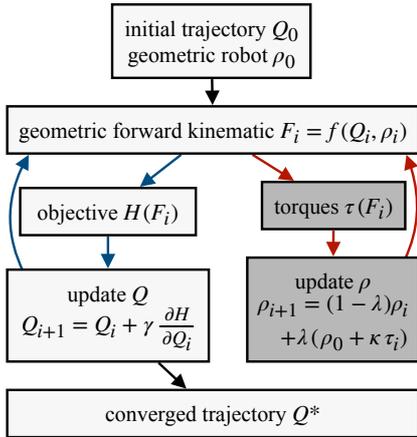


Fig. 3. Flowchart of the optimization loop in light gray. Dark gray shows the additional loop for the static torque equilibrium. As the outer loop converges, no additional passes through the inner torque loop are necessary.

to move to each configuration so that the camera system can collect the corresponding marker positions. One crucial concern regarding experiments with robots is always the time involved. To perform short and collision-free paths from pose to pose, we use an optimization-based path planner; the same planner we want to make more accurate through calibration, but with extended safety margins for the nominal kinematic.

To reduce the time further, we ordered the randomly selected poses to minimize the distance between them. By solving a traveling salesman problem on batches of 100 samples, we could reduce the time by a factor of two. Now the average time to collect one sample is ten seconds. From this, nine seconds fall on the robot performing the trajectory, and one second is added as a pause for the measurement.

V. COMPENSATION

The calibration goal is to find a set of parameters Θ which describes the robot as accurately as possible. Nevertheless, also the speed and ease of use of the calibration model, e.g. in motion planning, is crucial. Incorporating the new set of DH parameters ρ is straightforward and works without any changes or additional costs as they replace the old DH parameters. The same holds for masses m and compliances κ . However, to determine the elastic effects, one must find the static equilibrium between acting torques and elasticities described in (8). While it might be feasible for calibration (offline procedure) to use the iterative algorithm (10), it is more prohibitive for compensation (online). One should carefully evaluate this trade-off between accuracy and simplicity when choosing a calibration model for a robot.

Knowing that we will use the forward kinematics mostly in the framework of an optimization-based path planner has further implications. Such a planner works on paths in configuration space $Q = [q_1, q_2, \dots, q_n]$ and performs iterations to get from an initial path Q_0 to a converged path Q^* . For each step, the optimizer considers the objective function $H(Q_i)$

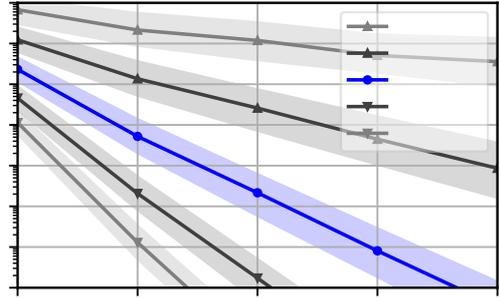


Fig. 4. Simulated convergence towards the static torque equilibrium for robots of different stiffness. For iteration 0 the elastic effect is ignored. The bands indicate the standard deviation for 1000 different joint configurations.

and updates the path using the gradient information. The nested structure of the forward kinematics

$$f^*(q) = f(q, \rho(f(q, \rho(f(\dots)))))) \quad (14)$$

leads to highly non-linear gradients, making the model more difficult to use. As a solution to this, we separate the torque equilibrium search in a separate loop, as shown in Fig. 3. After determining the acting torques accurately and updating the DH parameters accordingly, we assume these as constant for the gradient calculation $\partial F / \partial q$. This allows us to use the pure geometric forward kinematics, implicitly assuming $\partial \rho / \partial q = 0$. Although we completely omit the torque iterations for the gradients, this approximation does not hinder convergence in our tests.

The second important aspect visualized in Fig. 3 is the combination of the optimization loop and the torque equilibrium loop. Even if the updates of the configurations are large at the beginning of the optimization, when the planner converges, the pose updates get small. If those updates are significantly smaller than the offset produced by a torque update, it is unnecessary to search for a new static equilibrium iteratively. In other words, it is sufficient to reuse the already computed frames, and calculate the acting torques and update the DH parameters only once, while nevertheless solving the forward kinematics f^* exactly. The outer loop of the converging optimizer makes it unnecessary to perform inner iterations when searching the torque equilibrium.

VI. EXPERIMENTAL EVALUATION

A. Calibration

We collected $N_{\text{all}} = 500$ samples with the procedure described in Section IV-C and split them into a calibration set with $N = 300$ and a test set with $N_{\text{test}} = 200$ samples. This data, as well as an overview of the nominal and the calibrated parameters, are provided online³.

In what follows, we calibrate the parameters $\Theta = [\rho_0, \kappa, c]$, i.e., not v (the mass m and the center of mass w) because

³<https://dlr-ai.r.github.io/dlr-elastic-calibration>

TABLE I
CALIBRATION RESULTS: RESIDUAL TCP ERRORS [MILLIMETERS].

	Frames	DH parameter		Compliance	
	c	$\dots+\theta$	$\dots+d, r, \alpha, (\beta)$	$\dots+\kappa^\theta$	$\dots+\kappa^\alpha, (\kappa^\beta)$
μ	21.33	18.18	10.5	5.64	3.12
σ	9.71	7.8	5.74	2.57	1.71
max	63.41	45.9	36.37	12.83	8.23

before calibration

full calibration

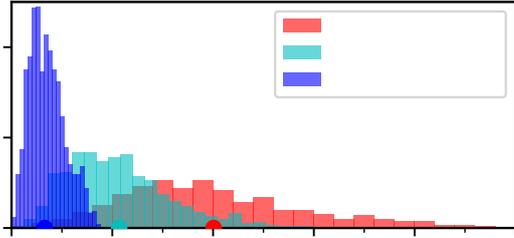


Fig. 5. Histogram of the residual error at the end effectors before and after calibration, highlighting the need for non-geometric modeling of Justin. The circles on the x -axis mark the mean μ of the error (see also Table I).

they could be adopted from the CAD-files of the robot. In a test, we found that adding v to the calibration parameters did not improve the residual error but significantly increased the runtime of the optimization algorithm. The priors used for the following calibrations were chosen based on previous experiments with the robot, with uncertainty σ_p in lengths of 0.1 m, angles of 0.2 rad, and elasticities of 0.1 rad/kNm.

In all experiments, we used multi start for the optimization with initial guesses Θ_0 randomly sampled from the prior distribution. All optimization runs converged to the same optimum, showing the stability of our calibration approach.

As the first step, the additional frames parametrized by c must be determined to close the measurement loop. Only calibrating those frames gives the accuracy of the nominal kinematics before calibration, with a mean residual error of 20 mm - averaged over the 200 test poses and both arms. In the worst cases, the error was larger than 60 mm. After calibration, the mean residual error of the full model is reduced to 3.1 mm and the maximum error to 8.2 mm. Those two cases are reported in Table I in the left- and rightmost columns and a detailed distribution of the errors is shown in Fig. 5. Here the result of a pure geometric calibration without elasticities is also shown, which highlights the need for a non-geometric calibration model for this robot.

We conducted further experiments to evaluate the significance of the different model parameters. Table II reports the residual error when making the model more expressive by adding more parameters step by step, starting with the uncalibrated model (leftmost column) up to the full model (rightmost column). The order in which we added the different types of parameters (always for all joints at once) followed "standard practice" with joint offsets as first and transversal elasticities as the last addition.

TABLE II

"LEAVE-ONE-OUT ANALYSIS": RESIDUAL TCP ERRORS [MILLIMETERS].

	full	$-\kappa^\theta$	$-\alpha$	$-\theta$	$-\kappa^\alpha$	$-r$	$-d$
μ	3.12	9.09	6.48	6.35	4.85	4.33	3.46
σ	1.71	4.23	2.93	3.16	2.28	2.03	1.73
max	8.23	24.59	17.25	18.89	13.37	11.09	8.97

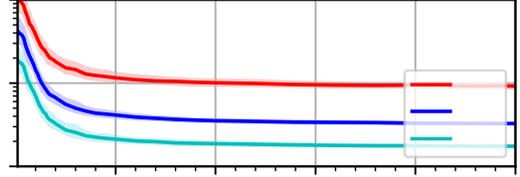


Fig. 6. Test error over the calibration set size N . The bands show the standard deviation over 1000 different calibration sets of a given size.

That this "standard ordering" does not represent the actual influence of the parameters on the residual error can be seen in Table II. Here, starting from the full model (rightmost column), only a specific parameter type was left out one at a time. From this, it can be seen how crucial joint and transversal elasticities are for modeling Justin's kinematics. One reason for the joint elasticities being that prominent is that we not only model the mechanical elasticity of a joint but also the elasticity of the joint position controller⁴.

Fig. 6 shows how the test error decreases when more samples N are used for calibrating the full model. Furthermore, one can see the influence of which robot configurations (joint angles) are in a set. The light bands show the standard deviation when selecting 1000 different sets of a given size. While there is still some minor improvement beyond $N = 100$ samples, the main effect of the calibration happens with as little as $N = 50$ samples (those could be collected in less than ten minutes). This information is especially beneficial for recalibrating the robot with a subset of the parameters in the future. A smaller calibration model corresponds to a smaller set necessary for calibration, making the effect of selecting an optimal set of configurations [15] more significant.

B. Compensation

Fig. 4 visualizes the non-geometric effects in more detail by showing the convergence towards a static equilibrium for solving the kinematics via iteration according to (10). We show results for the real and virtual versions of Agile Justin with lower and higher elasticities. For the real robot, the difference after just one iteration is already considerably below our calibration accuracy. But if the robot is softer or the accuracy requirements are stricter, one needs more iterations to converge to the equilibrium configuration.

In addition, we tested the algorithm described in Fig. 3 on these versions of Agile Justin. While for the real or stiffer

⁴We use a relatively simple joint position controller as it is robust (e.g., it does not rely on the torque sensors, which are notoriously drifting and hard to maintain). However, it results in an additional joint-level elasticity.

versions no additional iterations were necessary to converge to the grasping poses, the procedure also works for soft robots with stiffnesses in the range of $\sim 100\text{N/rad}$. In this case, the number of iterations increased by 30%. In all cases, the wall clock time for one loop of the optimizer increases by only one percent. Because no additional calculation of the forward kinematics is necessary, and one pass through the torque loop is enough.

VII. CONCLUSIONS

In this paper – to our knowledge for the first time – a calibration model including all DH parameters, joint and transversal elasticities was formulated and successfully calibrated for a humanoid robot.

For DLR’s Agile Justin, the average error could be significantly reduced from prohibitively large 21 mm to 3.1 mm in the whole workspace. We provided a clear and concise derivation of the implicit kinematic model with elasticities from a torque equilibrium. We showed that this complex implicit model can be used in optimization-based motion planning without any performance impact (for mildly elastic robots like Agile Justin). And even for a simulated soft robot with 20 times higher elasticities, the slowdown was only 30%. We achieved this by tightly integrating the iterative solver for the implicit kinematic model into the optimization planner loop of the planner. Finally, we provided an in-depth discussion of the influence of the individual components of the model on the residual position error.

In the future, we want to make the calibration completely automatic and self-contained by replacing the use of an external tracking system by using the robot’s head-mounted cameras, so combining this work with our previous work on automatic camera calibration [16].

REFERENCES

- [1] R. Wagner, U. Frese, and B. Bauml, “3D modeling, distance and gradient computation for motion planning: A direct GPGPU approach,” in *2013 IEEE International Conference on Robotics and Automation*, no. Iii. IEEE, 5 2013, pp. 3586–3592.
- [2] B. Bauml *et al.*, “Agile Justin: An upgraded member of DLR’s family of lightweight and torque controlled humanoids,” in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 5 2014, pp. 2562–2563.
- [3] Z. Roth, B. Mooring, and B. Ravani, “An overview of robot calibration,” *IEEE Journal on Robotics and Automation*, vol. 3, no. 5, pp. 377–385, 10 1987.
- [4] U. S. Pathre and M. R. Driels, “Simulation experiments in parameter identification for robot calibration,” *The International Journal of Advanced Manufacturing Technology*, vol. 5, no. 1, pp. 13–33, 2 1990.
- [5] L. S. Ginani and J. M. S. T. Motta, “Theoretical and practical aspects of robot calibration with experimental verification,” *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, vol. 33, no. 1, pp. 15–21, 3 2011.
- [6] I. W. Park and J. H. Kim, “Estimating entire geometric parameter errors of manipulator arm using laser module and stationary camera,” *IECON Proceedings (Industrial Electronics Conference)*, pp. 129–134, 2011.
- [7] D. Maier, S. Wrobel, and M. Bennewitz, “Whole-body self-calibration via graph-optimization and automatic configuration selection,” in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, vol. 2015-June, no. June. IEEE, 5 2015, pp. 5662–5668.
- [8] K. Stepanova, T. Pajdla, and M. Hoffmann, “Robot Self-Calibration Using Multiple Kinematic Chains-A Simulation Study on the iCub Humanoid Robot,” *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1900–1907, 2019.
- [9] J. Klodmann, R. Konietschke, A. Albu-Schaffer, and G. Hirzinger, “Static calibration of the DLR medical robot MIRO, a flexible lightweight robot with integrated torque sensors,” in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 9 2011, pp. 3708–3715.
- [10] P. Besset, A. Olabi, and O. Gibaru, “Advanced calibration applied to a collaborative robot,” in *2016 IEEE International Power Electronics and Motion Control Conference (PEMC)*. IEEE, 9 2016, pp. 662–667.
- [11] J. Caenen and J. Angue, “Identification of geometric and nongeometric parameters of robots,” in *Proceedings., IEEE International Conference on Robotics and Automation*. IEEE Comput. Soc. Press, 1990, pp. 1032–1037.
- [12] W. Khalil and S. Besnard, “Geometric calibration of robots with flexible joints and links,” *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 34, no. 4, pp. 357–379, 2002.
- [13] B.-J. Lee, “Geometrical Derivation of Differential Kinematics to Calibrate Model Parameters of Flexible Manipulator,” *International Journal of Advanced Robotic Systems*, vol. 10, no. 2, p. 106, 2 2013.
- [14] J. Zhou, H.-N. Nguyen, and H.-J. Kang, “Simultaneous identification of joint compliance and kinematic parameters of industrial robots,” *International Journal of Precision Engineering and Manufacturing*, vol. 15, no. 11, pp. 2257–2264, 11 2014.
- [15] H. Carrillo *et al.*, “On task-oriented criteria for configurations selection in robot calibration,” in *2013 IEEE International Conference on Robotics and Automation*. IEEE, 5 2013, pp. 3653–3659.
- [16] O. Birbach, U. Frese, and B. Bäuml, “Rapid calibration of a multi-sensorial humanoid’s upper body: An automatic and self-contained approach,” *International Journal of Robotics Research*, vol. 34, no. 4-5, pp. 420–436, 2015.
- [17] O. Birbach and B. Bauml, “Calibrating a pair of inertial sensors at opposite ends of an imperfect kinematic chain,” in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, no. Iros. IEEE, 9 2014, pp. 422–428.