Robotics and Autonomous Systems 61 (2013) 580–592

Contents lists available at SciVerse ScienceDirect

Robotics and Autonomous Systems

journal homepage: www.elsevier.com/locate/robot

Joint origin identification of articulated robots with marker-based multi-camera optical tracking systems

Nadia B. Figueroa [a,*], Florian Schmidt [b], Haider Ali [b], Nikolaos Mavridis [a]

[a] Department of Engineering, New York University Abu Dhabi (NYUAD), Abu Dhabi, United Arab Emirates

[b] Institute of Robotics and Mechatronics (RM), German Aerospace Center (DLR), Oberpfaffenhofen, Germany

HIGHLIGHTS

- We use marker-based multi-camera optical tracking systems to verify kinematic chains of articulated robots.
- The identification of the mounted tracking bodies relative to the joints of an articulated robot is essential for an accurate representation of the kinematic model.
- We identify the origin of joints relative to tracking bodies by using state-of-the-art center of rotation (CoR) and axes of rotation (AoR) estimation methods.
- The applicability of our method is proven with two case studies.

ARTICLE INFO

Article history:
Received 9 October 2012
Received in revised form
12 February 2013
Accepted 22 February 2013
Available online 14 March 2013

Keywords:
Joint identification
Marker-based multi-camera optical tracking system
Calibration
Articulated robots

ABSTRACT

Marker-based multi-camera optical tracking systems are being used in the robotics field to track robots for validation, verification, and calibration of their kinematic and dynamic models. These tracking systems estimate the pose of tracking bodies attached to objects within a tracking volume. In this work, we explore the case of tracking the origins of joints of articulated robots when the tracking bodies are mounted on limbs or structures relative to the joints. This configuration leads to an unknown relative pose between the tracking body and the joint origin. The identification of this relative pose is essential for an accurate representation of the kinematic model. We propose an approach for the identification of the origin of joints relative to tracking bodies by using state-of-the-art center of rotation (CoR) and axis of rotation (AoR) estimation methods. The applicability and effectiveness of our approach is demonstrated in two successful case studies: (i) the verification of the upper body kinematics of DLR's humanoid *Rollin' Justin* and (ii) the identification of the kinematic parameters of an ST Robot arm relative to its environment for the embodiment of a situated conversational assistant.

1. Introduction

In the ongoing field of robotics research, high-precision marker-based multi-camera optical tracking systems, such as the VICON Motion Capture System[1] or the Infrared Optical Tracking System from ART Advanced Realtime Tracking GmbH,[2] are being used for validation, verification, and calibration of robotic systems and applications. The 3-DoF/6-DoF (degree of freedom) pose of the robot or limb is localized with respect to (w.r.t.) the world coordinate frame of the tracking system by placing markers or *rigid bodies* (sets of markers) on the robotic structures. In the case of autonomous mobile robots (ground and aerial), researchers use the positions and orientations of the robots to improve control and path planning algorithms [1–3]. Regarding articulated robots and bipeds, the localization of the positions of the limbs or joints is used to evaluate positioning errors due to lightweight non-rigid structures, as well as the validation, verification, and simulation of the dynamic and kinematic models for motion control [4,5].

The method used for identifying the pose of markers or rigid bodies mounted on a robot – relative to their joints or center of motion – depends on the complexity of the kinematic model of the robot at hand. Mobile robots are modeled as a single rigid body on wheels with three DoFs (x, y, θ) (i.e., their kinematic model consists of only one origin or joint); thus the center of motion of the robot can be easily identified using the robot's embodiment [6]. For example, for non-holonomic mobile robots (such as the Pioneer

* Corresponding author. Tel.: +971 56 7336784.
E-mail addresses: nbfigueroa@gmail.com, nadia.figueroa@nyu.edu (N.B. Figueroa), florian.schmidt@dlr.de (F. Schmidt), haider.ali@dlr.de (H. Ali), nmav@alum.mit.edu (N. Mavridis).

[1] VICON Motion Capture System. http://www.vicon.com/index.html.

[2] ARTrack tracking system. http://www.ar-tracking.com/home/.

0921-8890/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.robot.2013.02.008

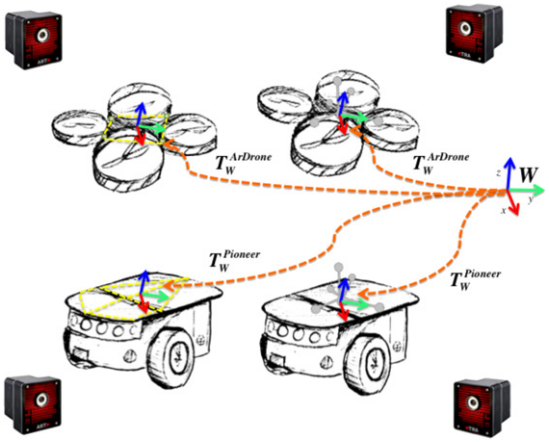


Fig. 1. The origins ($T_W^{Pioneer}$ and $T_W^{ArDrone}$) represent the 6-DoF pose of the center of the robotic platform w.r.t. the world coordinate frame (W) of the multi-camera optical tracking system.

robotic platform,[3]) their center of motion is a point centered between the two drive wheels. Therefore, the origin of these types of robot is identified with the tracking system either by (i) placing markers around the mobile robot's perimeter and computing the center or (ii) placing a rigid body between the wheels (Fig. 1). For flying robots (such as the quadrotor ARDrone[4]), the procedure for finding the center of motion is analogous to that of mobile robots. The kinematics of a quadrotor are modeled as a point mass centered at the crossing of the four rotor structures with six DoFs (x, y, z, roll, pitch, yaw) [7,8]. Thus, the origin of the quadrotor is identified with the tracking system either by (i) placing a rigid body in the crossing of the rotors or (ii) placing markers in each rotor structure and computing the center (Fig. 1).

In the case of articulated robots, their kinematic model is a chain of rigid bodies connected by joints (i.e., a kinematic chain) [9]; hence the use of tracking systems is more involved than it is for mobile robots. For articulated robots, the goal is not only to track the global pose of the robot, but also to track individual joints for the verification and calibration of the robot's kinematic model. Hofschulte et al. [5] mounted rigid bodies directly on the joints of LISA the biped robot to analyze and simulate its walking behavior. Gashler [4] rigidly attached markers on the ball and socket of the joints of the compliant ECCE humanoid robot to estimate its unknown kinematic model. In these examples, the robots have no outer frame; in other words, the joints and limbs are not covered (similar to a skeleton). Therefore, the tracking bodies are directly mounted on the joints or near the joints. In the latter case, the relative pose between the tracking body and the joint is measured with a ruler or measuring tape. In this work, we tackle a more complex problem, which is tracking the individual joints of an articulated robot whose joint origins are encapsulated within the robot's structure/frame and whose tracking bodies are mounted on limbs or structures relative to the joints. As seen in Fig. 2, the origins of the hand (T_W^{hand}) and head (T_W^{head}) of the complex humanoid mobile robot *Rollin' Justin*[5] are within the robotic embodiment (i.e., within the flange and pan–tilt unit). This can also be seen in robots with simpler kinematic chains, such as the R17 robot arm from ST Robotics[6] (Fig. 3) (i.e., the origin of the base (T_W^{base}) and the origin of the last joint (T_W^{endJ})).

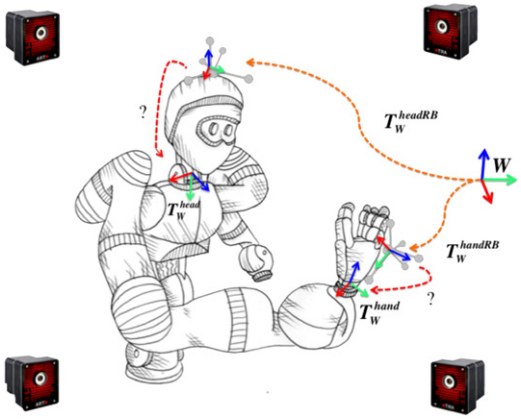


Fig. 2. Humanoid robot *Rollin' Justin*. (Unknown pose between T_W^{headRB}–T_W^{head} and T_W^{handRB}–T_W^{hand}.)

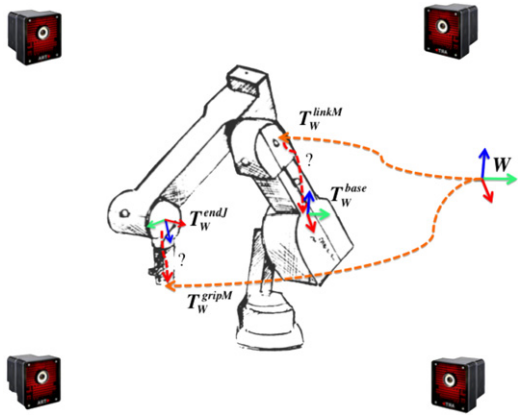


Fig. 3. ST Robot arm R17. (Unknown pose between T_W^{linkM}–T_W^{base} and T_W^{gripM}–T_W^{endJ}.)

The proposed approach to solve this problem is to use the mechanical properties of the joints, namely the motion that they generate with the attached limbs. Most articulated robots are constructed with revolute joints, which are modeled as a single DoF that rotates around an axis [9]. In complex kinematic chains, the combination of these joints produces a multi-DoF joint [9]. For example, a redundant 7-DoF kinematic chain, such as the arm of *Rollin' Justin*, produces a 3-DoF joint at the end-effector [10]. Also, the intersection of the rotating base and the shoulder of a simple 6-DoF manipulator produces a 2-DoF joint. These multi-DoF joints generate a spherical workspace [9]. Therefore, by placing markers or rigid bodies on the moving limbs attached to these multi-DoF joints, measurements with a sphere-like shape are obtained. The center of these sphere-like shapes is the origin of the joints. Moreover, the axis of rotation of each DoF represents the orientations of the 6-DoF frame of the origin of the multi-DoF joint. The axis of rotation is the normal vector of the planar/circular trajectory of measurements produced by the individual motion of each DoF. This approach derives from work from the biomechanics community, namely that of Gamage [12], Cerveri [13], Halvorsen [14], and Chang and Pollard [15,16], who proposed center of rotation (CoR) and axis of rotation (AoR) estimation methods for human joints such as knees, wrists, and ankles.

The main contribution of this paper is the identification of the origins of individual joints of articulated robots, by estimating their CoR and AoR, without any physical assumptions of the positioning of the markers or rigid bodies w.r.t. the joints. The CoR/AoR estimation methods are adapted from the biomechanics community; they are used to find the origins of anatomical

[3] Mobile robots. http://www.mobilerobots.com/Mobile_Robots.aspx.

[4] ARDrone Parrot. http://ardrone2.parrot.com.

[5] *Rollin' Justin* of the Institute of Robotics and Mechatronics at DLR (German Aerospace Center) [10,11].

[6] STRobotics http://www.strobotics.com/.

joints. Furthermore, a systematic approach is introduced for using tracking systems on articulated robots that can provide unambiguous measurements for applications such as (i) validation and verification of kinematic chains, (ii) calibration of end-effectors to robotic arms, and (iii) identification of relative poses of external objects to certain joints of the robot.

The organization of this paper is as follows. Section 2 presents an overview of techniques used in different fields to identify the pose of markers or rigid bodies relative to a certain point or joint using geometrical fitting methods. In Section 3, a description and evaluation of the two main sphere-fitting methods for computing the CoR of the joints is presented. Section 4 describes and evaluates the plane-/circle-fitting method used to estimate AoRs. In Sections 5 and 6 two case studies are presented. In the first case study, a marker-based pose estimation method using the ARTrack IR Optical tracking system for DLR's *Rollin' Justin* is presented. This method was used to evaluate a marker-less verification routine for the identification of the maximum bounds of the TCP (tool center point) end-pose errors of *Justin* by using the on-board stereo vision system, implemented by Figueroa et al. [17]. The second case study deals with the identification of the kinematic parameters (and their relative poses to the environment) of an ST Robot arm used to embody a new version of the situated conversational assistant (*Ripley*) from the work of Mavridis [18,19] using a VICON Motion Capture (MoCap) system. Finally, Section 7 presents the conclusions of this work.

2. Background

The identification of the pose of a marker or rigid body w.r.t. to a certain point in a world coordinate frame is a known problem in both the augmented reality and biomechanics communities. Several methods have been developed to solve it for specific applications. For example, Tuceryan et al. [20] introduce the *hot spot calibration* technique to find the geometry of a pointing device used by the GRASP augmented reality system [21]. This pointer calibration calculates the position of the tip of the pointing device (*i.e., a digital wand or pen*) relative to a mounted tracking target, in other words, the transformation between the coordinate system of the tracking target and the coordinate system of the tip of the pointing device. The calibration consists of fixing the tip of the pointing device to a certain point and acquiring n measurements of the tracking target with n different orientations. An overdetermined system of equations is constructed by using the measurements made from reading a point at n different orientations. This system of equations can be solved using a least-squares method. Fuhrmann et al. [22] use a similar approach to calibrate a pointing device used in the Studierstube augmented reality system [23]. They fix the pointing device in a small pit drilled in a table, and move the tracking target on a hemisphere. The acquired tracking target measurements are fitted to a sphere, and the estimated center of the sphere is the position of the tip of the pointing device w.r.t. the tracking target. This is estimated by optimizing the offset vector from the tracking target to the tip of the pointing device. The optimization of this vector yields a least-squares-fit solution. To enforce stability on this solution, the measurements should cover a large part of the hemisphere.

The hot spot calibration approach has also been extended to medical augmented reality applications. Sielhorst et al. [24] presented CAMPAR, a framework for integrating multiple tracking and visualization systems for medical augmented reality applications. The tools used in a medical application could be drills, probes, needles, etc. These tools are calibrated to the augmented reality systems using this approach.

In biomechanics, the modeling of joint kinematics using non-invasive measurements is a key application for motion analysis [12]. The anatomical joints of human bodies are modeled as spherical joints or as rotational joints with a fixed axis of rotation [14]. To describe motion determined by joint angles, the CoR or origin of the joint needs to be approximated. This CoR is estimated by the relative motion of adjacent body segments (i.e., measured by markers or rigid bodies mounted on the moving anatomical parts) [15]. This is directly applicable to the problem at hand. The most common solutions to estimating the CoR proposed by the biomechanical community rely on the same assumption as in hot spot calibration, namely that the measurements from the markers or rigid bodies trace out a sphere centered at the joint's CoR. Halvorsen [14], Gamage and Lasenby [12], and Chang and Pollard [15], to name a few, have proposed least-squares methods for the sphere-fitting problem. The main differences between them are the cost functions, the weighting factor, and geometrical assumptions. Chang and Pollard [15] present an interesting review on the existing sphere-fitting methods and propose an improved method that yields an exact solution and is robust to joints with a small range of motion (RoM).

The distinction between the hot spot calibration and the *CoR estimation* methods is that in the latter only the 3-DoF coordinates of the markers or rigid bodies are used, compared to the former, which needs the full 6-DoF pose. Additionally, in *hot spot calibration* there is no spherical geometrical constraint applied to the least-squares problem as in CoR estimation. In Section 3, an evaluation and comparison of the *Hot Spot Calibration* method and the CoR estimation method proposed by Chang and Pollard [15] is presented.

The estimation of the CoR provides only the translational component (x, y, z) of the origin of the joint. In order to find the full 6-DoF coordinates, estimations of the orientations of the rotational motion are needed as well. This problem is common in the biomechanics community and similar to CoR estimation. It is called axis of rotation (AoR) estimation [14]. Joints like the knees or fingers may be modeled with one AoR. Solutions for estimating the AoR of a single-DoF joint model rely on optimization or plane-fitting techniques which assume the circular motion of the tracking targets around the AoR [14,12,13]. Human joints (as well as robotic joints) exhibit more than one degree of freedom. Additional rotations and range of motion limitations due to a second or third degree of freedom are present in such types of joint. Estimating the dominant AoRs of joints with these extra limitations using the plane-fitting approach can lead to poor estimations of an AoR direction [16]. Chang and Pollard [16] propose using a combined cost function: instead of just minimizing the error along the AoR with plane fitting, they also include the error orthogonal to the AoR direction by using a circle-fitting approach. In Section 4, an evaluation and description of this AoR method adapted to articulated robot joints is provided.

3. Center of rotation (CoR) estimation

In this section the two main CoR estimation methods introduced in the previous section are evaluated (hot spot calibration and geometrically constrained sphere fitting).

3.1. Hot spot calibration

The first method used to estimate the CoR is hot spot calibration (Fig. 4). In this approach, the position of the origin (O) of the joint w.r.t. the coordinate system of the tracking system (O_W) is computed by estimating the relative position of the rigid body (O_{rb}) and the joint origin (O).

Assuming that the offset between the coordinate system of the rigid body and the origin is constant, the relation between origin

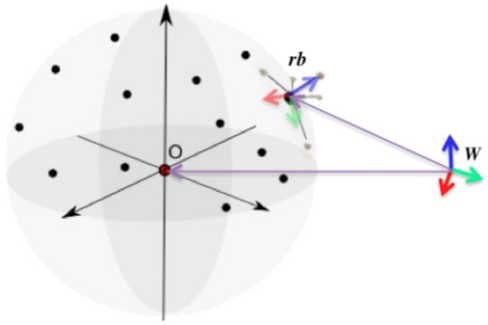


Fig. 4. Hot spot calibration.

in the tracking system coordinate frame O_W and the origin in the rigid body's coordinate system O_{rb} is given by

$$\begin{bmatrix} O_W \\ 1 \end{bmatrix} = \begin{bmatrix} R_i & t_i \\ 0 & 1 \end{bmatrix} \begin{bmatrix} O_{rb} \\ 1 \end{bmatrix}, \tag{1}$$

where R and t are the 6-DoF rotation and translation of the rigid body in the coordinate frame of the tracking system, with $i = 1, \ldots, n$ and n being the number of measurements. $O_W = (x_W, y_W, z_W)$ and $O_{rb} = (x_{rb}, y_{rb}, z_{rb})$; therefore six unknown variables exist. A linear equation system $Ax = b$ can be constructed using Eq. (1), as follows:

$$\begin{bmatrix} R_1 & -I \\ R_2 & -I \\ \vdots & \vdots \\ R_n & -I \end{bmatrix} \begin{bmatrix} O_{rb} \\ O_W \end{bmatrix} = \begin{bmatrix} -t_1 \\ -t_2 \\ \vdots \\ -t_n \end{bmatrix}. \tag{2}$$

With an overdetermined set of linear equations like in Eq. (2), x is usually solved with a least-squares formulation. Two solutions are proposed.

3.1.1. Exact solution using singular value decomposition

A is a $3n \times 6$ non-invertible matrix, so its pseudoinverse A^+ is used to solve for x, as follows:

$$x = A^+ b. \tag{3}$$

A computationally simple and accurate way to compute the pseudoinverse is by using SVD (singular value decomposition). If $A = U \sum V^*$, then

$$A^+ = V \sum{}^+ U^*. \tag{4}$$

3.1.2. Approximate solution using least-squares fitting

$Ax = b$ can be solved by computing a vector x that minimizes the Euclidean 2-norm of the sums of residuals as follows:

$$\min \|b - Ax\|^2. \tag{5}$$

3.2. Geometrically constrained sphere fitting

The second method used is the geometrically constrained sphere-fitting approach proposed by Chang and Pollard [15]. They assume that the markers maintain constant distance from the CoR, but the relative motion between the measurements is not necessarily rigid. Therefore, a non-rigid sphere-fitting approach is used. This is implemented by considering a minimal error ϵ_k in the spherical fit. This error is defined by the difference between the radius of the sphere r and the distance of the separation between the target measurement v_k and the CoR of the sphere m (Fig. 5).

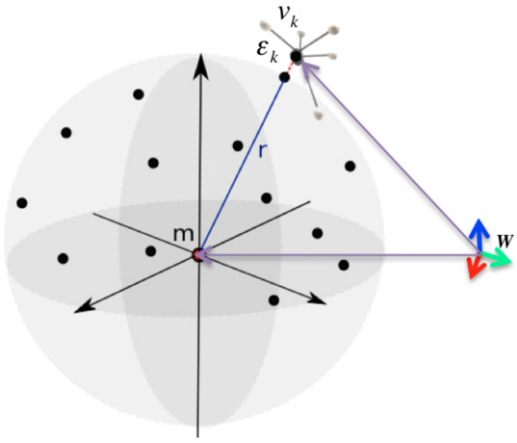


Fig. 5. Geometrical sphere-fitting error. (Illustration based on error depicted by Chang and Pollard [15].)

Gamage and Lasenby [12] model this error by the difference of squared lengths, as follows:

$$\epsilon_k = \|v_k - m\|^2 - r^2. \tag{6}$$

Following Eq. (6), Chang and Pollard [15] proposed the following constrained optimization problem:

$$\left\{ \begin{matrix} \text{minimize :} & u^T S u \\ \text{subject to :} & u^T C u = 1 \end{matrix} \right\}, \tag{7}$$

where C is a matrix form of the normalization constraint for spherical surfaces introduced by Pratt [25]. $S = D^T D$, where D is the algebraic distances between n measurements and the center of the sphere and u is the spherical fit of those measurements (see Appendix A for the derivation). Two solutions for solving this constrained minimization problem are evaluated.

3.2.1. Exact solution using eigenvalue decomposition (EVD)

This solution was proposed by Chang and Pollard [15]. They convert Eq. (7) to an unconstrained minimization problem with the following Lagrangian function:

$$L = u^T S u - \lambda(u^T C u - 1), \tag{8}$$

where λ is the Lagrangian multiplier [26,27]. By differentiating Eq. (8) w.r.t. u, the following generalized eigenvalue problem is obtained:

$$Su = \lambda C u, \tag{9}$$

where λ is the scalar eigenvalue and u is its corresponding eigenvector. Bookstein [28] and Fitzgibbon et al. [29] show that the best-fit solution for this optimization problem is the generalized eigenvector u with non-negative eigenvalue λ with the least cost according to $u^T S u$.

3.2.2. Approximate solution using constrained optimization by linear approximation (COBYLA)

An approximate solution is proposed for Eq. (7) by using a constrained optimization by linear approximation algorithm (COBYLA). COBYLA is an implementation of Powell's nonlinear derivative-free constrained optimization that uses a linear approximation approach [30]. Ten different initializations for this optimization are used. This helps in avoiding local minima and increasing the quality of the solution. The resulting vector u that yields the minimum cost of $u^T S u$ is selected as the solution.

3.3. CoR simulation and evaluation

In a real-world tracking system, like the ARTrack,[7] deviations in measurements lie within the range of 1–2 mm in translation and 0°–1° in rotation coordinates,[8] which are caused by (i) camera calibration errors, (ii) camera occlusions, (iii) environmental noise (i.e., external reflective materials in the tracking volume), or (iv) rigid body or marker movement. This range of measurement deviations is used as a reference to create synthetic data for the evaluation of each proposed CoR estimation method. Because the CoR estimation methods are essentially sphere-fitting algorithms, they can be evaluated by analyzing how well a sphere is fitted to randomly noisy spheres. Therefore, the simulated data is synthetic rigid-body measurements produced by a sphere-like surface with a range of minimal to extreme translational and rotational errors.

These noisy sphere-like measurements are simulated by adding random normally distributed errors with zero mean and a standard deviation of 1 mm (minimum)/2.5 mm (*real*)/5 mm (*extreme*) to the translational components of the synthetic sphere measurements and 0°–5° to the rotational components. The stability and performance of each method are evaluated by analyzing their behavior throughout the range of minimal to extreme errors. Three different numbers of measurements (63, 16, and 9) are used for the simulation. These numbers are chosen to uniformly sweep the RoM of two rotation axes of a simulated joint. The following methods are evaluated.

- HSE: Hot spot method using SVD.
- HSA: Hot spot method using approximate least squares.
- SFE: Geometrically constrained sphere-fitting method using EVD.
- SFA: Geometrically constrained sphere-fitting method using COBYLA.

The hot spot calibration methods yield almost identical results throughout all of the simulations. The two methods (HSE, HSA) show a stable behavior with low rotational error; however, the higher the rotational error of the measurements the higher the CoR error. This is obvious since this estimation method depends on the 6-DoF measurement of the tracking targets. The SFA method is very unstable: it can yield the best results as well as the worst-case results. The SFE method is the most stable of all: it is not affected by the rotational errors on the measurements, and even with the least number of measurements (9) it shows a mean error of 0.27 mm for minimum measurement errors. The SFE method is then further evaluated by computing the fitting error between the estimated CoR from the simulated noisy data and the nominal CoR for all the previously simulated scenarios. The maximum identified fitting error from the simulations is of 1 mm (for the 16 measurements under extreme noise). However, for the real noise level simulations it shows a fitting error of approximately 0.2 mm for all numbers of measurements. Therefore, this method is expected to have a deviation of 0.2–1 mm for CoR estimations with real data.

4. Axis of rotation (AoR) estimation

As mentioned earlier, the common approach for estimating the AoR of a joint is by creating a circular set of measurements around the AoR and minimizing their motion along the axis direction. The approach proposed by Halvorsen [14] estimates the combined CoR and AoR as the line which best describes the collection of instantaneous rotational axes generated from pairs of measurements; the performance of this method depends highly on the choice of separation distance between the measurements. Gamage [12] and Cerveri [13] both propose a two-step procedure: (i) estimate the CoR by using sphere-fitting techniques and (ii) find the CoR direction (AoR) by plane fitting those measurements along the axis direction. Chang and Pollard [16] improve this approach by combining plane fitting and circle fitting in their minimization function. They show that their method is more appropriate for practical applications where the joints exhibit multiple degrees of freedom and the assumption that the joint will generate a perfect circular rotation around an axis does not hold. This approach consists of minimizing a cost function that models how well the measurements v_k maintain a fixed distance from the CoR m (i.e., minimal deviation from circle/sphere radius r) and remain on a plane orthogonal to the AoR (Fig. 6). This is estimated as an error vector e_k with two components.

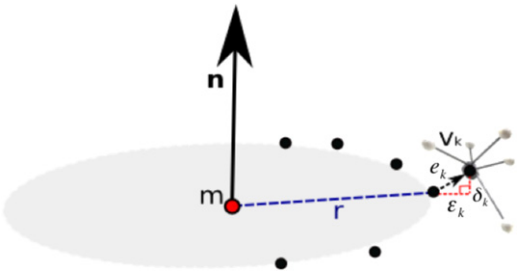


Fig. 6. AoR estimation (measurement v_k is constrained to a circle with radius r on a plane normal to n with error e_k).

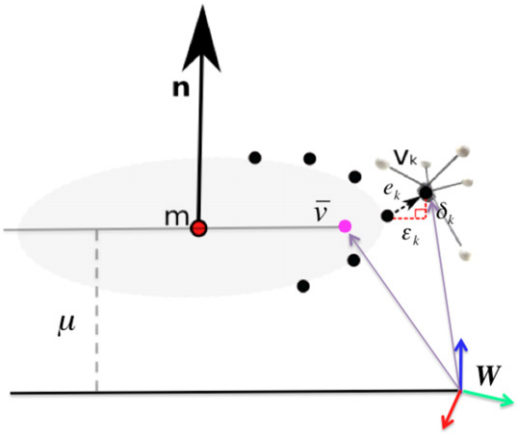


Fig. 7. Planar error estimation: the plane is modeled with normal $\boldsymbol{n}$ and a nominal distance μ from the plane to the tracking system world coordinate frame (*measurement* v_k *is described in terms of the offset* μ *and centroid* $\bar{v}$).

- Planar error δ_k: the magnitude of e_k parallel to the AoR.
- Radial error ϵ_k: the magnitude of e_k in the plane orthogonal to the AoR.

When considering the planar error δ_k and the radial error ϵ_k individually, this would model plane fitting and cylinder fitting, respectively. Therefore, Chang and Pollard [16] propose the following combined cost function:

$$f = \sum_{k=1}^{N} (\delta_k^2 + \epsilon_k^2). \tag{10}$$

The planar error is modeled by calculating the difference between the component of v_k parallel to the normal of the plane $\boldsymbol{n}$ and the nominal distance of the plane to the world coordinate system μ

[7] ARTrack tracking system. http://www.ar-tracking.com/home/.

[8] These measurement deviations are obtained empirically from a mounted Real-Time ART Tracking system using four cameras with a mean accuracy of 0.04 pixels and a system speed of 100 fps.

(Fig. 7). The planar cost function is given by the following equation:

$$f_p = \sum_{k=1}^{N} (\delta_k)^2 = \sum_{k=1}^{N} (u_k \cdot \boldsymbol{n})^2, \tag{11}$$

where $u_k = v_k - \overline{v}$ are the marker measurements expressed relative to the centroid of their trajectories (see Appendix B.1 for derivation). The radial error measures the deviation between (i) the distance of the measurement v_k to the CoR m and (ii) the radius r of the best-fitting cylinder whose axis of symmetry is the direction of the CoR $\boldsymbol{n}$. This cost function is formulated as follows:

$$f_r = \sum_{k=1}^{N} (\epsilon_k)^2 = \sum_{k=1}^{N} (\|w_k\| - r)^2, \tag{12}$$

where $w_k = (I - \boldsymbol{n}\boldsymbol{n}^T)(v_k - m)$ is the shortest vector from the target measurement to the line of the CoR m (see Appendix B.2 for derivation). Finally, the combined planar–radial cost function is constructed, as follows:

$$f = f_p + f_r = \sum_{k=1}^{N} (\delta_k)^2 + \sum_{k=1}^{N} (\epsilon_k)^2 \tag{13}$$

$$f = \boldsymbol{n}^T S \boldsymbol{n} + \sum_{k=1}^{N} (\|(I - \boldsymbol{n}\boldsymbol{n}^T)(v_k - m)\| - r)^2, \tag{14}$$

where S is the matrix form representation of $(u_k)^2$. We solve for $\boldsymbol{n}$ with an iterative solution, the downhill simplex method [31]. The final equation for the AoR is

$$AoR = m + \tau \boldsymbol{n}. \tag{15}$$

This is a straight line passing through m in the direction of n with a scalar τ. τ is a scalar constant proposed by Gamage and Lasenby [12] that defines the centers of the circles traced out by the measurements. This constant is computed as follows:

$$\tau = (\overline{v} - m) \cdot \boldsymbol{n}. \tag{16}$$

The AoR estimation method is simulated with the RoM and kinematic limitations of a hand joint. Circular rotations around the z-axis and around the x-axis are simulated by applying the same three noise levels applied to the CoR simulations. The robustness of the AoR estimation is evaluated using different numbers of measurements for each circular trajectory (19, 10, and 6 measurements). The z-rotation estimation showed a maximum error of 0.009 rad (0.5°) and the x-rotation estimation showed a maximum error of 0.012 rad (0.6°). Overall the AoR estimation for the z-axis has a mean error of 0.2° and for the x-axis of 0.4°.

5. Case study 1: verification of humanoid *Justin's* upper body kinematics

As mentioned in Section 1, a marker-based verification routine is implemented with the ART tracking system to evaluate a marker-less verification routine for the identification of the maximum bounds of the TCP end-pose errors of DLR's *Justin* [17]. The routine consists on estimating the end-pose of *Justin's* hand w.r.t. its camera reference frame relative to *Justin's* head. This is done by estimating the pose of 3D point clouds of the hand. In this work, the pose is estimated by mounting rigid bodies on the hand and head of *Justin*.

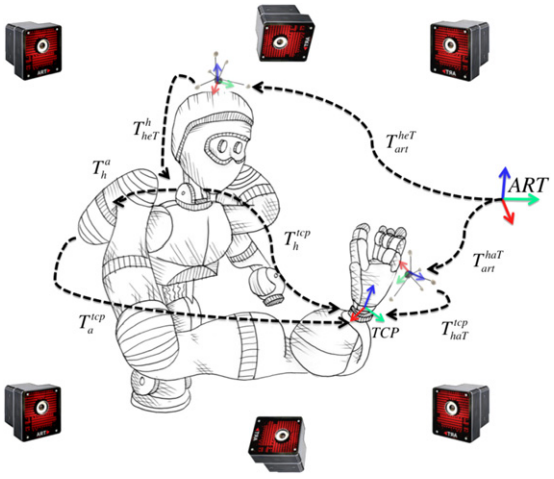


Fig. 8. *Justin's* implicit loop closure using the ART tracking system.

5.1. ART set-up for pose estimation

Justin's lab has a mounted calibrated system of six tracking cameras from ARTrack tracking systems GmbH,[9] with a mean camera accuracy of 0.04 pixels and speed of 100 fps for pose computation. *Justin* is positioned at the center of the tracking volume (Fig. 8). Two rigid bodies are mounted on *Justin*: (i) on his head (*heT*) and (ii) on his hand (*haT*).

T^h_{art} is the transformation of the tracking system's base coordinate system to the head joint.

$$T^h_{art} = T^{heT}_{art} T^h_{heT}. \tag{17}$$

T^{TCP}_{art} is the transformation of the tracking system's base coordinate system to the TCP pose of the hand.

$$T^{TCP}_{art} = T^{haT}_{art} T^{TCP}_{haT}. \tag{18}$$

The coordinate frames T^{heT}_{art} and T^{haT}_{art} are the 6-DoF pose of the rigid bodies obtained by the tracking system. T^h_{heT} and T^{TCP}_{haT} are their transformations to their respective joint coordinate system. Once T^h_{art} and T^{TCP}_{art} are obtained, the pose of the TCP w.r.t. the head coordinate system T^{TCP}_h can be estimated.

$$TCP_{art} = T^{TCP}_h = (T^h_{art})^{-1} T^{TCP}_{art}. \tag{19}$$

T^{TCP}_h can also be computed by the forward kinematics between the head joint and the arm kinematic chain.

$$TCP_{fk} = T^{TCP}_h = T^a_h T^{TCP}_a. \tag{20}$$

Thus, an implicit loop closure of *Justin's* upper body kinematics between the measured pose using the tracking system TCP_{art} and the measured pose from forward kinematics TCP_{fk} can be generated. *Justin's* head is used as the reference coordinate system to close the sensory system kinematic chain with the arm kinematic chain. T^a_h is the transformation of the head to the arm base, computed by a simple forward kinematic model. T^{TCP}_a is the transformation from the arm base to the TCP, computed by a forward kinematic model considering the measured torques and gear stiffness. Ideally $TCP_{fk} := TCP_{art}$; however, due to unidentified errors in the kinematic chain this equality does not hold, and the errors between them must be identified. The estimation of TCP_{art} is quite straightforward when T^h_{heT} and T^{TCP}_{haT} are known. This is not the case. It cannot be assumed that the rigid bodies

[9] ARTrack tracking system. http://www.ar-tracking.com/home/.

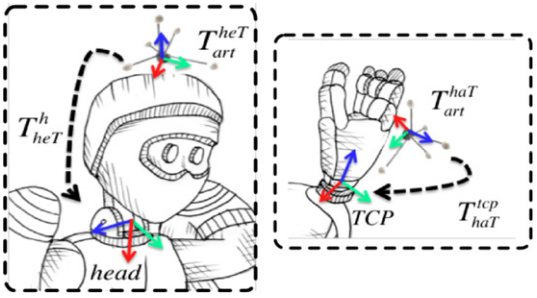


Fig. 9. Close-up view to the rigid bodies relative to joint coordinate systems. (Left) head rigid body (*heT*) relative to *head* coordinate system and (right) hand rigid body (*haT*) relative to arm *TCP* coordinate system.

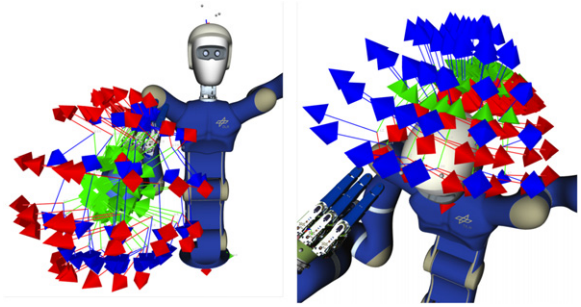


Fig. 10. Generated spheres for complete RoM (left) of the hand joint (right) of the head joint.

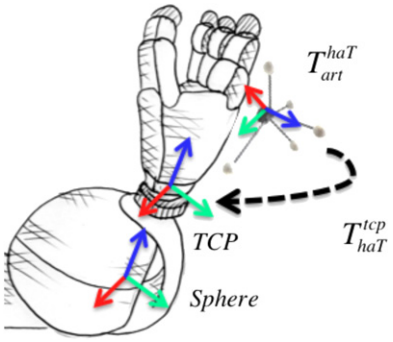


Fig. 11. TCP sphere hand kinematics.

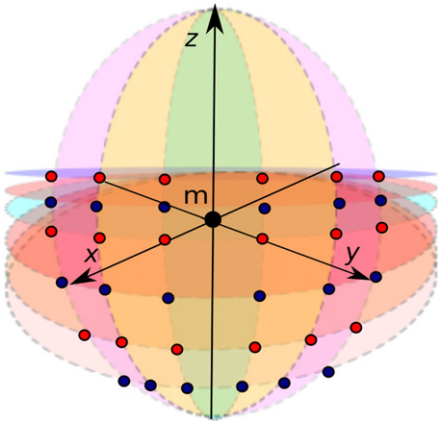


Fig. 12. 36 measurements constructed of six circular trajectories around the *x*-axis and six around the *z*-axis.

are always mounted at every same position and orientation. To identify T^{h}_{heT}, the distance and orientation between the head rigid body origin and the origin of the coordinate system of the head joint is needed. For T^{TCP}_{haT}, the distance and orientation between the mounted hand marker and the TCP end-pose is needed. These distances and orientations cannot be physically measured with a ruler or measuring tape, mainly because for the former the origin of the head joint is inside the pan–tilt unit used to move the head and for the latter the TCP end-pose is located inside the flange of the last joint of the kinematic chain. Therefore, the main task of this pose estimation method lies on the identification of relative pose of the rigid bodies w.r.t. their corresponding joint origin coordinate system. A calibration procedure to estimate them is presented in the following section.

5.2. Calibration of rigid bodies to Justin

The calibration of the rigid bodies to *Justin* consists of estimating the transformations T^{h}_{heT} and T^{TCP}_{haT} (Fig. 9), thus identifying the pose of the head T^{h}_{art} and the pose of the TCP T^{TCP}_{art} w.r.t. the tracking system's base coordinate frame.

The CoR and AoR estimation methods presented in Sections 3 and 4 are used to find these relative transformations. Therefore, the calibration procedure is divided into two steps.

1. Estimation of the position of the origin (CoR).
2. Estimation of the orientations of the obtained origin (AoR).

The first step for the calibration procedure is to create sphere-like measurements around each joint. The joint of *Justin's* head is composed by a pan–tilt unit (i.e., two DoFs). The pan unit has an RoM of -90° to 90° and the tilt unit has RoM of -20° to 40°. The last joint of the arm (i.e., the hand TCP) has three DoFs (x, y, z); the RoM of the x-axis is -45° to 80°, that of the y-axis is -170° to 170°, and that of the z-axis is -45° to 135°. Neither of these two joints can create a complete sphere, but they can create at least one hemisphere of the sphere (Fig. 10).

As mentioned earlier, the head joint has only two DoFs (y-axis and z-axis). The hand has three DoFs (x-axis, y-axis, and z-axis); however, these three DoFs at the TCP end-pose are achieved by the redundant kinematic chain of the complete arm. The TCP end-pose is actually a virtual pose obtained by applying an offset to the last coordinate frame of the kinematic chain, which is the sphere (Fig. 11). When computing the CoR or AoR of the TCP end-pose, what is actually computed is the sphere joint coordinate frame, and then an offset is applied to reach the TCP. When rotating around the x-axis or z-axis of the TCP, the x-axis or z-axis of the sphere rotates, respectively; however, due to the kinematics of the sphere joint the y-axis does not exhibit this behavior. Therefore, only two AoRs are estimated for the head and hand joint, and the last orientation is obtained by computing an orthogonal vector to the two estimated ones. The main goal of this calibration procedure is to identify the rigid transformation between the mounted tracking targets and their joint's coordinate system, T^{TCP}_{haT} and T^{h}_{heT} (Fig. 8). This T is constructed as follows:

$$R = \left[AoR_x, AoR_y, AoR_z\right] \tag{21}$$

$$t = \left[m_x, m_y, m_z\right]^T \tag{22}$$

$$T = \left[R, t\right], \tag{23}$$

where AoR_x, AoR_y, AoR_z are the directions of the x, y, and z-axes estimated with the planar–radial fitting technique, and m is the CoR estimated with the sphere-fitting technique. To avoid taking too many measurements, we create a sphere composed of measurements tracing out circular planes around the x/y-axis and around the z-axis. Therefore, if N planes are created from uniformly separated angles between the RoM of the z-axis and M planes from

uniformly separated angles between the RoM of the x/y-axis, $N \times M$ measurements are obtained from every plane intersection (Fig. 12). Each plane is used to estimate an AoR of its corresponding axis. The final AoR is the average of all the planes; this is applied to the two estimated axes. For the hand, it is computed as follows:

$$AoR_z = rotAvg(AoR_z^0, AoR_z^1, \ldots, AoR_z^N) \quad (24)$$

$$AoR_x = rotAvg(AoR_x^0, AoR_x^1, \ldots, AoR_x^M). \quad (25)$$

To compute the average of rotations, we solve for a minimization problem based on the Frobenius norm with a Euclidean metric on the rotation group $SO(3)$, as described by Sharf et al. [32] and Gramkow [33].

The third orientation is computed as the cross-product of Eqs. (24) and (25) as follows:

$$AoR_y = AoR_z \times AoR_x. \quad (26)$$

The full procedure to compute the transformation of a tracking target to the origin of its joint is listed in Algorithm 1.

Algorithm 1 Frame Estimation

Input: N (planes around Z), M (planes around X)
Output: T (rigid transformation of measurements to origin)

$D \leftarrow constructDforCoR(N \times M)$
$m \leftarrow computeCoR(D)$
$D_z \leftarrow extractZplanes(D)$
$A_z \leftarrow constructAforAoR(D_z, N)$
$AoR_z \leftarrow computeAoR(A_z)$
$D_x \leftarrow extractXplanes(D)$
$A_x \leftarrow constructAforAoR(D_x, M)$
$AoR_x \leftarrow computeAoR(A_x)$
$AoR_y = AoR_x \times AoR_z$
$T = (AoR_x, AoR_y, AoR_z, m)$

In practice, it cannot be assumed that AoR_x and AoR_z are perfectly orthogonal to each other, because they are estimated independently. We deal with this by normalizing the cross-product of AoR_x and AoR_z. Therefore, line $AoR_y = AoR_z \times AoR_z$ of Algorithm 1 is actually computed as follows:

$$Cross_{xz} = AoR_x \times AoR_z \quad (27)$$

$$AoR_y = Cross_{xz} / \|Cross_{xz}\|. \quad (28)$$

5.3. Error identification

Now that the missing transformations (T_{heT}^h and T_{haT}^{TCP}) of the implicit loop closure (Fig. 8) are obtained from the calibration procedure, the pose of *Justin's* TCP can be estimated. To recall the implicit loop closure, TCP_{fk} and TCP_{art} are both the pose of the TCP w.r.t. the head joint (T_h^{TCP}); however each of them is obtained with a different loop closure:

$$TCP_{art} = T_h^{TCP} = (T_{art}^h)^{-1}(T_{art}^{TCP}) = (T_{art}^{heT} T_{heT}^h)^{-1}(T_{art}^{haT} T_{haT}^{TCP}) \quad (29)$$

$$TCP_{fk} = T_h^{TCP} = T_h^a T_a^{TCP}. \quad (30)$$

TCP_{fk} is the pose of the TCP computed using the measured joint positions with a forward kinematic model (i.e., this is the pose that *Justin* is reaching according to its position sensors on each joint). TCP_{art} is the pose of the TCP estimated by the tracking system. Ideally, TCP_{art} and TCP_{fk} should yield the same measured pose; however, as can be seen in Fig. 13, this is not the case.

The error between TCP_{art} and TCP_{fk} is represented as an error tuple $e = \langle e_t, e_\theta \rangle$ extracted from the $\Delta T = (TCP_{art})^{-1} TCP_{fk}$, as shown in Fig. 14.

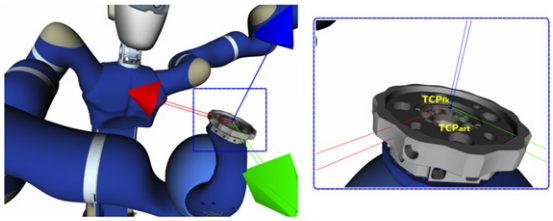


Fig. 13. Measured TCP_{fk} versus estimated TCP_{art}.

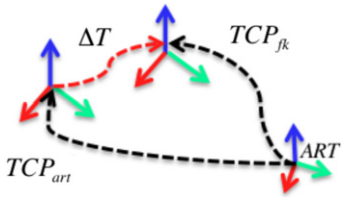


Fig. 14. Error between TCP_{art} and TCP_{fk}: $\Delta T = (TCP_{art})^{-1} TCP_{fk}$.

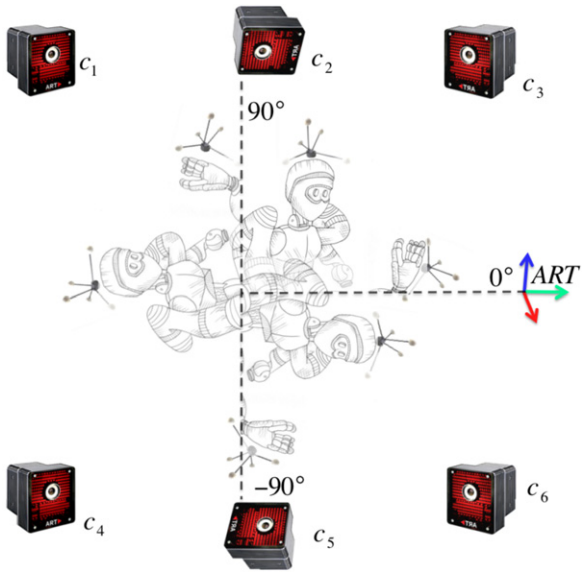


Fig. 15. *Justin* within ART camera setup.

The error tuple e is composed of $e_t = (\Delta T(t))$, which is the translational component of ΔT that represents the translational error between the coordinate frames. $e_\theta = angleaxis(\Delta T(R))$ is the angle around the rotation axis of the rotational component of ΔT in angle–axis representation. 30 random poses are generated and the error e between TCP_{art} and TCP_{fk} is estimated for each. *Justin's* kinematic chain has the ability to reproduce the same joint positions with extremely high accuracy (0.001°). Therefore, to evaluate the reliability and performance of the tracking system, the same hand end-poses with different *Justin* torso positions are reproduced. *Justin's* torso is set to 0°, 90°, and −90° w.r.t. the ART base coordinate system, as shown in Fig. 15.

Five tests were performed, and their results are listed in Table 1. The translation component of the error tuple e_t is evaluated by comparing its length $\|e_t\|$ throughout all the different hand poses and torso positions. In the real set-up, only four cameras were functional (c_1, c_2, c_4, c_6); therefore moving *Justin* around will cause occlusion of the hand rigid body to some of the cameras. The standard deviations of errors for all of the torso positions are around 1 mm for $\|e_t\|$ and between 0.2° and 0.3° for $\|e_\theta\|$; this deviation reflects the noise in the measurements system as well as the deviations in the estimation procedure. The mean errors

Table 1
Estimated errors with different *Justin* torso positions.

Error/torso angle	0°	90°	90°	−90°	90°
Max $\|\|e_t\|\|$ (cm)	0.86	0.79	0.79	1.04	0.82
Mean $\|\|e_t\|\|$ (cm)	0.62	0.56	0.56	0.76	0.48
Std $\|\|e_t\|\|$ (cm)	0.1	0.11	0.11	0.12	0.15
Max $\|\|e_\theta\|\|$ (°)	1.34	1.86	1.86	1.59	1.23
Mean $\|\|e_\theta\|\|$ (°)	0.93	1.20	1.66	1.14	0.89
Std $\|\|e_\theta\|\|$ (°)	0.27	0.31	0.32	0.27	0.21

Fig. 16. *Ripley* v1.0.

do show some variations between torso positions; this is caused partly from the position inaccuracy of the robot joints and the flexibility in the gears and light-weight structures of the kinematic chain. However, the translation errors of the fourth position (torso in −90°) are considerably higher (2 mm) than all of the other torso positions. This variation can be caused by camera occlusions due to the position of the head of *Justin*, by occluding the main cameras (c_1 and c_2) the errors in the measurements from the tracking system are higher (nonetheless, still within the range of the overall estimated errors). This is a limitation for this estimation method; however, since the goal is to validate our previous work [17], the environment is controlled to take account of these limitations. From the complete evaluation, it is concluded that this method is robust and consistent, and thus it suitable to be used as a *ground truth* for any other method.

6. Case study 2: identification of Ripley the Robot v2.0's kinematics

Ripley the Robot is the name of a situated conversational assistant created by Mavridis [18,19]. This conversational assistant embodies a modular architecture in which a grounded situated model (GSM) resides in a centrally located module, surrounded with language, perception, and action-related modules. The full framework for the GSM enables the robot to integrate both language and sensory information about the situation and environment. The robot's environment consists of a table populated with objects and a human interacting with both the robot and the objects (Fig. 16).

The situation model contains all of the information the robot acquires about itself and the environment, via its sensory modules (vision, language, proprioception). It contains representations of entities potentially constructed as agents (itself, the human, the objects) and the spatial relations between them. Each potential agent has a physical (geometric) description of its body associated with it, as well as a mental (beliefs, etc.) description. Alongside these standardized representations, in the GSM reside a number of standard processes for updating the representations given incoming information from the senses (vision, proprioception) as well as language (verbal descriptions), for reducing certainty in

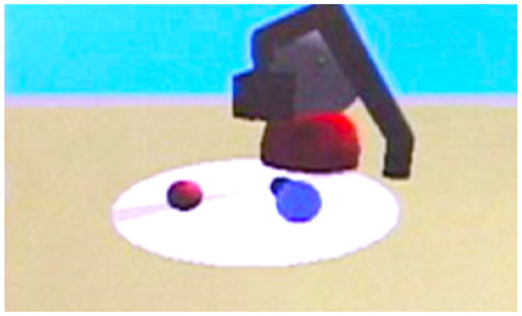


Fig. 17. *Ripley* v1.0 mental model partial visualization of the most likely state of affairs of a situation.

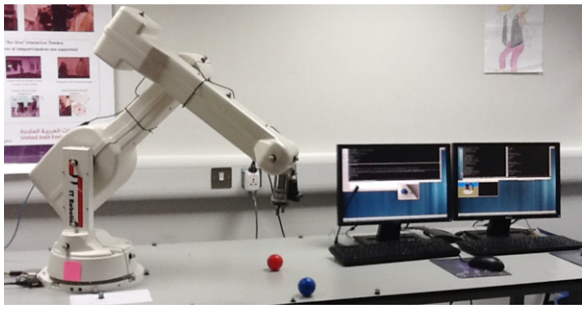


Fig. 18. *Ripley* v2.0 set-up.

the case that no new incoming information exists, and for keeping moments and events in an elementary episodic memory [34]. This situation model and its visualization (Fig. 17) make up the core module of the GSM architecture. It consists of agents (itself, the human, the objects) and the spatial relations between them.

Every time new sensory data is obtained, the situation model is updated, and the resulting internal visualizations change. The sensory data includes not only vision-derived data, but also the feedback from the proprioceptor module, which uses the position of the encoders of the motors of each joint and feeds them to a kinematic model in order to represent the position and configuration of the robot in the visualization of the current situation. If the internal kinematic model that the robot constructs of its body is incorrect, the GSM will not function correctly. For example, it will assume that it is viewing the world from a position that differs from reality; and thus the positioning of its visual percepts within its mental model will be wrong (i.e., objects might not only appear at incorrect positions, but might also appear to be moving when the robot is moving, while in reality they might be stationary). In this case study, we adapt the GSM framework to a completely different embodiment as compared to the original embodiment of Ripley v1.0: a new robotic arm, with a different kinematic model and fewer DoFs (Ripley v1.0 with seven DoFs versus Ripley v2.0 with five DoFs) (Fig. 18).

In order for the GSM framework to work with a different robotic arm (ST Robot R17), certain relative positions need to be identified, mainly the relative position of the origin of the first joint of the robot w.r.t. the table and the position of the camera mounted on the robot end-effector. As shown in Fig. 19, the origin of the first joint is somewhere inside the robots embodiment, as well as the end-pose of the last link, so these positions cannot be measured. The identification of these parameters is crucial for the adaption of the new robotic arm to the GSM framework. Therefore, the VICON Motion Capture system[10] is used to identify the origins of the first joint of the robot (i.e., the base) and the last joint of the robot (i.e., end-effector).

[10] VICON Motion Capture System. http://www.vicon.com/index.html.

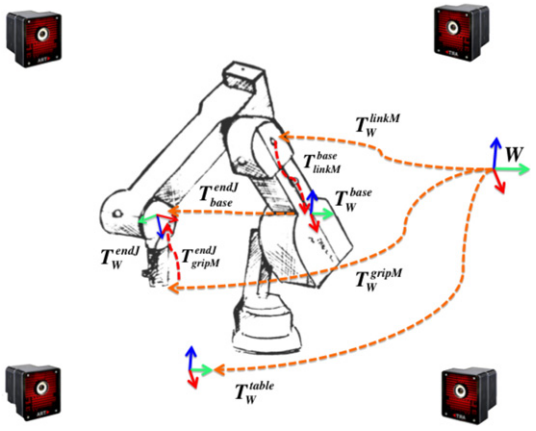


Fig. 19. *Ripley* v2.0 setup and loop closures in a VICON MoCap system.

6.1. VICON set-up for unknown pose estimation

Ripley v2.0's lab has a mounted VICON MX Motion Capture (MoCap) system with eight cameras and speed of up to 2000 fps. Ripley's table is positioned at the center of the tracking volume and mounted markers throughout the manipulator structure and the table (Fig. 19).

T_W^{base} is the pose of the base joint in the tracking system's coordinate frame, which is estimated as follows:

$$T_W^{base} = T_W^{linkM} T_{linkM}^{base}. \tag{31}$$

T_W^{endJ} is the pose of the end-effector in the tracking system's coordinate frame, which is estimated as follows:

$$T_W^{endJ} = T_W^{gripM} T_{gripM}^{base}. \tag{32}$$

T_W^{gripM} and T_W^{linkM} are the poses of the markers mounted in the robot obtained from the MoCap system. T_{linkM}^{base} and T_{gripM}^{endJ} are the unknown relative poses that are estimated using the proposed joint origin identification method. Knowing these two poses, we can compute the relative pose of the robot w.r.t. the table:

$$T_{base}^{table} = (T_W^{table})^{-1} T_W^{base}, \tag{33}$$

where T_W^{table} is the pose of the center of the table obtained from the MoCap system and the relative pose of the camera w.r.t. end-effector is T_{gripM}^{endJ}.

6.2. Unknown pose estimation

Sphere-like measurements are created around the first multi-DoF joint of the manipulator, by rotating the first link of the robot throughout its RoM ($-140°$ to $160°$) in $10°$ increments; the base of the robot was rotated from $-70°$ to $180°$. The end-effector has a more limited RoM: it rotates around just one axis from $-100°$ to $100°$. In Fig. 20, a visualization of the obtained measurements from the joint trajectories is shown. The red spheres are the markers from the table, robot base, and end-effector; the green spheres are the joints of the robot; and the black spheres are the marker measurements mounted on the moving bodies.

The same approach from Case Study 1 is used to identify the unknown relative poses T_{linkM}^{base} and T_{gripM}^{endJ}. However, in this case only the three DoFs of the origins (i.e., the CoR) are computed, because the rotational components of the robot and the table have the same orientation. Moreover, the rigid body of the table is calibrated in the tracking system to have the same orientations as the world coordinate frame, so estimating the AoRs was not necessary. Once the missing relative poses are identified, the kinematic model from the GSM framework is updated accordingly, and an accurate situation model is obtained (Fig. 21).

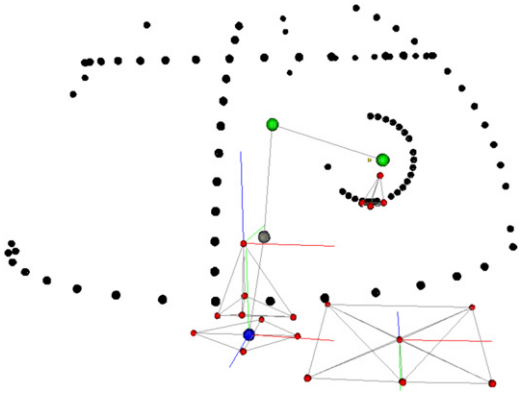


Fig. 20. *Ripley* v2.0 calibration. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Fig. 21. *Ripley* v2.0 mental model partial visualization.

Table 2
Estimated translational error for Ripley v2.0 end-poses.

Error	Max	Mean	Std
$\|e_t\|$ (mm)	4.61	4.56	0.04

6.3. Estimated pose evaluation

To evaluate the quality of the obtained estimations, the relative pose of T_W^{endJ} w.r.t. T_W^{base} is estimated as follows:

$$T_{base}^{endJ} = (T_W^{endJ})^{-1} T_W^{base}. \tag{34}$$

When recording the measurements of the markers from the tracking system, the positions of the encoders of each motor of the manipulator are recorded as well. Therefore, the error between the absolute translational component of T_{base}^{endJ} (t_{base}^{endJ}) and the absolute Cartesian pose of the end-effector w.r.t. the base computed by the robot's on-board controller (t_{robot}) is used to evaluate the system. The difference between the norms of these two vectors is the translational error $\|e_t\|$. This translational error is computed for ten configurations of the kinematic chain generated from the sphere-like measurements (Table 2). The mean $\|e_t\|$ is 4 mm. These errors can caused by several sources: (i) calibration of the encoders of the manipulator, (ii) positioning of the markers, and/or (iii) errors in the estimation procedure. This is not a direct evaluation of the quality of the estimation, because it is not certain that the positioning of the robot is completely accurate. However, with this evaluation it can be concluded that the estimation of the relative poses has a translational error below 4 mm and standard deviation of 0.04 mm.

7. Conclusion and discussion

This paper presented an approach for identifying joint origins of articulated robots using CoR and AoR methods adapted from the biomechanics community. This approach was demonstrated to be used with multi-camera optical tracking systems; however, it is not limited to them: any other type of tracking system that generates 3-DoF poses of tracking markers can be used.

An alternative to identifying the poses of the joint origins of articulated robots is to physically measure or approximate them (just like is done with mobile platforms). However, these measurements or approximations are unreliable and uncertain: if a real ground truth needs to be extracted or if the possible errors in the kinematic chains are within millimetric range, a joint identification process like the one proposed has to be applied. If a certain method needs to be evaluated, the evaluation method should be implemented in a systematic and mathematical approach. This is what motivated us to investigate this topic and present two case studies to show the importance of this identification procedure.

In the first case study, a pose estimation method using the IR optical tracking system mounted in *Justin's* lab was implemented. A thorough description of the rigid body calibration procedure is presented. The calibration procedure is evaluated on synthetic and real data. With real data, a standard deviation of 0.1 mm on the translational component of the hand joint and 0.3 mm on the translational component of the head joint is achieved. From the evaluation results, we concluded that this method is robust and efficient under certain limitations. These limitations are regarding possible occlusions by the position of *Justin* within the lab. However, this is not a problem, since this method was used as a *ground truth* for a marker-less pose estimation method.

The second case study was less complex: it was only necessary to find the CoR of two joints of a 5-DoF robotic arm mounted on a table, in order to adapt the GSM cognitive architecture to a new embodiment. However, the identification of these origins was crucial for the adaptation of the GSM framework to the new robotic arm or else not only the situation model would have been inaccurate, but also the objects would appear to be moving while they were not, creating multiple problems, such as triggering and registering false events in the episodic memory of the robot. Fortunately, our method was readily applied to this case too, achieving a standard deviation of 0.04 mm on translation estimation and alleviating any such problems.

Thus, by providing motivation, comparison to existing work, derivation and first evaluations, and application to two cases of our method, we were able to concretely illustrate its wide applicability and effective performance, towards creating accurate kinematic models for many different robotic embodiments.

Appendix A. Derivation of CoR estimation by Chang and Pollard [15]

The CoR estimation by Chang and Pollard [15] is based on a method for non-rigid sphere fitting of measurements with a constant radius r, developed by Pratt [25]. Considering a point $v = (x, y, z)^T$ on the surface of a sphere with center $m = (x_c, y_c, z_c)$ and radius r, the following equation for a sphere can be formulated:

$$(x - x_c)^2 + (y - y_c)^2 + (z - z_c)^2 - r^2 = 0. \tag{A.1}$$

A basis function of coefficients $u = (a, b, c, d, e)^T$ is extracted from Eq. (A.1) by rewriting it as

$$aw + bx + cy + dz + e(1) = 0. \tag{A.2}$$

$w = x^2 + y^2 + z^2$ and $(w, x, y, z, 1)$ are basis functions whose coefficients are defined in u. Therefore, to find the u that fits to a data point v, the following algebraic distance δ is calculated:

$$\delta(u) = (w, x, y, z, 1)^T u. \tag{A.3}$$

The values of the polynomial coefficients in u determine the CoR and radius of the sphere. These values are relative rather than absolute; this enables a to be a free parameter used to set a constraint to the cost function. The choice of this constraint is the most important step for achieving a good fit [25]. Chang and Pollard [15] propose using the following normalization constraint developed by Pratt [25]:

$$b^2 + c^2 + d^2 - 4ae = 1. \tag{A.4}$$

Using Eqs. (A.1) and (A.2), (A.4) can be rewritten as

$$a^2 r^2 = 1. \tag{A.5}$$

A least-squares problem is formulated using Eq. (A.3), which is the algebraic distance δ_k. The algebraic distances δ_k of all n measurements are written in a data matrix D as follows:

$$\begin{bmatrix} \delta_1 \\ \vdots \\ \delta_k \\ \vdots \\ \delta_n \end{bmatrix} = \begin{bmatrix} w_1 & x_1 & y_1 & z_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ w_k & x_k & y_k & z_k & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ w_n & x_n & y_n & z_n & 1 \end{bmatrix} u = Du. \tag{A.6}$$

To find the spherical fit u, the sum of squared algebraic distances δ results in the following cost function:

$$f = (Du)^T (Du) = u^T D^T D u. \tag{A.7}$$

The matrix form of the normalization constraint Eq. (A.4) can be expressed as follows:

$$\begin{bmatrix} a & b & c & d & e \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 0 & -2 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ -2 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \\ e \end{bmatrix} = 1 = u^T C u \tag{A.8}$$

C is the constraint matrix, and renaming $D^T D$ from Eq. (A.7) to S leads to the following constrained optimization problem proposed by Chang and Pollard [15]:

$$\begin{Bmatrix} \text{minimize :} & u^T S u \\ \text{subject to :} & u^T C u = 1 \end{Bmatrix}. \tag{A.9}$$

As stated earlier, the vector u is composed of the basis function coefficients that define a sphere $u = [a, b, c, d, e]^T$. By comparing Eqs. (A.1) and (A.2), m and r are calculated as follows:

$$m = (x_c, y_c, z_c)^T = -\frac{1}{2a}(b, c, d)^T \tag{A.10}$$

$$r = \|m\|^2 - \frac{e}{a}. \tag{A.11}$$

Appendix B. Derivation of AoR estimation by Chang and Pollard [16]

The combined function proposed by Chang and Pollard [16] is a combination of a planar and radial cost function:

$$f = \sum_{k=1}^{N} (\delta_k^2 + \epsilon_k^2) = f_p + f_r. \tag{B.1}$$

B.1. Planar cost function

The planar cost function is formulated as follows:

$$f_p = \sum_{k=1}^{N} (\delta_k)^2 = \sum_{k=1}^{N} (v_k \cdot n - \mu)^2. \tag{B.2}$$

μ can be found by setting the derivative of Eq. (B.2) to zero. This yields the following equation:

$$\mu = \left(\frac{1}{N} \sum_{k=1}^{N} v_k \right) \cdot n = \overline{v} \cdot n. \tag{B.3}$$

$\overline{v}$ is the centroid of the trajectory of all measurements. Eq. (B.2) can be rewritten as

$$f_p = \sum_{k=1}^{N} (v_k \cdot n - \overline{v} \cdot n)^2. \tag{B.4}$$

If the measurements are expressed relative to the centroid of their trajectories, as follows:

$$u_k = v_k - \overline{v}. \tag{B.5}$$

Then Eq. (B.4) can be rewritten as

$$f_p = \sum_{k=1}^{N} (\delta_k)^2 = \sum_{k=1}^{N} (u_k \cdot n)^2. \tag{B.6}$$

This cost function can be rewritten to matrix form using the standard formulation of fitting of algebraic surfaces proposed by Pratt [25]. Construct a matrix A composed of each measurement relative to the trajectory centroid u_k, as follows:

$$A = \left[u_1, \ldots, u_k, \ldots, u_n \right]^T. \tag{B.7}$$

Then Eq. (B.6) is converted to

$$fp = \|An\|^2 = n^T A^T A n = n^T S n. \tag{B.8}$$

B.2. Radial cost function

The radial error measures the deviation between (i) the distance of the measurement v_k to the CoR m and (ii) the radius r of the best-fitting cylinder whose axis of symmetry is the direction of the CoR n. Given the axis direction n and any point on the line of the CoR m, the shortest vector from the target measurement to the line is estimated as the vector w_k:

$$.w_k = (I - nn^T)(v_k - m) \tag{B.9}$$

I is a 3×3 identity matrix. $(I - nn^T)$ is a matrix that operates on a vector by subtracting the parallel component of the vector to direction n. The resulting vector w_k is the orthogonal component of the vector to direction n. The radial cost function is estimated as the total squared radial error, as follows:

$$f_r = \sum_{k=1}^{N} (\epsilon_k)^2 = \sum_{k=1}^{N} (\|w_k\| - r)^2. \tag{B.10}$$

r is calculated by setting the derivative of Eq. (B.10) to zero. This yields the following equation:

$$r = \frac{1}{N} \sum_{k=1}^{N} \|w_k\|. \tag{B.11}$$

Summing Eqs. (B.6) and (B.10), the combined planar–radial cost function is constructed, as follows:

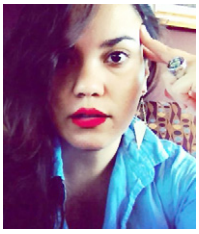
$$f = f_p + f_r = \sum_{k=1}^{N} (\delta_k)^2 + \sum_{k=1}^{N} (\epsilon_k)^2 \tag{B.12}$$

$$f = n^T S n + \sum_{k=1}^{N} (\|(I - nn^T)(v_k - m)\| - r)^2. \tag{B.13}$$

References

[1] D. Mellinger, N. Michael, V. Kumar, Trajectory generation and control for precise aggressive maneuvers with quadrotors, International Journal of Robotics Research 31 (2012) 664–674.
[2] M. Stilman, K. Nishiwaki, S. Kagami, J.J. Kuffner, Planning and executing navigation among movable obstacles, in: Proc. 2006 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pp. 820–826.
[3] J. Fink, N. Michael, S. Kim, V. Kumar, Planning and control for cooperative manipulation and transportation with aerial robots, International Journal of Robotics Research 30 (2011) 324–334.
[4] A. Gashler, Real-time marker-based motion tracking: application to kinematic model estimation of a humanoid robot, M.Sc. Thesis, TU Munich Department of Informatics, 2011.
[5] J. Hofschulte, Zweibeiniger Roboter mit parallelkinematischen Hüftgelenken, Dissertation, Universität Hannover, Books On Demand GmbH, Norderstedt, 2006.
[6] R. Siegwart, I.R. Nourbakhsh (Eds.), Introduction to Autonomous Mobile Robots, 1st edition, MIT Press, 2004.
[7] G.M. Hoffmann, S.L. Wasl, C.J. Tomlin, Quadrotor helicopter trajectory tracking control, in: Proc. AIAA Guidance, Navigation, and Control Conf.
[8] Y. Bouuktir, M. Haddad, T. Chettibi, Trajectory planning for a quadrotor helicopter, in: 16th Mediterranean Conference on Control and Automation.
[9] B. Siciliano, O. Khatib (Eds.), Springer Handbook of Robotics, 1st edition, Springer, 2008.
[10] C. Ott, O. Eiberger, W. Friedl, B. Bauml, U. Hillenbrand, C. Borst, A. Albu-Schaffer, B. Brunner, H. Hirschmuller, S. Kielhofer, R. Konietschke, M. Suppa, T. Wimbock, F. Zacharias, G. Hirzinger, A humanoid two-arm system for dexterous manipulation, in: 2006 6th IEEE-RAS International Conference on Humanoid Robots, pp. 276–283.
[11] A. Albu-Schäffer, S. Haddadin, C. Ott, A. Stemmer, T. Wimböck, G. Hirzinger, The DLR lightweight robot: design and control concepts for robots in human environments, Industrial Robot: An International Journal 34 (2007) 376–385.
[12] S.S. Gamage, J. Lasenby, New least squares solutions for estimating the average centre of rotation and the axis of rotation, Journal of Biomechanics 35 (2002) 87–93.
[13] P. Cerveri, N. Lopomo, A. Pedotti, G. Ferrigno, Derivation of centers and axes of rotation for wrist and fingers in a hand kinematic model: methods and reliability results, Annals of Biomedical Engineering 33 (2005) 402–412.
[14] K. Halvorsen, M. Lesser, A. Lundberg, A new method for estimating the axis of rotation and the center of rotation, Journal of Biomechanics 32 (1999) 1221–1227.
[15] L.Y. Chang, N.S. Pollard, Constrained least-squares optimization for robust estimation of center of rotation, Journal of Biomechanics 40 (2007) 1392–1400.
[16] L.Y. Chang, N.S. Pollard, Robust estimation of dominant axis of rotation, Journal of Biomechanics 40 (2007) 2707–2715.
[17] N. Figueroa, H. Ali, F. Schmidt, 3D registration for verification of humanoid Justin's upper body kinematics, in: Canadian Conference on Computer and Robot Vision 2012, CRV 2012.
[18] N. Mavridis, D. Roy, Grounded situation models for robots: where words and percepts meet, in: IROS, pp. 4690–4697.
[19] N. Mavridis, Grounded situation models for situated conversational assistants, Ph.D. Thesis, Massachusetts Institute of Technology, Dept. of Architecture, Program In Media Arts and Sciences, 2007.
[20] A. Fuhrmann, R. Splechtna, J. Prikryl, Calibration requirements and procedures for a monitor-based augmented reality system, in: IEEE Transactions on Visualization and Computer Graphics, Vol. 1, pp. 255–273.
[21] K. Ahlers, C. Crampton, D. Greer, E. Rose, M. Tuceryan, Augmented vision: a technical introduction to the grasp 1.2 system, 1994.
[22] A. Fuhrmann, R. Splechtna, J. Prikryl, Comprehensive calibration and registration procedures for augmented reality, in: Proceedings Eurographics Workshop on Virtual Environments, pp. 219–228.
[23] D. Schmalstieg, A. Fuhrmann, G. Hesina, Z. Szalavari, L.M. Encarnacao, M. Gervautza, W. Purgathofer, The studierstube augmented reality project, 2000.
[24] T. Sielhorst, M. Feuerstein, J. Traub, O. Kutter, N. Navab, Campar: a software framework guaranteeing quality for medical augmented reality, International Journal for Computer Assisted Radiology and Surgery 1 (2006) 29–30.
[25] V. Pratt, Direct least-squares fitting of algebraic surfaces, in: Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH'87, ACM, New York, NY, USA, 1987, pp. 145–152.
[26] D.P. Berstekas, Nonlinear Programming, second ed., Athena Scientific, Cambridge, MA, 1999.
[27] I. Vapnyarskii, Lagrange Multipliers, Springer, 2001.
[28] F. Bookstein, Fitting conic sections to scattered data, Computer Graphics and Image Processing 9 (1979) 56–71.
[29] A. Fitzgibbon, M. Pilu, R. Fisher, Direct least square fitting of ellipses, IEEE Transactions on Pattern Analysis and Machine Intelligence 21 (1999) 476–480.
[30] M. Powell, A direct search optimization method that models the objective and constraint functions by linear interpolation, in: S. Gomez, J.-P. Hennart (Eds.), Advances in Optimization and Numerical Analysis, Kluwer Academic, Dordrecht, Oaxaca, Mexico, 1994, pp. 51–67.
[31] J.A. Nelder, R. Mead, A simplex method for function minimization, The Computer Journal 7 (1965) 308–313.
[32] I. Sharf, A. Wolf, M. Rubin, Arithmetic and geometric solutions for average rigid-body rotation, Mechanism and Machine Theory 45 (2010) 1239–1251.
[33] C. Gramkow, On averaging rotations, Journal of Mathematical Imaging and Vision 15 (2001) 7–16.

[34] N. Mavridis, M. Petychakis, Human-like memory systems for interactive robots: desiderata and two case studies utilizing grounded situated models and online social networking, in: AISB.



Nadia B. Figueroa received her M.Sc. in Automation and Robotics from the Technical University of Dortmund in 2012, after receiving her B.Sc. in Mechatronics from Monterrey Tech. During her studies in Germany, she worked on a number of research topics such as automated vehicles, object recognition, and kinematic calibration, as well as machine vision. Her thesis, entitled "3D registration for Verification of Humanoid Justin's Upper Body Kinematic", developed at the Institute of Robotics and Mechatronics (RM) of the German Aerospace Center (DLR), addressed the problem of self-verification of the complex kinematic chains of the humanoid Justin by using its on-board stereo vision system. She is currently a research assistant at New York University Abu Dhabi (NYUAD). Her research interests include robotics, computer vision, machine learning, and cognitive systems.

Florian Schmidt received his M.Sc. in 2007 from the University of Applied Sciences Munich, and since then has been affiliated with the German Aerospace Center (DLR), working there on the further development of the humanoid robot *Rollin' Justin*.

Haider Ali is currently a Senior Researcher at the Institute of Robotics and Mechatronics (RM) of the German Aerospace Center (DLR). His research is focused on developing efficient 3D object detection and pose estimation methods for real-time robotics applications. He received his Bachelor of Science in Computer Science from Bahauddin Zakariya University, one of Pakistan's major universities, in 1998. After that he worked for several multinational IT companies in Pakistan as a Software Engineer and Project Consultant until 2004. Thereafter he pursued a master's degree in software technology from Leuphana University of Lueneburg, and graduated in 2006. He received his Ph.D. from the Technical University of Vienna in 2010.

Nikolaos Mavridis received his Ph.D. from MIT in 2007, after receiving his M.S.E.E. from UCLA and his M.Eng. in ECE from the Aristotle University of Thessaloniki. From 2007 to 2011, he served as Assistant Professor at the College of IT of the United Arab Emirates University, where he founded the Interactive Robots and Media Laboratory (IRML). He is currently serving as an Assistant Professor at New York University Abu Dhabi, where his lab is now located. The lab is home to the "FaceBots" social robots project, as well as to "IbnSina", the first Arabic-language conversational android robot. In his Ph.D. Thesis at MIT, he introduced the "Grounded Situation Model" proposal, and demonstrated its benefits by implementing it on Ripley, a manipulator robot with vision, touch, and speech synthesis/recognition. The sensory motor/linguistic abilities of the resulting system were comparable to those implied by a standard psychological test for 3-year-old children (The "Token Test"). The research interests of Dr. Mavridis include robotics, especially interactive and social robotics, as well as cognitive systems.