

STAC: Simultaneous Tracking and Calibration

Tingfan Wu*, Yuval Tassa^o, Vikash Kumar^o, Javier Movellan*, Emanuel Todorov^o
Dept. Computer Science and Engineering, UC San Diego *
Dept. Computer Science and Engineering, University of Washington^o
{tingfan,yuval.tassa,jrmovellan,etodorov}@gmail.com

Abstract—System identification is an essential first step in robotic control. Here we focus on the calibration of kinematic sensors, such as joint angle potentiometers, tendon/actuator extension sensors and motion capture markers, on complex humanoid robots.

Manual calibration with protractors and rulers does not scale to complex humanoids like the ones studied here. Classic automatic approaches cross-calibrate multiple sensor systems on the same robot by exploiting their redundancy. However, these approaches make the strong assumption that the observed joint angles are functions of the sensor measurements plus observation noise. This assumption is too restrictive on modern humanoids where linear actuators and tendons span multiple joints.

Here we formulate the calibration problem as a Bayesian inference process on a generative model where hidden joint-angles generate sensor observations. A novel alternating optimization approach is developed to simultaneously track space-time joint angles and calibrate parameters (STAC). Explicit estimation of joint angles makes it possible to calibrate sensors that otherwise cannot be handled by classical approaches, such as tendons wrapping on complicated surfaces and spanning multiple joints. We evaluate STAC to calibrate joint potentiometer, tendon length sensor and motion capture marker positions, on a 38-DoF humanoid robot with 24 optical markers, and a 24 DoF tendon driven hand with 12 markers. We show that STAC can be applied to problems that cannot be handled with classical approaches. In addition we show that for simpler problems STAC is more robust than classical approaches and other probabilistic approaches such as the Extended Kalman Filter.

I. INTRODUCTION

System identification (ID) is an essential first step in robotic control, and can be divided into kinematic and dynamic system identification. Dynamic ID deals with quantities which emerge when there is movement, like moments-of-inertia and friction coefficients. Kinematic ID deals with parameters which are relevant even when the velocity is zero, i.e. geometric properties. For example, a pick-and-place robot with an inaccurate kinematics model will do a poor job, regardless of how slow it moves. In this paper, we focus on kinematic ID, including calibrating joint angle and cylinder extension sensors which are typically measured by potentiometers, magnetic or optical encoders. Some of these sensors are directly mounted on the joints; others are connected through a transmission mechanism, such as cranks, cables, tendons or gears. These mechanisms while useful, may change their dynamic range, linearity, and even accuracy of the sensors, all of which essentially contribute additional parameters to be identified.

Manual calibration approaches usually rely on ground-truth joint angle measurements using protractors, and the corresponding joint angle sensor readings. Regression models are then used to learn a function that maps sensor reading into joint angle estimates. We found this approach to be inaccurate and inefficient when applied to complex robot platforms. Accurate measurements are hard to obtain with a protractor, our robot’s arms are covered with tubes and their surface is uneven; making it difficult to align a protractor to a joint. Empirically, measurement precision can be as bad as 5 degrees. The approach is also very time-consuming especially on a large number of joints. One of the humanoid platforms studied in this paper has 38 joints and many of them come with non-linear transmission mechanisms which require multiple measurements over different angles to fully identify the underlying parameters. Furthermore, calibration values can change after each repair or intensive use. The development of a fully automatic calibration system is necessary to keep the robot fully functional.

More sophisticated kinematic ID approaches cross-calibrate multiple sensor systems on the same robot by exploiting their redundancy [3]. For example, consider a robot with potentiometers measuring joint angle and motion capture markers attached to some of the bodies. Assuming both sensors are calibrated, one can infer the pose of the robot using either system, thus the redundancy. However, before the calibration, neither of the systems is accurate. Both come with unknown parameters: the gain and bias for potentiometer and the positions of the markers. To calibrate these parameters, one first collects synchronized measurement from both sensors for multiple frames, and then tries to find the optimal values, such that the two sensor systems agree with each other.

However current kinematic ID approaches assume that the observed joint angles are a function of the sensor measurement plus some observation noise. This assumption raises two issues: First, the assumption would only work for simple one-joint-to-one-sensor sensor types as in Fig. 3abc. Pose sensor measurement on modern humanoids may depend on multiple joint angles. For example, modern dexterous hands are driven by tendons where the length of a tendon is a linear function of multiple joint angles (Fig. 3e). For hydraulic or pneumatic systems, it is common to apply linear actuators to multiple DoF joints such as the 2-DoF rotational gimbal as in Fig. 3d and the Stewart platform [2]. In these cases, it is generally not trivial to write joint angle as a function

of the sensor measurement. More importantly, in some cases the noise-free mapping between sensors and joint angles may not be a function. In this work, we formulate the relationship between sensors and joint angles as a Bayesian generative model in which sensor measurements are noisy observations generated from joint angles. This contrasts with the classical regression-based approach in which the joint angles are treated as noisy observations generated by noiseless sensor readings. This approach lets us calibrate robots in which the relationship between joint angles and sensors is very complex and cases in which the observed angles are not a single-valued function of the sensor readings.

II. MULTI-SENSOR PARAMETER AND JOINT ANGLE ESTIMATION

Consider a tree-structured robot of n -joints with a known (skeletal) kinematics model, including link lengths, joint positions and types. The robot is equipped with different types of pose sensors which measure some quantities as *parameterized* functions of one or more joint angles and derived quantities, such as body positions or orientations. The goal is to identify these (fixed) sensor parameters from multiple synchronized measurements from the multiple sensors.

A. Classical Methods

The past decades have seen the development of kinematic calibration methods that do not require ground truth knowledge of joint angles [3]. Let the forward kinematics function h of a robot arm be as follows

$$\hat{\mathbf{x}} = h(q, \theta) \quad (1)$$

where $\hat{\mathbf{x}}$ is the end-effector pose, $q \in R^n$ are the joint angles and θ are parameters to be identified including potentiometer gains θ_{gain} and biases θ_{bias} . Typically, potentiometer readings are linear functions of the joint angles,

$$q_j = \theta_{gain,j} \cdot p_j + \theta_{bias,j} \quad (2)$$

where p_j is the reading from the corresponding potentiometer. Therefore we can rewrite the (1) as

$$\hat{\mathbf{x}} = h(\theta_{gain,j} p_j + \theta_{bias,j}, \theta) = h(p_j, \theta). \quad (3)$$

Under this framework the system ID problem can be formulated as a non-linear regression problem. Given a sufficient number synchronized measurement of (\mathbf{x}_t, p_t) , the goal is to find a parameter vector θ that minimizes a sum of squared errors cost function

$$L(\theta; \mathbf{x}_{1:T}, p_{1:T}) = \sum_{t=1}^T \|\hat{\mathbf{x}}_t - \mathbf{x}_t\|_2^2 = \sum_{t=1}^T \|h(p_t, \theta) - \mathbf{x}_t\|_2^2. \quad (4)$$

Note that explicit estimation of joint angle q is unnecessary. Such formulation is convenient. However regression approaches rely on the assumption that the observed joint angles q are a function of the sensor readings plus some observation noise. This assumption is often violated in complex, biologically-inspired humanoid robots.

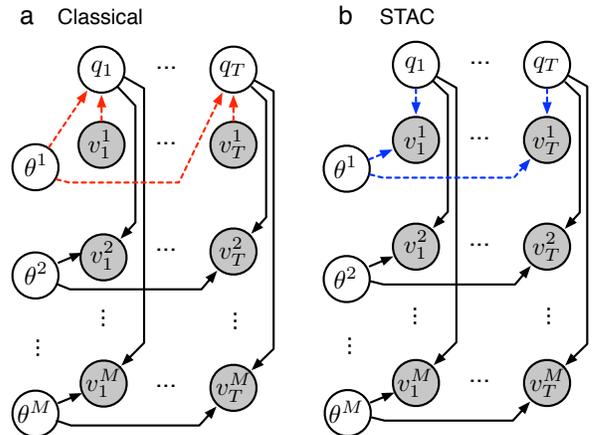


Fig. 1. The graphical model of the (a) classical approach (b) our approach. Joint angles q and sensor parameters θ are hidden and sensor measurements v are observed (shaded).

B. Generative Model

While regression models assume that joint angles are a function of sensor readings, here we use the much weaker assumption that sensor reading are a function of joint angles. Figure 1 illustrates the probabilistic graphical models corresponding to the classical regression based approach and to the approach we propose (STAC). On the top row, each unshaded node q_t represents the *hidden* joint angles in snapshot t while on the left column, each unshaded node θ^i represents *unknown* parameters of sensor i . In the middle, the *observed* sensor measurements are organized into the array of shaded nodes, where the rows are measurements from the same sensor and columns are measurements in the same snapshot. The measurement from sensor i in snapshot t is then denoted as v_t^i . The left panel (a) shows the model for classical regression based approach. Joint angles $q_{1:T}$ are generated from potentiometer reading $v_{1:T}^1$ and parameters θ^1 as shown in dashed arrows. For STAC on right panel (b), we re-assign these arrows that all the sensor measurement (whether it is from potentiometer or not) are generated from joint angles and sensor parameters.

Standard graphical model machinery can be used to compute the negative log likelihood function used by STAC

$$LL(q_{1:T}, \theta) = - \sum_{m=1}^M \sum_{t=1}^T \log p(v_t^m | q_t; \theta^m). \quad (5)$$

where v represents sensor observations, θ represents kinematic parameters and $q_{1:T}$ represents joint angles.

C. Alternating Descent Optimization

The standard maximum likelihood approach to estimate θ is to directly minimize $LL(q, \theta)$. However, direct optimization over all the parameters and joint angles is difficult due to large number of parameters. Consider a 38-DoF humanoid with 2 parameters for each joint and a collection of 100 snapshots would easily amount to 3876 parameters!

We found that the optimization process could be greatly accelerated by using an approach that alternated between

two phases: a p-phase that estimates sensor parameters while keeping joint angle fixed and a q-phase that estimates joint angles using the updated sensor parameters. The optimization in each phase can be further divided into multiple simpler and parallelizable sub-problems taking advantage of the special structure of the graphical model. The idea is justified by the following two key observations:

- joint angles q_t 's of different time frames are conditionally independent from each other when θ^i 's are known;
- on the other hand, the parameter θ^i 's of different sensors are conditionally independent from each other when joint angles are known.

Using this property we can re-write the maximum likelihood problem as

$$\begin{aligned} \min_{\theta, q_{1:T}} \sum_{m=1}^M \sum_{t=1}^T \log p(v_t^m | q_t; \theta^m) \\ = \sum_m \min_{\theta^m} \left(\sum_{t=1}^T \min_{q_t} \log p(v_t^m | q_t; \theta^m) \right). \end{aligned} \quad (6)$$

Notice that the maximization subproblems inside the summations can be optimized independently. In many cases, the subproblems solving each q_i or θ_i are simple enough to have closed-form solutions. Even when numerical optimization is necessary, these subproblems can be optimized in parallel over multiple processors.

D. Sensor Observation Noise Model

Typically, sensor measurement is assumed to be contaminated by additive zero-mean σ_v^2 variance Gaussian noise,

$$P(v_t^m | q_t; \theta) = \mathcal{N}(v | \hat{v}^m(q_t; \theta), \sigma_v^2). \quad (7)$$

Then, the maximum likelihood problem is equivalent to a least squares problem. In other words, the likelihood function can be written as

$$LL(q, \theta) = \frac{1}{2} r(\theta, q)^T r(\theta, q) + \frac{TM}{2} \log 2\pi \quad (8)$$

where the residual vector is

$$r(\theta, q) = \begin{bmatrix} r(\theta, q_1) \\ r(\theta, q_2) \\ \vdots \\ r(\theta, q_T) \end{bmatrix}, \quad r(\theta, q_t) = \frac{1}{\sigma_v} \begin{bmatrix} v_t^1 - \hat{v}_t^1(\theta, q_t) \\ v_t^2 - \hat{v}_t^2(\theta, q_t) \\ \vdots \\ v_t^M - \hat{v}_t^M(\theta, q_t) \end{bmatrix}. \quad (9)$$

Here we put the variance σ_v at sensor level as the same type of sensor typically share similar noise variance. One can certainly use a per-sensor variance option when necessary. There are off-the-shelf tools solving non-linear least squares problem, such as Gauss-Newton or Levenberg-Marquardt approaches.

III. CASE STUDY: CALIBRATING JOINT/TENDON POTENTIOMETER AGAINST MOTION CAPTURE

Motion capture systems, are becoming standard measuring tools in robotic labs for various control and identification

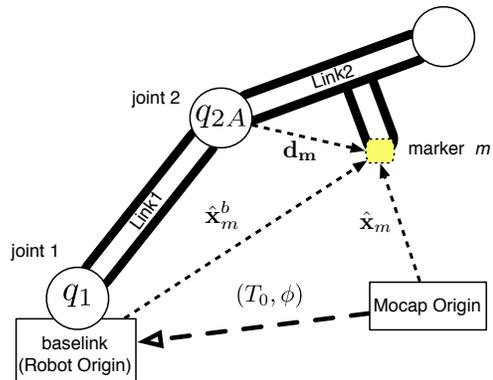


Fig. 2. Markers on kinematics chain

tasks [1], [5], [6]. In this section, we study the case of using motion capture system to calibrate other sensors.

There are two type of pose sensors on the robot: rotary potentiometers for 1-DoF rotary joints and linear potentiometers for 2-DoF Gimbal joints. To identify the potentiometer parameters, we attach M motion capture markers to some links of the robot as auxiliary sensors. In total, the sensor parameters to be identified are potentiometer gains and bias, the translation and rotation of motion capture coordinate frame from the robot baselink frame and the marker positions on the links, or

$$\theta = [\theta_{gain} \quad \theta_{bias} \quad R \quad T_0 \quad d_{1:M}]. \quad (10)$$

Next, we describe how the observations of the various types of sensors are generated and how to solve for the parameters analytically in the “p-phase”. For notational clarity, we further split the observation variable $v : \{\mathbf{x}, p\}$ into \mathbf{x} for motion capture markers and p for potentiometers.

A. Motion Capture Markers

Figure 2 shows the spatial relationship of a marker m and the parent link (link 2) it is attached to. Let d_m be the unknown marker local position in the parent link frame. Then the marker position in the robot (baselink) frame is,

$$\hat{\mathbf{x}}_m^b(d_m, q_t) = h_r(q_t)d_m + h_p(q_t), \quad (11)$$

where $h(\cdot)$ is the forward kinematics function that calculates the position (h_p) and orientation (h_r) of the parent link, to transform d_m into the baselink frame.

The motion capture system measures 3-dimensional position of the markers $x_{m,t} \in \mathbb{R}^3, m = 1, 2, \dots, M$ in the motion capture frame. We denote the transformation from baselink- to motion-capture-frame by rotation matrix $R \in \mathbb{R}^{3 \times 3}$ and translation $T_0 \in \mathbb{R}^3$. Then the prediction of marker position in the motion capture frame is

$$\hat{\mathbf{x}}_m(R, T_0, d_m, q_t) = R\hat{\mathbf{x}}_m^b(d_m, q_t) + T_0. \quad (12)$$

During the “p-phase” of alternating descent, we solve for the parameters R, T_0, d_m while fixing the joint angles $q_{1:T}$. The transformation parameters (R, T_0) , which affect all markers at all times, can be solved for using Procrustes analysis of

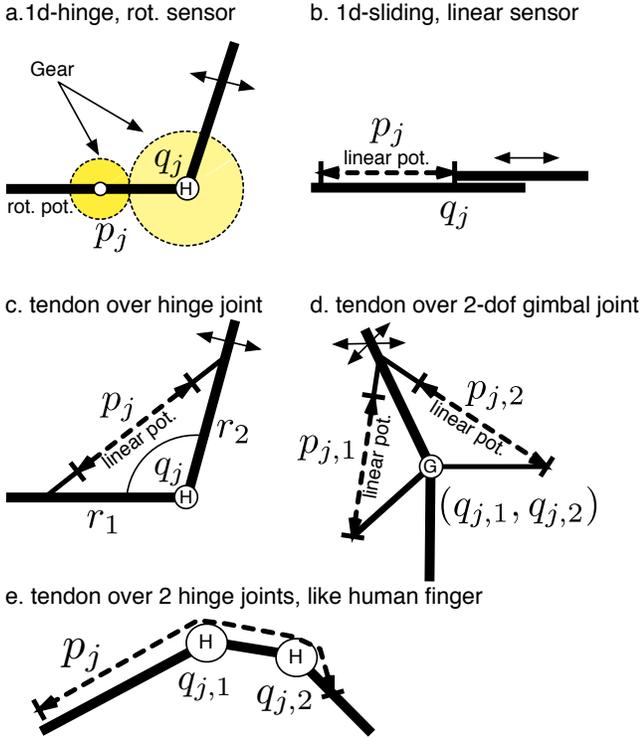


Fig. 3. Exemplar joint types where q_j are generalized joint angle/displacement, p_j are sensor readings.

rigid body motion problem [7]: First we calculate the center of the markers in each coordinate system,

$$\bar{\mathbf{x}}^b = \frac{1}{MT} \sum_{m=1}^M \sum_{t=1}^T \hat{\mathbf{x}}_{m,t}^b, \quad \bar{\mathbf{x}} = \frac{1}{MT} \sum_{m=1}^M \sum_{t=1}^T \mathbf{x}_{m,t}. \quad (13)$$

Next, the rotation matrix can be obtained through singular-value-decomposition of the ‘‘covariance matrix’’,

$$\sum_{m,t} (\hat{\mathbf{x}}_{m,t}^b - \bar{\mathbf{x}}^b)(\mathbf{x}_{m,t} - \bar{\mathbf{x}})^T \stackrel{\text{SVD}}{=} U \Sigma V^T. \quad (14)$$

Then,

$$R = \text{sign}(\det(\Sigma)) V U^T \quad (15)$$

$$T_0 = \bar{\mathbf{x}} - R \bar{\mathbf{x}}^b \quad (16)$$

Finally, the local position of the markers in the corresponding link can be identified by taking the average of observed marker position in the link coordinate (12) :

$$\bar{d}_m = \frac{1}{T} \sum_{t=1}^T h_r(q_t)^{-1} \underbrace{(R^{-1}(\mathbf{x}_{m,t} - T_0) - h_p(q_t))}_{\mathbf{x}_{m,t}^b} \quad (17)$$

B. Potentiometers

There are several popular types of potentiometer mountings as shown in Fig.3. Although potentiometers are typically linear in the rotation angle or linear in the displacement, the transmission mechanism can be either linear or non-linear.

For linear (fixed gearing or direct driving) transmission, such as a hinge joint with one rotary potentiometer (Fig.3a)

or a sliding joint with a linear potentiometer (Fig.3b), the output voltage \hat{p} is linear in the joint angle.

$$\hat{p}(\theta, q) = \theta_{\text{gain}} q + \theta_{\text{bias}} \quad (18)$$

C. Tendons

Tendons are force transmission mechanism connecting two links. The length of a tendon can be used to determine the joint angles between the two links. Classical regression based approaches cannot handle this case because typically there are many joint angle combinations that yield the same tendon length i.e., the mapping from tendon lengths to joint angles is not a single-valued function. Figure 3e shows the tendon used in an anthropomorphic tendon-driven finger. Another type of tendon setup uses linear actuators/sensors on 2-DoF rotation joints. As it is not easy to attach rotary potentiometers on the 2DoF Gimbal structure, two linear potentiometers are attached across the two links on distinct pairs of points (Fig.3d) to measure the joint angles. The anchor points are available from the CAD model. For convenience, the measured voltage and calculated length of these potentiometers are referred to as p_1, p_2 and L_1, L_2 while the corresponding joint angles are q_1 and q_2 . The predicted measurement is then

$$\hat{p}_1(\theta, q) = \theta_{\text{gain}} L_1(q_1, q_2) + \theta_{\text{bias}} \quad (19)$$

$$\hat{p}_2(\theta, q) = \theta_{\text{gain}} L_2(q_1, q_2) + \theta_{\text{bias}}. \quad (20)$$

Note that it is easy to calculate $L(\cdot)$ from q as part of the analytical forward kinematics routine but inferring q from L would require numerical inverse kinematics.

When solving for the potentiometer parameter given the joint angle q_t and the measured voltage p_t at snapshot t , whether the transmission is linear or nonlinear, the equations (18)-(20) are always linear in θ_{gain} and θ_{bias} , and thus can be estimated using linear least squares methods.

D. Space-Time Joint Angles

Once the sensor parameters are updated, we turn to the ‘‘q-phase’’: optimizing for the joint angles. The goal is to maximize the log-likelihood,

$$q_t^* = \max_{q_t} \frac{\sum_m \|\hat{\mathbf{x}}_{mt} - \mathbf{x}_{mt}\|_2^2}{\sigma_x^2} + \frac{\sum_j \|\hat{p}_{jt} - p_{jt}\|_2^2}{\sigma_p^2}. \quad (21)$$

All we need is the Jacobian:

$$\frac{\partial \hat{p}_j(\theta, q_t)}{\partial q_t} = \begin{cases} \theta_{\text{gain}} & \text{rotary pots.} \\ \theta_{\text{gain}} \frac{\partial L(q_t)}{\partial q_t} & \text{linear pots.} \end{cases} \quad (22)$$

$$\frac{\partial \hat{\mathbf{x}}_m(\theta, q_t)}{\partial q_t} = \frac{\partial h_m(q_t; d_m)}{\partial q_t}, \quad (23)$$

where both $\frac{\partial L(q_t)}{\partial q_t}$ and $\frac{\partial h_m}{\partial q_t}$ are standard kinematics Jacobian available in almost every kinematics packages such as [8].

E. Dynamic Variance Adjustment

Distinct types of sensors produce residual of different dynamics range so the variances $\sigma_v^2 : \{\sigma_x^2, \sigma_p^2\}$ have to be adjusted to the right scale. We initialize these parameter to the corresponding sensors’ dynamic range and re-estimate the variance in each iteration based on the residuals.

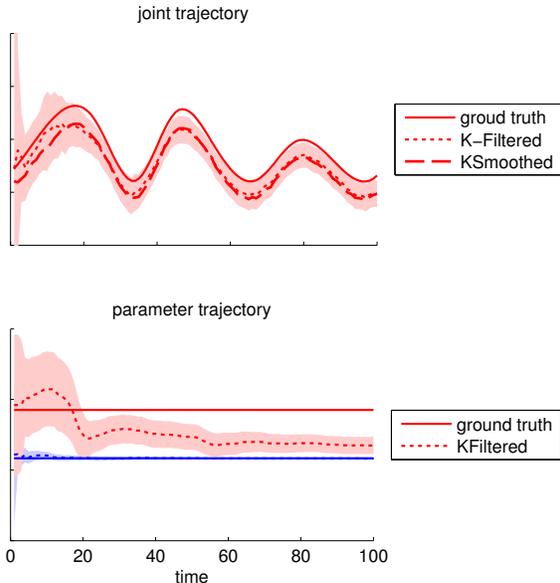


Fig. 4. Extended Kalman Filter for simultaneous calibration and tracking. The variance (shaded band) of parameter shrinks as more data is seen.

IV. RELATED WORK

We compare our formulation to two related approaches here and present the experimental comparison in the next Section.

A. Kalman Filter for Tracking and Calibration

The Extended Kalman filter (EKF) is an algorithm for tracking the state of a system with known dynamics and observation function from a noisy time series of observations. It has been applied to human skeleton tracking and kinematics identification [9]. Figure 4 gives an example how the EKF can be used to solve our tracking and calibration problem.

In this method, the state space consists of both joint angles q_t and parameters θ_t . Since the control sequence applied to the robot is assumed unknown, the dynamics equation contains only a drift term w with large variance Σ_q for time-varying joint angles and zero-variance for fixed parameters,

$$\begin{bmatrix} \theta_{t+1} \\ q_{t+1} \end{bmatrix} = \begin{bmatrix} \theta_t \\ q_t \end{bmatrix} + w, \quad w \sim \mathcal{N}(\mathbf{0}, \begin{bmatrix} \mathbf{0} & \\ & \Sigma_q \end{bmatrix}) \quad (24)$$

The observation function is same as what we use in STAC, see Sec.III. For both potentiometers \hat{p} and motion capture markers \hat{x} with observation noise z

$$v_t = [\hat{p}(\theta_t, q_t); \hat{x}(\theta_t, q_t)] \quad (25)$$

$$v_{t+1} = v_t + z, \quad z \sim \mathcal{N}(0, \sigma_v^2). \quad (26)$$

The EKF updates require linearization v_t around current state which needs the Jacobian

$$\frac{\partial v_t}{\partial(\theta_t, q_t)} = \begin{bmatrix} \frac{\partial \hat{p}}{\partial \theta_t} & \frac{\partial \hat{p}}{\partial q_t} \\ \frac{\partial \hat{x}}{\partial \theta_t} & \frac{\partial \hat{x}}{\partial q_t} \end{bmatrix} \quad (27)$$

The derivatives with respect to q are in (22) and (23) and those with respect to θ are also analytical. In practice,

because the linearized observation function is only valid at a local neighborhood around the current state, the EKF is prone to divergence without proper seeding of initial state.

Even with proper starting seed, once EKF loses track of the target due to observation noise at time t , the estimation will typically remain off after t . This is one of the major difference between STAC and EKF: while EKF tracks the parameters and space-time joint-angles sequentially in time, STAC jointly optimizes for them across time. Therefore, an erroneous estimate at one time frame would not propagate as in EKF. We will compare the robustness of EKF to STAC under various seeding and observation noise in Sec.VI-B.

B. Classical Regression-Based Approaches

Classical approaches (Sec. II-A) can be seen as a special case of STAC with zero joint potentiometer sensor noise $\sigma_p = 0$. In this way, the potentiometer term in the likelihood function (21) approaches infinity, effectively making it a constraint. Then the overall least squares problem can be simplified as

$$\begin{aligned} & \max_{\theta, d} \sum_t \max_{q_t} \sum_m \|\hat{\mathbf{x}}_{mt}(\theta, q_t) - \mathbf{x}_{mt}\|_2^2 / \sigma_x^2 \\ & \text{subject to } \hat{p}_{jt}(\theta, q_t) = p_{jt} \end{aligned} \quad (28)$$

If the constraints are all about simple one-joint-to-one-potentiometer, such that the joint angles can be inferred from measurement analytically, we can re-write the constraints as $q_j = q_j(p_j, \theta_j)$. Plugging q_j into the objective function in place of $q_j(\cdot)$, we obtain

$$\max_{\theta, d} \sum_t \sum_m \|\hat{\mathbf{x}}_{mt}(q(\hat{p}_t, \theta)) - \mathbf{x}_{mt}\|_2^2, \quad (29)$$

which is identical to (4)

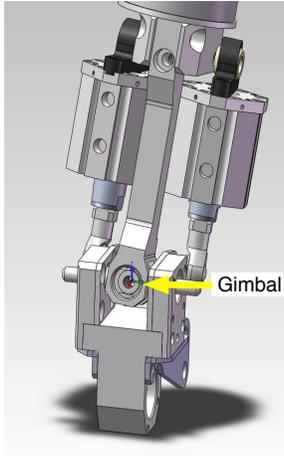
V. EXPERIMENT SETUP

We performed experiments on a complex pneumatics-based humanoid robot [10] named ‘‘Diego San’’ as well as a dexterous tendon-driven hand [4].

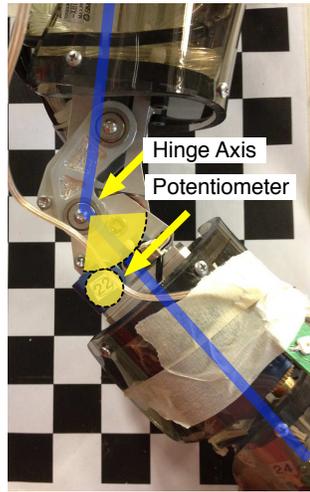
A. Humanoid

Figure 5(c) shows a picture of Diego San. It is a pneumatic humanoid with body parts proportional to that of a 1-year old human body. Among 38 joints, 4 are 2-DoF Gimbal joints (as in Fig. 3(d)) with two linear potentiometers measuring the length of the two pneumatic cylinders (e.g., neck, Fig. 5(a)); the rest of the joints are hinge type (as in Fig. 3(a)) with gear transmitted rotary potentiometers (e.g., elbow, Fig. 5(b)).

We attached 24 markers to every other link counting from the baselink taking advantage of the fact that the rotation axes of adjacent joints are mostly non-parallel. In this way, we were able to perform full-body sensor calibration without putting markers to every link. During motion capture, the robot was driven by a simple PID controller to move the joints randomly. Both marker positions and potentiometer readings were captured synchronously at 100Hz for 250 seconds. The traces are visualized in Fig.5(d).



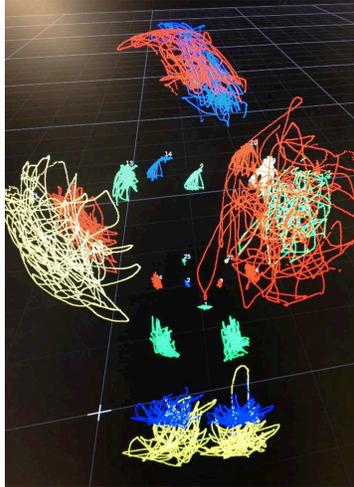
(a) 2-DoF Gimbal joint for Neck



(b) Hinge joint for elbow



(c) The robot



(d) Marker trajectories

Fig. 5. Diego San – the humanoid robot used in this study.

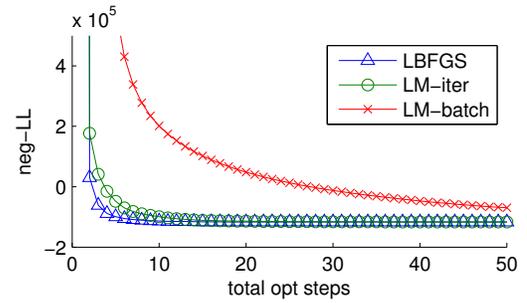
The kinematic model was extracted from the CAD file provided by the manufacturer (Kokoro Robotics), which includes link lengths, joint locations and orientations. The baselink of the robot is the waist, which was hung from a stable crane. Therefore we could safely assume that the transformation between the robot baselink and motion capture coordinate systems was constant.

B. Tendon Driven Hand

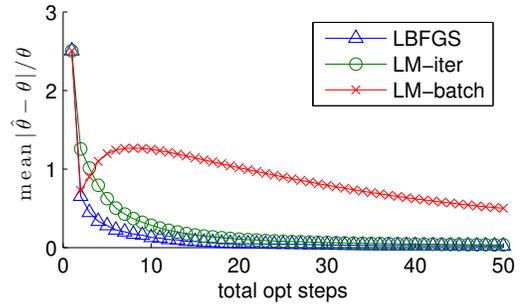
The dexterous hand by *Shadow Robot* is a human sized hand with 24 joints [4] (Fig.7(c)). The joints are actuated by pairs of tendons with the pneumatic pistons mounted at the fore-arm. Each joint has a Hall-effect joint angle sensor and the tendon lengths are also measured at the pistons. The tendon lengths are functions of one or more joint angles depending on the anchor points.

VI. SIMULATION-BASED EXPERIMENTS

To evaluate how precise our methods can recover the unknown parameters, we started with a set of random ground-



(a) negative likelihood



(b) mean relative error between estimated parameter $\hat{\theta}$ and ground truth θ

Fig. 6. Performance comparison of optimization algorithms in “q-phase”.

truth parameters internally, and then generated simulated noisy sensor observations.

A. Selection of Optimization Algorithms for q-phase

Here we explore different strategies for optimizing the joint angles in q-phase. The simulation experiments were performed using synthesized 200 random joint angles and corresponding noisy observations from the Hand robot model. The initial seeding parameters and joint angles are all zero except for the rotation matrix which is set to the identity.

LM-batch: To start, we optimize for the joint angles until convergence and then solve for sensor parameters until convergence. This gives the Levenberg-Marquardt method enough steps to find the right step size. However, we observed that allowing full LM convergence tended to drive the optimization process into local minima before the sensor parameters had a chance to settle into the correct region.

LM-iter: In addition, we observed that after few alternations, LM converged in less than 5 iterations without much progress on the objective value. To address this problem we tested a second approach in which we alternated between joint angles and sensor parameters after each LM step iteration.

BFGS: In addition to LM we also evaluated another popular optimization algorithm, BFGS. When computing the gradient $\partial LL / \partial q$, we first optimize for the parameters until convergence.

Figure 6(a) and 6(b) show the negative likelihood of the objective function and mean-relative-error between the estimated parameters and ground truth parameters. We use

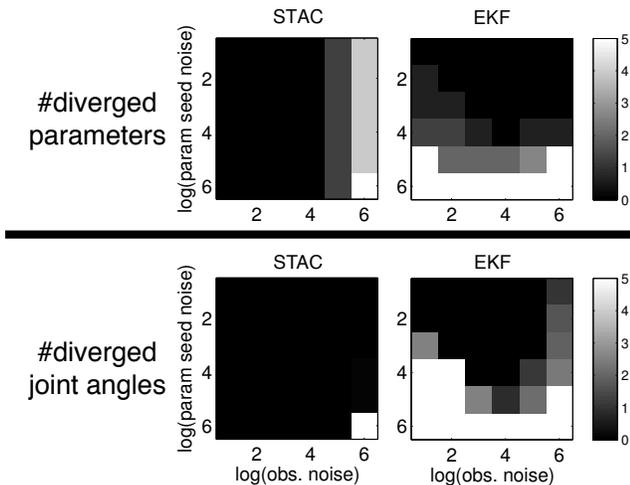


Fig. 7. The number of diverged parameters (top row) and space-time joint angles (bottom row) under various parameter seeding error and sensor observation noise. The first column is the seeding parameter error for comparison.

relative error because the range of different parameters is quite different.

It is observed that BFGS performs best immediately followed by LM-iter. LM-Batch converged slower and the parameter actually diverged in the first 10 iterations but recovered later.

B. STAC vs Kalman Filter - Resistance to Noise

We analyzed the sensitivity of the different algorithms to the quality of seeding parameter as well as observation noise. Typically the seeding parameters are from manual calibration. While precise manual calibration is time-consuming, rough eyeballing measurement is generally enough to get the algorithm to converge to the correct local minimum.

We first synthesized smooth random robot movement traces for 500 time steps within the nominal joint limits. Next we added various amount of Gaussian noise to both seeding parameter and the simulated sensor readings (3d markers and generalized potentiometers).

Then the data was fed to both STAC and EKF to evaluate how well they tracked the joint angles and parameters over time. The seeding and observation noise σ_v in EKF and in STAC were set to match the injected noise.

Note that for EKF, the initial state consists of both initial parameter and joint angle for the first frame. We set EKF initial joint angle q_0 to ground truth as it diverges immediately otherwise. On the other hand, STAC required seeding for both parameter and the entire joint angle trajectory. We therefore initialized every frame to the first frame given to EKF.

The performance evaluation was separated into two parts: the accuracy of estimated parameter and space-time joint angles. Figure 7 plots the number of diverged variables as gray level images. Divergence is quantified by thresholding the distance between the estimated and ground truth variable. With regard to estimation of sensor parameters we found

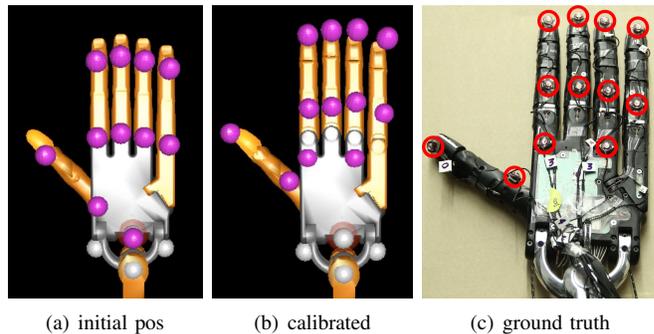


Fig. 8. Hand robot: marker positions before/after calibration.

that EKF’s performance is quite robust to the presence of sensor noise but it is very sensitive to noise in the seeding parameter values. STAC was much more robust than EKF to noise in the seeding parameter values but slightly more sensitive than EKF when large amount of sensor noise was present. With regard to the tracking of joint angles, STAC greatly outperformed EKF.

VII. EXPERIMENTS WITH PHYSICAL ROBOTS

We evaluated the models learned by STAC using two different robots (Diego San and the tendon driven dexterous hand). The evaluation criteria were based on the precision of the estimated marker positions, and the ability to fit novel data.

A. Identified Marker Position

As we saw in the previous section, STAC is quite robust to errors in the parameter initialization. In both robots, we initialized the marker positions to the origin of the link it is attached to. Typically, the origin of a link is on the joint connection to its parent link. Fig.9(a) and Fig.8(a) shows the initial position.

With Diego San we used manual calibration of joint angles as seed values while for the dexterous hand we simply initialized all the parameter to zero. After optimization, we compared the estimated marker position to the pictures of the robots side-by-side in Fig. 8(b) and Fig.9(b) The close correspondence again verified that our calibration procedure worked properly.

B. Cross-Sensor Prediction on Novel Data

As ground truth parameters of to-be-calibrated robots are unknown, we employ a training/testing data approach to evaluate model accuracy. The collected traces were split into training and testing sets. First we estimated system parameters from the training set and then use these parameter to predict values of one type of sensor in the test set given other sensor readings. We report the mean absolute error in Tbl. I and Tbl. II. The prediction is obtained by first running the joint optimization algorithm given the selected subset of sensors (left column), and then use the obtained joint angles to infer the target sensor value (column header). For example, the first column “3D-marker” reports the mean

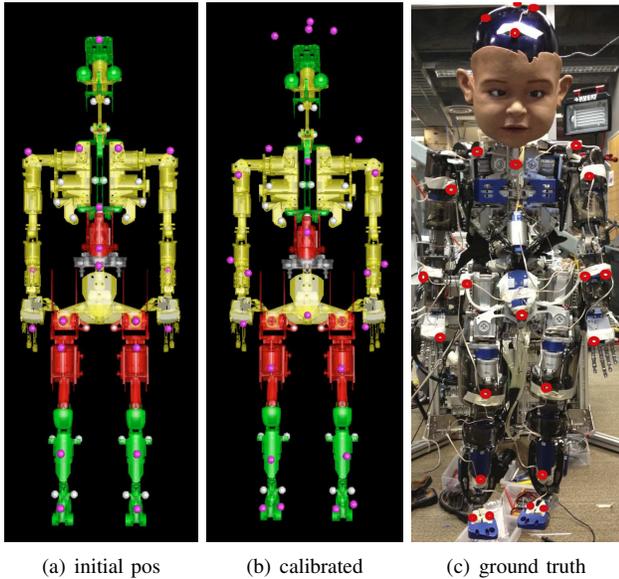


Fig. 9. Diego San Humanoid: marker positions before/after calibration.

TABLE I

HAND: CROSS SENSOR PREDICTION ON NOVEL DATA (MAE).

Sensor Set	3D Marker(m)	Joint (rad)	Tendon (m)
Joint(manual)	0.0201	0.2610	0.0022
Marker (stac)	(0.0026)	0.1384	0.0014
Joint (stac)	0.0119	(0.0020)	0.0010
Tendon (stac)	0.0077	0.0895	(0.0007)
Joint+Tendon (stac)	0.0095	(0.0164)	(0.0009)
Joint+Tendon+Marker (stac)	(0.0023)	(0.0313)	(0.0009)

distance between predicted and measured marker positions given individual or combination of other redundant sensors. The values in parentheses predict the same type of sensors, which can be seen as fitness of the model to the data.

For the hand robot (Tbl. I), the movement of a joint is observed by all three types of sensors simultaneously. In fact, it is possible to predict joint angle given only one type of sensor. Comparing to the manual calibration done by manufacturer, STAC reduced the error by half in all sensor types. For Diego San (Tbl. II), we performed a very coarse manual calibration for joint/tendon potentiometer gains and biases. We did not calibrate the markers. Comparing to manual calibration, STAC reduced marker error 7.3-fold when using joint/tendon sensors to infer joint angles.

The last row in Tbl. II reports the performance when the parameters are trained with the classical transparent joint-angle approach (see Sec. IV-B) assuming noise free joint angle sensor ($\sigma_p = 0$). The performance is worse than STAC.

TABLE II

HUMANOID: CROSS SENSOR PREDICTION ON NOVEL DATA (MAE).

Sensor Set	Marker(m)	Joint Pot(volt)	Tendon (m)
Joint+Tendon(manual)	0.0836	0.0000	0.9707
Joint+Tendon(stac)	0.0114	(0.0012)	(0.0028)
Joint+Tendon (stac- $\sigma_p = 0$)	0.0199	(0.0000)	(0.0035)

VIII. DISCUSSION

We proposed an efficient approach “STAC”, that jointly estimates sensor parameters as well as joint angles from multiple redundant sensors. Contrary to previous approaches, STAC can handle complex biologically inspired configurations in which the mapping from sensors to joint angles is one to many (i.e. not a function). This allows STAC to handle a much wider range of sensors than classical methods, like linear length sensors linking multiple joints. With the aid of multiple markers, our approach converges with little or no initialization. The algorithm was evaluated on complex 38-joint humanoid robot as well as a 24-joint tendon-driven hand – with very good results.

ACKNOWLEDGE

Support for this work was provided by NSF grants IIS-INT2-0808767.

REFERENCES

- [1] S. Aoyagi, A. Kohama, Y. Nakata, Y. Hayano, and M. Suzuki. Improvement of robot accuracy by calibrating kinematic model using a laser tracking system-compensation of non-geometric errors using neural networks and selection of optimal measuring points using genetic algorithm. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5660–5665. IEEE, 2010.
- [2] E. Fichter. A stewart platform-based manipulator: general theory and practical construction. *The International Journal of Robotics Research*, 5(2):157–182, 1986.
- [3] J. Hollerbach and C. Wampler. The calibration index and taxonomy for robot kinematic calibration methods. *The international journal of robotics research*, 15(6):573–591, 1996.
- [4] V. Kumar, Z. Xu, and E. Todorov. Fast, strong and compliant pneumatic actuation for dexterous tendon-driven hands.
- [5] M. Lee, D. Kang, Y. Cho, Y. Park, and J. Kim. The effective kinematic calibration method of industrial manipulators using igps. In *ICCAS-SICE, 2009*, pages 5059–5062. IEEE, 2009.
- [6] D. Mellinger and V. Kumar. Minimum snap trajectory generation and control for quadrotors. In *2011 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2520–2525. IEEE, 2011.
- [7] O. Sorkine. Least-squares rigid motion using svd. *Technical notes*, 2009.
- [8] E. Todorov. Mujoco: A physics engine for model-based control (under review), 2011a. URL <http://www.cs.washington.edu/homes/todorov/papers/MuJoCo.pdf>.
- [9] E. Todorov. Probabilistic inference of multijoint movements, skeletal parameters and marker attachments from diverse motion capture data. *IEEE Transactions on Biomedical Engineering*, 54(11):1927–1939, 2007.
- [10] E. Todorov, C. Hu, A. Simpkins, and J. Movellan. Identification and control of a pneumatic robot. In *2010 3rd IEEE RAS and EMBS International Conference on Biomedical Robotics and Biomechanics (BioRob)*, pages 373–380. IEEE, 2010.