Whole-Body Self-Calibration via Graph-Optimization and Automatic Configuration Selection

Daniel Maier Stefan Wrobel Maren Bennewitz

***Abstract*— In this paper, we present a novel approach to accurately calibrate the kinematic model of a humanoid based on observations of its monocular camera. Our technique estimates the parameters of the complete model, consisting of the joint angle offsets of the whole body including the legs, as well as the camera extrinsic and intrinsic parameters. We cast the parameter estimation as a least-squares optimization problem. In the error function, we consider the residuals between camera observations of end-effector markers and their projections into the image based on the estimate of the calibration parameters. Furthermore, we developed an approach to automatically select a subset of configurations for the calibration process that yields a good trade-off between the number of observations and accuracy. As the experiments with a Nao humanoid show, we achieve an accurate calibration for this low-cost platform. Further, our approach to configuration selection yields substantially better optimization results compared to randomly chosen viable configurations. Hence, our system only requires a reduced number of configurations to achieve accurate results. Our optimization is general and the implementation, which is available online, can easily be applied to different humanoids.**

I. Introduction

Knowledge about the parameters of a robot's kinematic model is essential for all tasks involving navigation and manipulation. For mapping unknown environments or estimating the positions of objects, the transforms between the exteroceptive sensors and the robot's internal reference frame, that bring the observations in relation to the robot, need to be known. Only then, it can avoid collisions with objects, and plan paths to target locations or grasps.

For manipulation, an accurate kinematic model is necessary for solving the inverse kinematics and controlling a manipulator to reach grasping targets, as well as for collision checking of arm trajectories with the environment. While techniques such as visual servoing allow to control a manipulator without accurate knowledge of the calibration, a certain degree of calibration is still necessary, and an accurate model of the robot allows for faster control and an easier implementation, including open-loop control.

The kinematic structure of a robot is usually known from the construction plan. However, errors can occur, e.g., as a result of imperfect manufacturing, wear, or repair. To compensate for such errors and to avoid time-consuming manual tuning of the true parameters of the models, we present an automatic self-calibration technique for humanoid robots. Our approach calibrates the robot's kinematic model, given its structure, based only on observations from the robot's internal monocular camera. We estimate the parameters of the kinematic model of the whole body in form of joint angle offsets as well as camera extrinsic and intrinsic parameters. We formulate the calibration as an error minimization problem and apply the g^{2}o optimization framework. In the error function, we consider the residuals between observed positions of end-effector markers in the camera image and their expected locations given the estimated parameters. Additionally, we present a method to automatically generate robot configurations for the calibration and to determine a small subset of configurations that lead to an accurate calibration, thereby trading of time and accuracy.

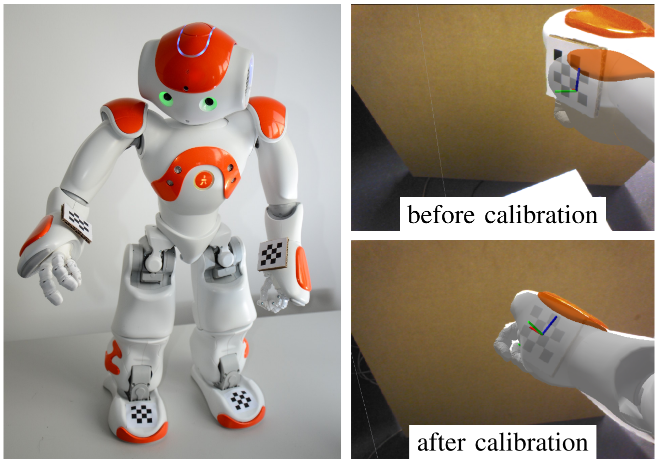


Fig. 1. Left: Nao humanoid with markers attached to the end-effectors (EEF) performing automatic self-calibration. Right: Marker on the right EEF observed by the onboard-camera before and after the calibration, with the robot model and the expected marker center overlaid.

We applied our approach to calibrate the parameters of a Nao humanoid and show in experiments that our method leads to an accurate calibration and needs only a small number of configurations to achieve this. Fig. 1 shows a Nao performing the self-calibration (left), as well as the expected marker location and a 3D visualization of the kinematic model overlaid to the onboard camera image, before (top right) and after (bottom right) the calibration.

To the best of our knowledge, this is the first approach to calibrate the complete kinematic model of a legged humanoid. We released the source code of our ROS-based implementation providing a one-click calibration routine for the Nao robot, which is the most commonly used humanoid. Adaptions for different humanoids can easily be obtained.

All authors are with the Humanoid Robots Lab, Univ. of Freiburg, Germany. M. Bennewitz is also with the Humanoid Robots Lab, Univ. of Bonn. This work has been supported by the German Research Foundation (DFG) under contract number SFB/TR-8 and the *BrainLinks-BrainTools* Cluster of Excellence (grant number EXC 1086).

II. Related Work

Yamane [1] proposed to calibrate the joint offsets of a humanoid's legs using measurements of a chest-mounted Inertial Measurement Unit (IMU) and the constraint that the feet of the robot are fixed on a plane, while manually moving the robot. However, the upper body cannot be calibrated with such a method as the corresponding joints do not affect the IMU measurements. Park *et al.* [2] use optimization of the Denavit-Hartenberg parameters of a manipulator using a laser attached to the end-effector and an external camera within a Kalman-Filter framework. While this allows for precise measurements, it also requires a substantial modification of the hardware. Asfour *et al.* [3] developed an approach to calibrate a stereo camera relative to the joints of a humanoid's head by observing a marker and using a least-squares formulation of the problem. Hence, they perform a classical head-eye calibration. Traslosheros *et al.* [4], considered calibration of a robot with two parallel manipulators and a camera. For their approach, it is assumed that the camera intrinsics are already well-known. The authors use a ball that is fixed within the environment as measurement marker and the geometrical parameters of the ball projected into the image plane are used for the constraint equations.

Pradeep *et al.* [5] presented a bundle adjustment approach for calibrating a robot with multiple sensors and arms. The main idea here is that multiple sensors can observe the same points in the world and hence, the parameters can be jointly optimized. Birbach *et al.* [6] proposed an approach to calibrate the intrinsic and extrinsic parameters of a stereo camera, the joint elasticities and the joint angle offsets of an upper body of a humanoid with two arms, and the head mounted IMU. As in [5], the authors jointly optimize the parameters from manually determined robot configurations. The authors use a single point marker located on each of the arms' wrists. Similarly, Hubert *et al.* [7] applied a maximum a priori approach (MAP) for estimating the parameters of hand-eye kinematics of a humanoid robot. The prior is assumed to be normal distributed with an empirically determined variance. If the prior is well chosen, the MAP approach requires less measurements and, thus, is faster as a maximum likelihood formulation. However, our approach automatically selects configurations for calibration such that the number of needed measurements is reduced.

Selection and generation of measurement configurations has been studied previously, especially in the context of manipulator calibration. For instance, Borm and Menq [8] studied the selection of measurement configurations for manipulator calibration. Their goal is to minimize the effect of noise in measurement. Using an observability measure based on the Jacobian of the error function, the authors show that the selection of the measurement configurations is more important than the actual number of measurements. Our system follows that idea by evaluating different observability indices. Also Li *et al.* [9, 10] used observability indices to choose configurations for calibration of a manipulator. The authors showed that using the optimized configurations yields better results than using configurations sampled from the boundary of the working space, which is otherwise a reasonable heuristic.

Most similar to our approach, Carrillo *et al.* [11] recently experimentally compared different criteria for the selection of measurement configurations for the calibration of a humanoid's upper body. The authors conclude that for greedy-selection algorithms, determinant-based criteria should be preferred. However, the paper only shows that applying a determinant-based criterion maximize the other criteria as well, but not whether it minimizes the actual calibration error. In our work, we present extensive cross-validation results for four different selection criteria.

III. Graphed-Based Optimization of the Calibration Parameters

Poor calibration of a robot leads to a discrepancy between self-observation, e.g., from a camera, and expected observation according to underlying models and their estimated parameters. Our calibration framework hence measures this discrepancy and adjusts the parameters so that the error is minimized. We hereby rely on camera observations of point markers attached to the robot's end-effectors. We determine their expected locations in the camera image and compute the error when compared to the actual observations.

A. Measurement Model and Parameters

One of the most obvious reasons for a misalignment between a true observation and the expected one is that the camera's intrinsic and extrinsic parameters are not known accurately. Our system uses the standard pinhole model to compute the projection of a point in the camera image and also considers the radial distortion of the lens, which is typically the most dominant reason for deviations. Let $\mathbf{p} = \begin{bmatrix} x & y & z \end{bmatrix}^\intercal$ be a 3D point in the camera frame, then the corresponding image coordinates are given by the function

$$
\begin{aligned}
\mathsf{proj} &: \mathbb{R}^3 \mapsto \mathbb{R}^2, \\
\mathsf{proj}(\mathbf{p}) &= \begin{bmatrix} u & v \end{bmatrix}^\intercal, && (1) \\
\text{where} \quad \begin{bmatrix} u & v & 1 \end{bmatrix}^\intercal &= K \begin{bmatrix} x' & y' & 1 \end{bmatrix}^\intercal, && (2) \\
\text{with} \quad \begin{bmatrix} x' & y' \end{bmatrix}^\intercal &= (1 + \kappa r^2) \begin{bmatrix} x/z & y/z \end{bmatrix}, && (3) \\
\text{and} \quad r^2 &= (x/z)^2 + (y/x)^2, && (4) \\
\text{and} \quad K &= \begin{bmatrix} f_x & 0 & k_x \\ 0 & f_y & k_y \\ 0 & 0 & 1 \end{bmatrix}. && (5)
\end{aligned}
$$

Here, the focal lengths (f_x, f_y) and the principle point (k_x, k_y) are the pinhole model's parameters, and κ models the strength of the radial distortion.

Also the calibration of the camera extrinsics is of great importance, i.e., its placement relative to the reference frame that it is attached to, e.g., the robot's neck joint. Let $\tilde{\mathbf{x}} \in \mathbb{R}^4$ be a 3D homogeneous point in the reference frame, then

$$\begin{bmatrix} x & y & z \end{bmatrix}^\intercal = R \left[I_{3\times3} \; -\mathbf{c} \right] \tilde{\mathbf{x}} \qquad (6)$$

describes the point in the camera frame. The extrinsics are thus modeled by the camera's rotation matrix R and its center point $\mathbf{c}$ relative to the reference frame.

One additional important reason for poor calibration is that the rotary joint encoders, which contribute to the kinematic state of the robot, are affected by systematic offsets due to lack of precision in the manufacturing process or wear. Hence, we model the true position of the joints at time i as the sum of the offsets $\mathbf{q}^{\text{off}}$ and the encoder readings $\hat{\mathbf{q}}_i$

$$\mathbf{q}_i = \hat{\mathbf{q}}_i + \mathbf{q}^{\text{off}}. \tag{7}$$

Finally, we need to consider the artificial markers attached to the end-effectors (EEF) as we rely on them as observations. While their approximate positions can be obtained from manual measuring, positional errors directly affect the automatic evaluation of the state of calibration of the robot. Hence, we include the position for the markers M_{EEF} on each end-effector in the estimation and model them as three-dimensional vectors relative to the EEF-frame by

$$\mathbf{m}_{\text{EEF}} = \begin{bmatrix} x & y & z \end{bmatrix}^{\mathsf{T}}. \tag{8}$$

To summarize, the parameters considered in the error minimization are

- the camera intrinsics and distortion fx, fy, k_x, k_y, κ,
- the camera extrinsics $R, \mathbf{c}$,
- one marker location $\mathbf{m}_{\text{EEF}}$ per end-effector, and
- the joint offsets $\mathbf{q}^{\text{off}}$.

For simplicity, in the remainder of this paper, we stack all these parameters in a vector $\boldsymbol{\theta} \in \mathbb{R}^L$.

B. Formulation as Least-Squares Optimization

Thus, our calibration procedure aims at estimating the parameters $\boldsymbol{\theta}$ that minimize the error between the estimated poses of the markers attached to the end-effectors and their observations in the camera image. To obtain a robust estimate for $\boldsymbol{\theta}$, we consider the accumulated error for a set ω consisting of n different robot configurations, where each configuration is defined by its joint encoder readings $\hat{\mathbf{q}}_i \in \omega$. We formulate the error minimization using least-squares optimization. Hence, we minimize

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} F(\boldsymbol{\theta}), \tag{9}$$

where $F : \mathbb{R}^L \longrightarrow \mathbb{R}$ is given by

$$F(\boldsymbol{\theta}) := \sum_{i=1..n} \mathbf{e}_i(\boldsymbol{\theta}, \hat{\mathbf{z}}_i, \hat{\mathbf{q}}_i)^{\mathsf{T}} \mathbf{e}_i(\boldsymbol{\theta}, \hat{\mathbf{z}}_i, \hat{\mathbf{q}}_i), \tag{10}$$

and e_i is the error function for the robot configuration $\hat{\mathbf{q}}_i$ and measurement $\hat{\mathbf{z}}_i$. Note that F depends on ω, but we omit the index to enhance readability. For the employed point markers, the error function is given by

$$\mathbf{e}_i(\boldsymbol{\theta}, \hat{\mathbf{z}}_i, \hat{\mathbf{q}}_i) := \hat{\mathbf{z}}_i - \mathsf{predict}_{M_{\text{EEF}}}(\boldsymbol{\theta}, \hat{\mathbf{q}}_i). \tag{11}$$

Here, $\hat{\mathbf{z}}_i \in \mathbb{R}^2$ is the observed marker location in the image, from which we subtract the estimate for the projection of the marker $\mathsf{predict}_{M_{\text{EEF}}}(\boldsymbol{\theta}, \hat{\mathbf{q}}_i)$, given the robot's kinematic structure, the calibration parameters $\boldsymbol{\theta}$, and the joint readings $\hat{\mathbf{q}}_i$. The predicted location of a marker M is given by

$$\mathsf{predict}_{M_{\text{EEF}}}(\boldsymbol{\theta}, \hat{\mathbf{q}}_i) := \mathsf{proj}_{\boldsymbol{\theta}}(\mathsf{forward}^{\mathsf{C}}_{M_{\text{EEF}}}(\boldsymbol{\theta}, \hat{\mathbf{q}}_i)), \tag{12}$$

where $\mathsf{proj}_{\boldsymbol{\theta}}$ is described in Eq. 1 using the current calibration parameters $\boldsymbol{\theta}$ and $\mathsf{forward}^{\mathsf{C}}_{M_{\text{EEF}}}(\boldsymbol{\theta},\, \hat{\mathbf{q}}_i)$ determines the location of the marker M_{EEF} relative to the camera using forward kinematics, given the joint readings $\hat{\mathbf{q}}_i$ and the calibration parameters $\boldsymbol{\theta}$. $\mathsf{forward}^{\mathsf{C}}_{M_{\text{EEF}}}$ can be computed by the multiplication of a series of homogeneous matrices that represent the transforms between joints in the kinematic chain from the camera to the marker M_{EEF}, taking into account the estimated joint positions $\hat{\mathbf{q}}_i + \mathbf{q}^{\text{off}}$ in the rotational part of the matrices.

To solve the least-squares problem defined in Eq. 9 - Eq. 12, we apply a linear approximation. Let

$$\mathbf{f}(\boldsymbol{\theta}) := \begin{bmatrix} \mathbf{e}_1(\boldsymbol{\theta}, \hat{\mathbf{z}}_1, \hat{\mathbf{q}}_1)^{\mathsf{T}} & \dots & \mathbf{e}_n(\boldsymbol{\theta}, \hat{\mathbf{z}}_n, \hat{\mathbf{q}}_n)^{\mathsf{T}} \end{bmatrix}^{\mathsf{T}}, \tag{13}$$

by which Eq. 10 can be formulated as

$$F(\boldsymbol{\theta}) = \mathbf{f}(\boldsymbol{\theta})^{\mathsf{T}} \mathbf{f}(\boldsymbol{\theta}). \tag{14}$$

We apply a linear first-order approximation of $\mathbf{f}$ by defining

$$\boldsymbol{\theta} := \tilde{\boldsymbol{\theta}} + \boldsymbol{\Delta\theta}. \tag{15}$$

$$\text{Thus, } \mathbf{f}(\boldsymbol{\theta}) = \mathbf{f}(\tilde{\boldsymbol{\theta}} + \boldsymbol{\Delta\theta}) \approx \mathbf{f}(\tilde{\boldsymbol{\theta}}) + J\,\boldsymbol{\Delta\theta}, \tag{16}$$

$$\text{where } J := \frac{\partial \mathbf{f}}{\partial \boldsymbol{\theta}}(\tilde{\boldsymbol{\theta}}) \tag{17}$$

is the Jacobian of $\mathbf{f}$ at the linearization point $\tilde{\boldsymbol{\theta}}$. Intuitively, J describes how the error changes with changes in the parameter-space around $\tilde{\boldsymbol{\theta}}$. By plugging Eq. 16 into Eq. 14, we obtain

$$F(\boldsymbol{\theta}) \approx (\mathbf{f}(\tilde{\boldsymbol{\theta}}) + J\boldsymbol{\Delta\theta})^{\mathsf{T}}\,(\mathbf{f}(\tilde{\boldsymbol{\theta}}) + J\boldsymbol{\Delta\theta}) \tag{18}$$

$$= \underbrace{\mathbf{f}^{\mathsf{T}}(\tilde{\boldsymbol{\theta}})\mathbf{f}(\tilde{\boldsymbol{\theta}})}_{=:c} + \underbrace{2\mathbf{f}^{\mathsf{T}}(\tilde{\boldsymbol{\theta}})J\boldsymbol{\Delta\theta}}_{=:2b^{\mathsf{T}}\boldsymbol{\Delta\theta}} + \underbrace{\boldsymbol{\Delta\theta}J^{\mathsf{T}}J\boldsymbol{\Delta\theta}}_{=:\boldsymbol{\Delta\theta}^{\mathsf{T}}H\boldsymbol{\Delta\theta}}. \tag{19}$$

Thus, we obtain an approximate error function that does not depend on $\boldsymbol{\theta}$ anymore but on $\boldsymbol{\Delta\theta}$. So we define

$$F_{\text{app}}(\boldsymbol{\Delta\theta}) := c + 2b^{\mathsf{T}}\boldsymbol{\Delta\theta} + \boldsymbol{\Delta\theta}^{\mathsf{T}}H\boldsymbol{\Delta\theta}. \tag{20}$$

To minimize Eq. 20, we set its derivative to zero and solve the linear system,

$$\frac{\partial F_{\text{app}}}{\partial \boldsymbol{\Delta\theta}} = 0 \iff 2b + 2H\boldsymbol{\Delta\theta} = 0 \tag{21}$$

$$H\boldsymbol{\Delta\theta} = -b \tag{22}$$

$$\boldsymbol{\Delta\theta} = -(J^{\mathsf{T}}J)^{-1}J^{\mathsf{T}}\mathbf{f}(\tilde{\boldsymbol{\theta}}). \tag{23}$$

An estimate for $\boldsymbol{\theta}$ can then be obtained using the definition in Eq. 15. We use the Levenberg-Marquardt (LM) algorithm to minimize the error. LM repeats the three steps, i.e., the linearization in Eq. 16, the minimization in Eq. 23, and the increment in Eq. 15, until convergence.

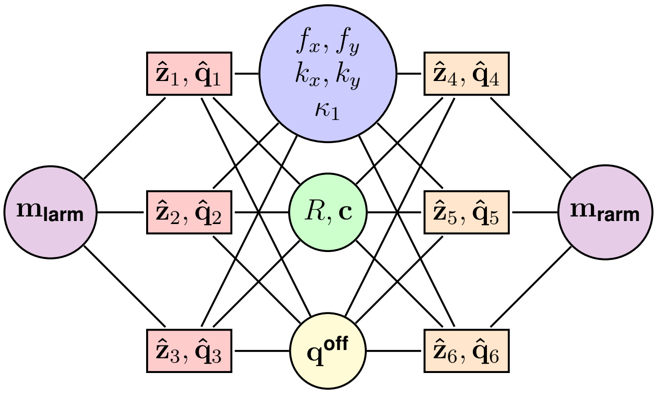

f_x, f_y
k_x, k_y
κ_1
$\hat{\mathbf{z}}_1, \hat{\mathbf{q}}_1$
$\hat{\mathbf{z}}_4, \hat{\mathbf{q}}_4$
$\mathbf{m}_{\mathbf{larm}}$
$\hat{\mathbf{z}}_2, \hat{\mathbf{q}}_2$
$R, \mathbf{c}$
$\hat{\mathbf{z}}_5, \hat{\mathbf{q}}_5$
$\mathbf{m}_{\mathbf{rarm}}$
$\hat{\mathbf{z}}_3, \hat{\mathbf{q}}_3$
$\mathbf{q}^{\mathbf{off}}$
$\hat{\mathbf{z}}_6, \hat{\mathbf{q}}_6$

Fig. 2. Example g^{2}o graph for the robot calibration. The graph includes nodes for the camera intrinsics, camera extrinsics, joint offsets, and the markers of the two arms. Three measurements per chain are indicated.

C. Implementation

We implemented our calibration approach using the g^{2}o library [12]. It solves problems of the form given in Eq. 9 and Eq. 10 by optimizing a hyper-graph. The graph consists of a set of vertices that encode the sub-components of the parameter vector $\boldsymbol{\theta}$ to be optimized and the measurements $(\hat{\mathbf{z}}_i, \hat{\mathbf{q}}_i)$. The edges represent the error functions from Eq. 11 between corresponding vertices. Fig. 2 shows an example for such a graph with six measurements and two markers. As can be seen, the edges for one measurement connect multiple vertices. Next to its good performance, g^{2}o automatically handles problems with singularities and over-parameterization of rotation representations.

In g^{2}o, it is also possible to use a robust kernel such as the Huber loss function in Eq. 10. Thus, it optimizes

$$F_H(\boldsymbol{\theta}) = \sum_{i=1..n} \rho_H(\mathbf{e}_i(\boldsymbol{\theta}, \hat{\mathbf{z}}_i, \hat{\mathbf{q}}_i)) \tag{24}$$

$$\text{where} \quad \rho_H(\mathbf{x}) = \left[\rho_H(\mathbf{x}_1), \rho_H(\mathbf{x}_2), \cdots, \rho_H(\mathbf{x}_m)\right]^\intercal \tag{25}$$

$$\text{and} \quad \rho_H(\mathbf{x}_i) = \begin{cases} \mathbf{x}_i^2 & \text{if } \|\mathbf{x}_i\| < b \\ 2b\,\|\mathbf{x}_i\| - b^2 & \text{else} \end{cases}, \tag{26}$$

where the $\mathbf{x}_i$ are scalars and b is a scaling parameter. This means, $\rho_H(\mathbf{x}_i)$ is quadratic for small errors $\mathbf{x}_i$, and linear for larger ones. In case of $b = \infty$, Eq. 24 is identical to Eq. 10.

IV. Automatic Selection of Robot Configurations

The result of the optimization described in Sec. III-B depends on the initial linearization point and the set of configurations with their measurements for which the error minimization is carried out. A higher number of configurations can lead to a more accurate parameter estimation but, obviously, they consume more time. A small set of configurations on the other hand might lead to overfitting on the calibration data. In order to reduce the number of needed configurations, while still enabling robust estimates, we developed an approach to automatically select a set of close-to-optimal configurations. The goal is to achieve a good trade-off between the time needed to carry out the calibration and the accuracy of the result.

A. Generating a Pool of Configurations

Our system initially samples a large pool Ω of configurations from which subsequently a subset of configurations that optimizes a certain criterion is selected. The configurations are uniformly sampled in joint space and checked for self-collisions as well as marker observability. A configuration $\mathbf{q}$ is a vector of admissible values for all the joints that are included in the calibration.

For each sampled configuration $\hat{\mathbf{q}}_i$, we first project the marker location into the camera image, given the current estimate of $\boldsymbol{\theta}$ and the kinematic structure of the robot as described in the Sec. III-B. The system only accepts $\hat{\mathbf{q}}_i$, if $\mathbf{z_i} = \mathsf{predict}_{M_{\text{EEF}}}(\boldsymbol{\theta}, \hat{\mathbf{q}}_i)$ is within the boundaries of the camera image minus some safety margin. Next, the algorithm tests whether the robot is in self-collision using a mesh-model and the Flexible Collision Library (FCL) [13]. Finally, our approach checks whether the marker is not occluded by any body part in this configuration and visible by the camera.

B. Selecting a Subset of Valuable Configurations

To automatically generate a set of good configurations for calibration, we examine the Jacobian from Eq. 17. The Jacobian is decomposed using Singular Value Decomposition (SVD) into

$$J = U \Sigma V^\intercal, \tag{27}$$

where U and V are orthogonal matrices and Σ is the diagonal matrix with the singular values $(\sigma_1, \ldots, \sigma_L)$ of J, with $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_L$. Here, we assume that there are $L = |\theta|$ columns, corresponding to the number of parameters, and $\nu > L$ rows in J, otherwise the system is under-determined. That means, at least $\lceil L/2 \rceil$ measurements need to be considered, as each measurement yields two constraints. On this decomposition, we apply a so-called observability index, which is based on the σ_i and yields higher values for larger σ_i and lower values for smaller σ_i. The rationale behind maximizing the σ_i is that given the approximation in Eq. 16, the covariance of $\boldsymbol{\theta}$ can be estimated as

$$\operatorname{cov}(\boldsymbol{\theta}) \approx (J^\intercal J)^{-1}. \tag{28}$$

Thus, maximizing the singular values σ_i of J is equivalent to minimizing the Eigenvalues λ_i of $\operatorname{cov}(\boldsymbol{\theta})$, as $\sigma_i = \lambda_i^{-2}$. Hence, we seek to reduce the variance or uncertainty of the parameters. To this end, we evaluated the following observability indices [11, 14]:

$$O_{\text{D}} = (\sigma_1 \cdots \sigma_L)^{1/L}\, \nu^{-1/2} \tag{29}$$

$$O_{\text{A}} = (\sigma_1^{-1} + \cdots + \sigma_L^{-1})^{-1} \tag{30}$$

$$O_{\text{NAI}} = \sigma_L^2 / \sigma_1 \tag{31}$$

$$O_{\text{E}} = \sigma_L. \tag{32}$$

O_{D} corresponds to the determinant of $\operatorname{cov}(\boldsymbol{\theta})$, O_{A} to its trace, and O_{E} to the largest Eigenvalue of $\operatorname{cov}(\boldsymbol{\theta})$, while O_{NAI} has no direct relation with the covariance, but can be seen as the product of the Jacobian's inverse condition number and it's smallest singular value.

Algorithm 1: O_D-select: Selects N^* configurations from a pool Ω for calibration and optimizes $\boldsymbol{\theta}$

```
Input:  Pool of configurations $\Omega$
        Initial calibration parameters $\boldsymbol{\theta}$, with $|\boldsymbol{\theta}| = L$
        Desired number of configurations $N^*$
        Max. number of retries for sampling initial set $T$
Output: Optimized configurations $\omega^*$
        Optimized calibration paramters $\boldsymbol{\theta}$
$\omega^* \longleftarrow$ uniformly sample $\lceil \frac{L}{2} \rceil$ configurations from $\Omega$
for $t = 1$ to $T$ do
    $\omega \longleftarrow$ uniformly sample $\lceil \frac{L}{2} \rceil$ configurations from $\Omega$
    $\omega \longleftarrow$ exchange$(\omega, \Omega)$
    if compute$O_D$($\omega$, $\boldsymbol{\theta}$) > compute$O_D$($\omega^*$, $\boldsymbol{\theta}$) then
        $\omega^* \longleftarrow \omega$
$\boldsymbol{\theta} \longleftarrow$ solve Eq. 9 for $\omega^*$
while $|\omega^*| < N^*$ do
    $q^+ \longleftarrow \arg\max_{q \in \Omega \setminus \omega^*}$ compute$O_D$$(\omega^* \cup \{q\}, \boldsymbol{\theta})$
    $\omega^* \longleftarrow \omega^* \cup \{q^+\}$
    $\boldsymbol{\theta} \longleftarrow$ solve Eq. 9 for $\omega^*$
```

Algorithm 2: exchange: Optimizes a set ω of N configurations for calibration from a pool Ω

```
Input:  Pool of configurations $\Omega$
        Initial configurations $\omega = \{q_1, \dots, q_N\}$
        Calibration parameters $\boldsymbol{\theta}$
Output: Optimized set of configurations $\omega^*$ of size $N$
repeat
    $q^+ \longleftarrow \arg\max_{q \in \Omega \setminus \omega^*}$ compute$O_D$$(\omega^* \cup \{q\}, \boldsymbol{\theta})$
    $\omega^* \longleftarrow \omega^* \cup \{q^+\}$
    $q^- \longleftarrow \arg\max_{q \in \Omega \setminus \omega^*}$ compute$O_D$$(\omega^* \setminus \{q\}, \boldsymbol{\theta})$
    $\omega^* \longleftarrow \omega^* \setminus \{q^-\}$
until $q^+ = q^-$
```

Consequently, we choose the configurations for the calibration, and thereby the entries of J, such that the chosen observability index is maximized. How to select such a set of N^* configurations is explained exemplary in the following and is shown in Alg. 1 and Alg. 2 for the index O_D. Here, the function computeO_D(ω, $\boldsymbol{\theta}$) computes O_D from a set of configurations $\omega \subset \Omega$ and $\boldsymbol{\theta}$. That means the Jacobian J is computed around the current estimate for $\boldsymbol{\theta}$ from the partial derivatives of all error functions as in Eq. 11, corresponding to the set of configurations ω. Alg. 1 initially samples a set of $\lceil L/2 \rceil$ configurations from the pool Ω, as this is the minimum number of configurations needed to solve the optimization problem. We randomly sample the configurations proportional to the degrees of freedom of the corresponding chains. This is to ensure that all the parameters can be determined. Each sampled set is then optimized by exchanging its configurations (see Alg. 2), such that the observability index is maximized. Then, Alg. 1 keeps the best optimized set of configurations ω^* and optimizes for $\boldsymbol{\theta}$. Finally, additional configurations are added to ω^* in a greedy fashion, while iteratively solving for $\boldsymbol{\theta}$ given the current set of parameters.

It has to be noted that the observability indices relate singular values whose units depend on the parameters that are used to calculate the corresponding Jacobian. This can lead to a bad choice of configurations if the units are differently scaled. Hence, our approach compensates for this problem by scaling the columns of the Jacobian [15]. Therefore, prior to computing computeO_D, the algorithm right-multiplies J by a scaling matrix $S = \mathsf{diag}(s_1, \dots, s_L)$, where

$$s_i = \begin{cases} \|\mathbf{j_i}\|^{-1} & \text{if } \|\mathbf{j_i}\| \neq 0 \\ 1 & \text{else} \end{cases}, \tag{33}$$

and $\mathbf{j_i}$ indicates the i$^{\text{th}}$ column of J. This approach has the advantage that the user does not need to provide a range for the parameters, which can sometimes be hard to determine (e.g., for camera intrinsics or translations).

V. Experiments

In this section, we present an extensive evaluation of our self-calibration method using a Nao humanoid robot. The robot is approx. 60 cm in height and has a total of 23 joints, one of which is a mimic joint. Our system respects mimic joints by only considering one of the joints in the configuration selection and optimization. Five of the joints are redundant wrt. the camera pose and the markers, i.e., the head pitch joint and the last joints for each EEF. Their offsets cannot be determined and thus, are left out in the calibration. In addition to the joint offsets, we estimate the four marker positions (3 DOF each), the camera pose (6 DOF), its intrinsics (4 DOF) and one radial distortion parameter. Hence, there is a total of 41 parameters to be estimated and at least 21 measurements are needed for a full calibration. Our algorithm sets the initial state for the calibration according to the manufacturer specifications (i.e., zero) for the joint offsets, while we quickly calibrated the camera with the standard OpenCV calibration routine. Furthermore, we manually measured the placement of the markers relative to the end-effectors.

To allow a quantitative evaluation of our system, we generated a large database of 3000 configurations, observing each of the robot's four end-effectors 750 times, using the sampling method described in Sec. IV-A and then recorded the camera observations and joint encoder readings. We performed a 5-fold cross-validation on the data set to evaluate the system, including the automatic configuration selection.

A. Pose Selection

To evaluate our approach to automatic configuration selection as described in Sec. IV, we compared the results from the O_D-select algorithm with a strategy that randomly selects configurations from the pool Ω. For the random strategy, we repeated the complete 5-fold cross-validation three times and accumulated the results to obtain more representative values. The results are shown in Fig. 3 (top) as root mean square error (RMSE) and its standard deviation over the folds (and repetitions in case of the random strategy). As can be seen, our O_D-select algorithm leads to accurate results with an RMS of 7.2 ± 1.1 px requiring only 25

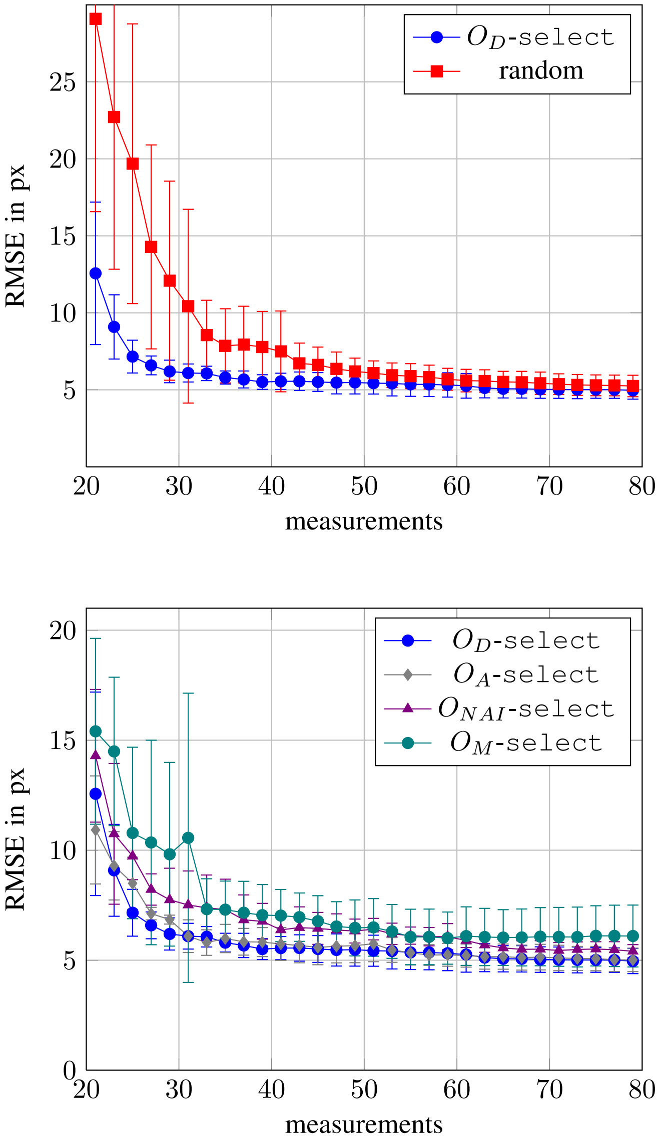


Fig. 3. Error on validation data. Top: comparison of O_D-select with a random selection strategy. Bottom: comparison of different observability criteria.

configurations, which is close to the theoretic minimum of 21. The random strategy yields a substantially higher RMS of 19.7 px with a large standard deviation of 9.1 px. Even for 50 configurations, O_D-select performs better with an RMS of 5.5 ± 0.8 px compared to 6.1 ± 0.8 px.

We also compared the different observability indices introduced in Sec. IV. The results are shown in Fig. 3 (bottom). As can be seen, the O_D-select performs better than the other algorithms for a small number of measurements, with O_A-select performing similar for more than 32 configurations. Both algorithms generally perform better than the remaining two, even for a large numbers of measurements.

Finally, we compared the performance for arms and legs separately, as some users might want to calibrate only subparts of the humanoids body. The results are shown in Fig. 4. As can be seen, in each case, it is advisable to use a selection algorithm. For the arms, we obtained slightly better results compared to the legs, which is due to the better observability of the former, as the legs typically need to be in front of the robot to be seen by the camera, which means that the configurations do not cover the whole parameter space.

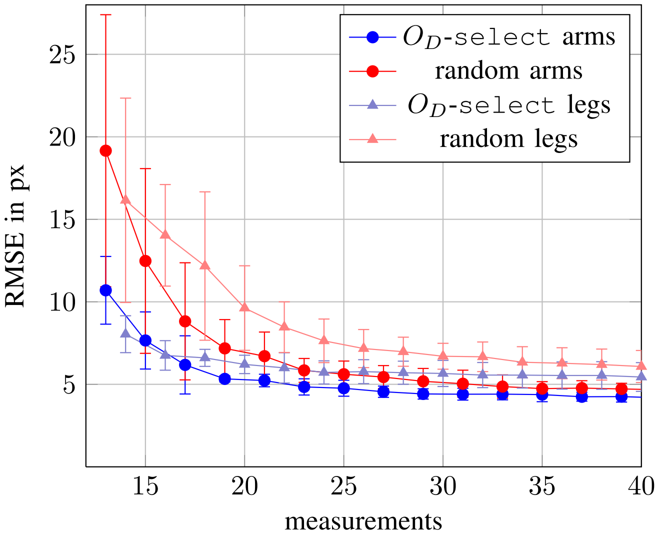


Fig. 4. Errors on validation data. Comparison of O_D-select with a random selection strategy for arms and legs separately.

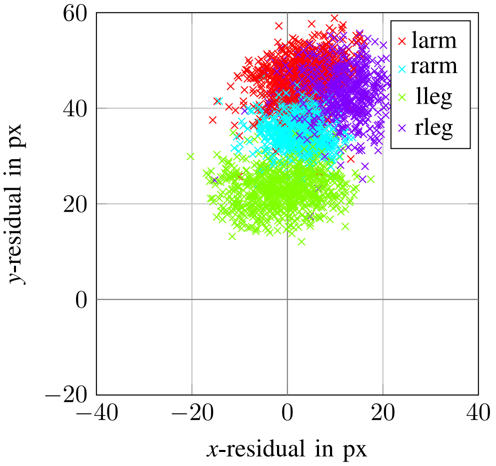
The complexity of the O_D-select is governed by three variables: the pool size $|\Omega|$, the number of initial samples T, and the number of configurations N^*. The computation of the initial set, involving Alg. 2, takes 2.5 s per iteration for $|\Omega| = 1500$ on an i7 quad-core CPU. In the experiments, we set $T = 50$. For $|\Omega| = 1500$, selecting a new configuration takes 5 s on average, and 19 s for $|\Omega| = 3000$. Solving Eq. 9 takes between 2 s (for $N^* \leq 15$) and 15 s (for $N^* \leq 60$).

B. Effect of the Calibration Compared to the Initial State

Fig. 5 shows the residuals in the camera for each of the four chains. The top plot depicts the residuals for the uncalibrated state according to the manufacturer specifications with a good initial guess for the markers. The lower plot shows the residuals after the calibration using 60 configurations selected by O_D-select. This plot overlays the residuals from each of the 5-folds of the data set. Initially, the error was 43.4 px. By calibration we achieved an RMS of 5.2 px.

C. Effect of Measurement Noise

Finally, we tested the effect of noisy measurements on the optimization. Therefore, we used 60 configurations selected by O_D-select and repeated the optimization, but this time we randomized the detected marker location over the entire image for different percentages of measurements. This is to simulate the effect of false marker detections, which often occurs in presence of clutter and uncontrolled lighting. We optimized using the noisy data with and without using a robust kernel and performed a 5-fold cross-validation as previously. The results are shown in Fig. 6. As can be seen, using a robust kernel yields substantially better results compared to a quadratic kernel. For up to 10 noisy measurements out of 60 total, the robust kernel performs



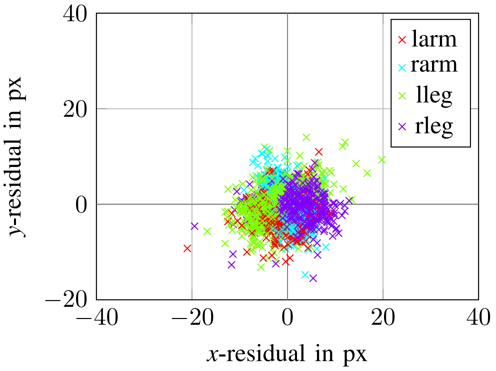


Fig. 5. Comparison of measurement residuals before (top) and after (bottom) calibration. The bottom plot show the residuals for the union of the evaluation data from a 5-fold cross-validation.

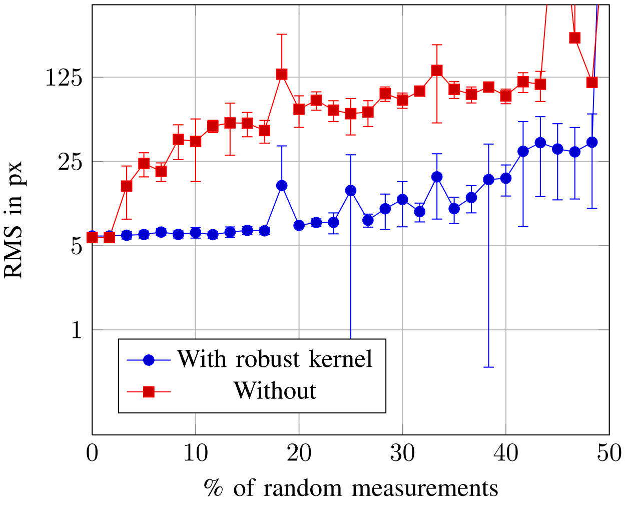


Fig. 6. Effect of noise on the optimization. Different percentages of camera observations were randomized in the image and optimized with and without a robust kernel. The plot shows the results of a 5-fold cross validation.

similar to using noise-free measurements (6.6 px RMS), whereas without robust kernel, the RMS rises up to 45.1 px. Concluding, the robust kernel handles up to 16% of noisy measurements without larger performance drops and clearly outperforms the standard kernel.

VI. Conclusions

We developed new techniques for automatic calibration of the complete kinematic model for legged humanoids. Our system relies on least-square minimization of the sum of residuals between camera observations of markers and their expected locations in the image. Our approach automatically chooses robot configurations that enable an accurate, robust calibration by minimizing the variance of the parameters to be estimated. The calibration framework is general and can be used for other robots with a different kinematic model and with different types and numbers of cameras. The ROS-compatible software is available online[1]. In extensive experiments with a real Nao humanoid, we showed that our framework requires only few observations to obtain accurate calibration results. In future work, the framework can be extended such that the robot constantly monitors the error metric in Eq. 11 and, when it exceeds a threshold, automatically initiates a re-calibration.

References

[1] K. Yamane. Practical kinematic and dynamic calibration methods for force-controlled humanoid robots. In *IEEE Int. Conf. on Humanoid Robots (Humanoids)*, 2011.

[2] I.-W. Park, B.-J. Lee, S.-H. Cho, Y.-D. Hong, and J.-H. Kim. Laser-based kinematic calibration of robot manipulator using differential kinematics. *IEEE/ASME Trans. on Mechatronics*, 2012.

[3] T. Asfour, K. Welke, P. Azad, A. Ude, and R. Dillmann. The karlsruhe humanoid head. In *IEEE Int. Conf. on Humanoid Robots (Humanoids)*, 2008.

[4] A. Traslosheros, J. Sebastian, E. Castillo, F. Roberti, and R. Carelli. One camera in hand for kinematic calibration of a parallel robot. In *IEEE Int. Conf. on Intelligent Robots and Systems (IROS)*, 2010.

[5] V. Pradeep, K. Konolige, and E. Berger. Calibrating a multi-arm multi-sensor robot: A bundle adjustment approach. In *Int. Symp. on Experimental Robotics (ISER)*, 2010.

[6] O. Birbach, B. Bäuml, and U. Frese. Automatic and self-contained calibration of a multi-sensorial humanoids upper body. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2012.

[7] U. Hubert, J. Stückler, and S. Behnke. Bayesian calibration of the hand-eye kinematics of an anthropomorphic robot. In *IEEE Int. Conf. on Humanoid Robots (Humanoids)*, 2012.

[8] J.-H. Borm and C.-H. Menq. Experimental study of observability of parameter errors in robot calibration. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 1989.

[9] T. Li, K. Sun, Y. Jin, and H. Liu. A novel optimal calibration algorithm on a dexterous 6 dof serial robot-with the optimization of measurement poses number. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2011.

[10] T. Li, K. Sun, Z.-w. Xie, and H. Liu. Optimal measurement configurations for kinematic calibration of six-DOF serial robot. *Journal of Central South University of Technology*, 18, 2011.

[11] H. Carrillo, O. Birbach, H. Taubig, B. Bauml, U. Frese, and J. Castellanos. On task-oriented criteria for configurations selection in robot calibration. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2013.

[12] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard. g2o: A general framework for graph optimization. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2011.

[13] J. Pan, S. Chitta, and D. Manocha. FCL: A general purpose library for collision and proximity queries. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2012.

[14] A. Nahvi and J. Hollerbach. The noise amplification index for optimal pose selection in robot calibration. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 1996.

[15] J. M. Hollerbach and C. W. Wampler. The calibration index and taxonomy for robot kinematic calibration methods. *Int. Journal of Robotics Research (IJRR)*, 1996.

[1] `http://github.com/danielmaier/nao_calibration`