Journal name
000(00):1–13

DOI:doi number
http://mms.sagepub.com

# Articulated Pose Estimation Using Tangent Space Approximations

**J. Brookshire and S. Teller**
*MIT Computer Science and Artificial Intelligence Laboratory, MIT, Cambridge, MA 02139*

**Abstract**

We describe an algorithm to estimate the pose of a generic articulated object. Our algorithm takes as input a description of the object and a potentially incomplete series of observations; it outputs an on-line estimate of the object's configuration. This task is challenging because: (1) the distribution of object states is often multi-modal; (2) the object is not assumed to be under our control, limiting our ability to predict its motion; and (3) rotational joints make the state space highly non-linear.

The proposed method represents three principal contributions to address these challenges. First, we use a particle filter implementation which is unique in that it does not require a reliable state transition model. Instead, the method relies primarily on observations during particle proposal, using the state transition model only at singularities. Second, our particle filter formulation explicitly handles missing observations via a novel proposal mechanism. Although existing particle filters can handle missing observations, they do so only by relying on good state transition models. Finally, our method evaluates noise in the observation space, rather than state space. This reduces the variability in performance due to choice of parametrization and effectively handles non-linearities caused by rotational joints.

We compare our method to a baseline implementation without these techniques and demonstrate, for a fixed error, more than an order-of-magnitude reduction in the number of required particles, an increase in the number of effective particles, and an increase in frame rate. We examine the effects of errors in the kinematic model and demonstrate a reduced dependence on state parametrization. The novel use of a precision matrix allows observations which do not provide complete 6-DOF pose information to be processed.

Source code for the method is available at `http://rvsn.csail.mit.edu/articulated`.

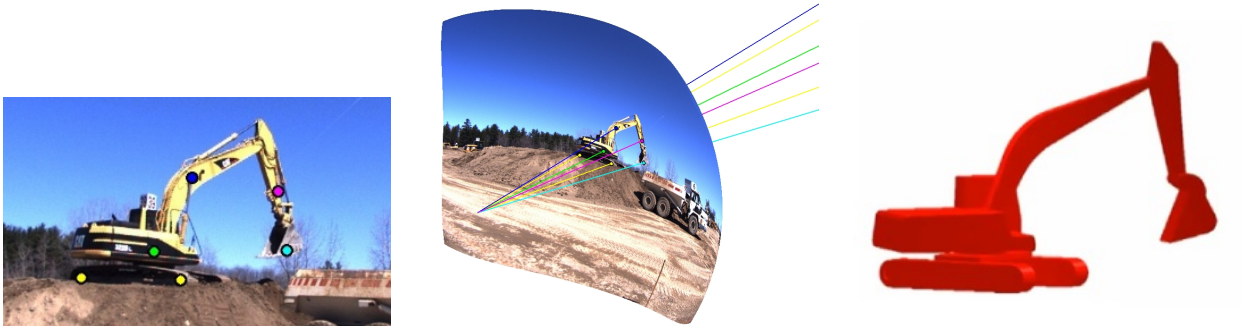
Keywords
articulated, tracking, particle filter, optimal importance function, kinematic chain, tangent space, noise projection

## 1. Introduction

To be useful in human-constructed environments, robots must interact with objects which have internal degrees of freedom (DOFs) or are constrained with respect to the environment. Such articulated mechanisms are found in homes (e.g., cabinet drawers, appliance doors, faucet handles, toaster levers, corkscrews, lidded boxes) and workplaces (e.g., construction equipment, pliers, screwdrivers, clamps, swivel chairs). Accurate estimates of the internal DOFs of such objects would facilitate manipulating or avoiding them. Knowledge of the state of an excavator, for example, would aid a robot navigating alongside it.

Our method takes as input:

1. A kinematic model of the articulated object, including: a function mapping joint values to observations (the "forward kinematics"); a function mapping small changes in configuration space to small changes in the observation space (the "Jacobian"); and any joint limits. In our application, this model is specified via a URDF file (Willow Garage, 2013).
2. A series of observations, sufficient for system observability (i.e., with sufficient information to estimate the object's configuration with finite variance (Kalman, 1961)). Observations may be noisy or incomplete, i.e., not included in each sensor update.



(a) Observations are 2D pixel locations on the articulated excavator.

(b) Each pixel location defines a ray observation in 3D space, emanating from the camera origin.

(c) Recovering the joint values enables 3D reconstruction.

**Fig. 1.** Example input and 3D reconstruction for an excavator.

The filter estimates the pose of one link (the "base pose") and the value of each joint parameter. In the excavator example (see Fig. 1), we might have independent feature detectors recovering pixel locations of the tracks, operator cab, boom, stick, and bucket. Of course, since these pixel measurements are not independent, the kinematic model can be used to improve the estimates.

In § 2 and § 3, we discuss related work and several alternative solutions to this tracking task, respectively. Of these alternative solutions, a "typical" implementation of a particle filter is discussed, along with several of its shortcomings. § 4 and § 5 detail our proposed solution, and § 6 demonstrates it on household dishwasher, PR2 robot, and construction-site excavator examples.

## 2. Background

### *2.1. The Kinematic Model*

The kinematic model is provided as an input to our system as a Unified Robot Description Format (URDF) file (Willow Garage, 2013). The URDF is parsed into a kinematic tree, consisting of joints and links, which we use to calculate forward kinematics, partial derivatives (Jacobians), and joint limits. This URDF can come from a variety of sources, including an object model database, user input, or some prior estimation process.

The object model database might contain URDF representations for a variety of pre-loaded articulated objects generated from Computer Aided Design (CAD) files. When a model is not available from the database, a human user might annotate sensor data to provide an *ad hoc* model. A user could employ a 3D drafting program to adjust model parameters to match 3D point cloud and image data, for example. In this way, the user can customize the model to match a particular object.

Related work has shown success automatically estimating a kinematic model from a moving, articulated object. These strategies generally proceed by tracking the movement of rigid bodies (links), then estimating the kinematic relationships

between these bodies (joints). In contrast, we observe the links and attempt to estimate the most likely configuration of the object (joint values).

Sturm et al. (2011), for example, give a method to compute the probability of particular joint types, finding a kinematic tree that maximizes the joint probability over all joints, offset with a penalty for model complexity. The authors construct a graph where vertices correspond to links and edges correspond to joint type (prismatic or rotational). The authors then use a combination of consensus sampling and Expectation Maximization (EM) (Dempster et al., 1977) to determine the most likely kinematic parameters.

Katz et al. (2012) also give a method to recover the DOFs of an articulated system. The authors actively manipulate an object and track image features. Using spatial, temporal, and appearance criteria, the features are segmented into clusters of rigidly co-moving parts. The resulting rigid bodies are then considered pairwise and classified as independent or as related by a fixed, rotational, or prismatic joint.

In an effort to enable door and drawer manipulation by robots, other researchers have also shown how to estimate the state of such objects. These methods are often specific to a single revolute or prismatic joint (e.g., Ruhr et al., 2012; Jain & Kemp, 2010; Meeussen et al., 2010) and focus on estimating the geometric parameters (e.g., door radius or drawer traversal axis).

## *2.2. Generic Articulated Object Tracking*

Much of the research in articulated object tracking is focused on tracking parts of the human body (see § 2.3). The only articulated tracking work validated on several different kinematic objects, of which we are aware, is by Comport et al. (2007). Comport tracks an articulated object by optimizing a single hypothesis at each time step. The hypothesis is projected into an image and errors in edge correspondences are correlated to velocities of the kinematic chain. The technique is demonstrated on a rotational joint, prismatic joint, helical joint, and a 2-DOF robotic arm. Their primary contribution is an efficient computation of the Jacobian between edge errors (i.e., errors between image edge detections and projected edges) and joint velocities. Their technique is specific to image detections and, as it maintains a single greedy hypothesis, is suited for uni-modal distributions.

## *2.3. Articulated Human Tracking*

Deformable part models (Felzenszwalb et al., 2008) have been widely used for object detection, especially for person tracking. These methods attempt to learn a model of each articulated object and to track it. Learning typically occurs in image space, hindering applicability to general mechanisms and motions.

In related work, Deutscher et al. (2000) used what they referred to as an "annealing particle filter" to estimate the configuration of a 29-DOF model of the human body. They employed a series of weighting functions designed to iteratively guide particles toward a global maximum. Although effective, the weighting functions must be customized to the application (in their case, the human body) and the method is not easily applicable to our situation where the features and kinematic tree are not known in advance.

Whereas we focus on situations where the process model is poor or unavailable, other researchers have focused on developing more accurate motion models. Urtasun et al. (2006) developed a Gaussian process technique which learns motion models. The learned model allows the pose of the human to be estimated during periods of occlusions, but is not appropriate for less predictable motions. Ziegler et al. (2006) tracked the upper torso using a 14-DOF model. Although effective, their UKF-based method would not handle multi-modal distributions and is specialized to 3D point cloud data.

Qu & Schonfeld (2007); Pan & Schonfeld (2008) recognized the complexity of tracking articulated object with particle filters (due to the high number of DOFs and resulting large number of particles) and decomposed the task of tracking a single, large kinematic chain into tracking smaller chains. Each link was tracked in image space and optimized in a pairwise

fashion with links connected via an immediate joint. With this simplification, they demonstrated real-time performance. However, the algorithm will generally fail to produce a global, joint optimization.

Research interests have also focused on tracking the human hand. Rehg & Kanade (1995); Cham & Rehg (1999); Rehg et al. (2003) tracked a 9-DOF model of two fingers on a hand. The authors developed a 2D kinematic chain which approximated a 3D kinematic chain and showed that the motion of the 2D chain is well defined even along the unobserved depth axis of the camera. By contrast, our method relies on the state transition model when unobserved dimensions arise.

## *2.4. Lie Groups & Lie Algebras*

In § 4, we rely on dual quaternions (DQ) and Lie algebra for processing in 6-DOF. DQ's are an eight element representation of rigid body transformations with two constraints (analogous to how quaternions correspond to $SO(3)$). Although the DQ and its Lie algebra are not required for our method, we found them a convenient way to process elements of $SE(3)$. A review of DQ's is beyond the scope of this paper; the interested reader should see (McCarthy, 1990, p. 53-62) for details. Here, we make use of the DQ space $\mathbb{H}$, but any representation of $SE(3)$ is appropriate.

Lie groups are smooth manifolds for which associativity of multiplication holds, an identity exists, and the inverse is defined (Govindu, 2004; Kanatani, 1990); examples include $R^n$, $SO(3)$, and $SE(3)$. However, $R^n$ is also a vector space (allowing addition, scaling, and commutativity), but $SO(3)$ and $SE(3)$ are not. For this reason, points in $SO(3)$ and $SE(3)$ cannot simply be interpolated or averaged.

Lie algebra describes a local neighborhood (i.e., a tangent space) of a Lie group. So the Lie algebra of DQ's, $\mathfrak{h}$, can be used to express a vector space tangent to some point in $\mathbb{H}$. Within this vector space, DQ's can be arithmetically manipulated. Once the operation is complete, points in the Lie algebra can be projected back into the Lie group.

The logarithm maps from the Lie group to the Lie algebra; the exponent maps from the Lie algebra to the Lie group. Both mappings are done at the identity. For example, if we have two elements of a Lie group $\{u, v\} \in \mathbb{H}$, the "box" notation of Hertzberg et al. (2011) expresses the difference between $v$ and $u$ as $d = v \boxminus u$. (Here, $\{u, v\}$ are each eight-element vectors and $d$ is a six-element vector.) That is, the box-minus operator connotes $d = 2\log(u^{-1} \circ v)$ where $d \in \mathfrak{h}$. In the Lie group, $u^{-1} \circ v$ is a small transformation relative to the identity, i.e., relative to $u^{-1} \circ u$ (see Fig. 2). (The factor of two before the $\log$ is a result of the fact that DQ's are multiplied on the left and right of the vector during transformation.)

Similarly, the box-plus addition operator Hertzberg et al. (2011) involves exponentiation. If $d \in \mathfrak{h}$, then $\exp d \in \mathbb{H}$. If $d = 2\log(u^{-1} \circ v)$, then $u \circ \exp\frac{d}{2} = u \circ \exp(\log(u^{-1} \circ v)) = u \circ u^{-1} \circ v = v$. Since $d$ applies a small transform to $u$, we use the box-plus operator to write $v = u \boxplus d = u \circ \exp\frac{d}{2}$.

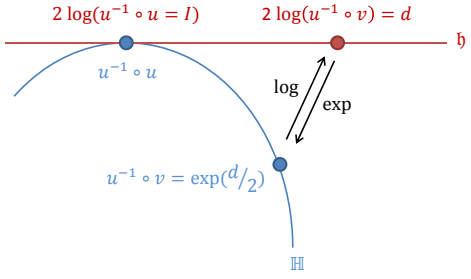



**Fig. 2.** The mapping between the Lie group, $\mathbb{H}$, and the Lie algebra, $\mathfrak{h}$, is performed at the identity, $u^{-1} \circ u$.

This definition of the difference between DQ's yields a Gaussian distribution as follows: imagine a Gaussian drawn on the line $\mathfrak{h}$ in Fig. 2. Exponentiating points on this Gaussian "projects" the distribution onto $\mathbb{H}$.

DQ's and the Lie algebra were also used by Kavan et al. (2008) and Eggert et al. (1997) for optimization over rigid body transformations. Kwon & Lee (2010) used the Lie algebra in a particle filtering application for SLAM. Choi & Christensen (2011) also employed the Lie algebra in particle filters for visual tracking.

## *2.5. Particle Filter (PF)*

A particle filter (Isard & Blake, 1998) maintains a discrete approximation to a probability distribution using a collection of state hypotheses, or particles. During each cycle of the filter, particles are proposed, weighted, and possibly duplicated/eliminated. During the proposal phase, the ideal proposal distribution, or optimal importance function (OIF) (Doucet et al., 2000), is $P(x_k \left| x_{k-1}^{(i)}, z_k \right.)$. In other words, the probability of a new state, $x_k$, depends on the $i$-th previous particle's state, $x_{k-1}^{(i)}$, and the new observation, $z_k$. Using Bayes rule, it can be shown that:

$$P\left(x_k \left| x_{k-1}^{(i)}, z_k \right.\right) = \frac{P\left(z_k|x_k\right) \cdot P\left(x_k|x_{k-1}^{(i)}\right)}{P\left(z_k|x_{k-1}^{(i)}\right)} \tag{1}$$

where $P\left(z_k|x_k\right)$ corresponds to the observation model and $P(x_k|x_{k-1}^{(i)})$ corresponds to the state transition model. The denominator, $P(z_k|x_{k-1}^{(i)})$, is the probability of the observation, given the previous state. In general we will not know this quantity, but can marginalize over the current state:

$$P\left(z_k|x_{k-1}^{(i)}\right) = \int P\left(z_k|x\right) \cdot P\left(x|x_{k-1}^{(i)}\right) dx \tag{2}$$

Note that the two terms in the integral are analogous to the two terms in the numerator of Eq. 1 and can be readily calculated. Doucet et al. (2000) also notes that the weight update for the OIF is:

$$w_k^{(i)} = w_{k-1}^{(i)} \cdot P\left(z_k|x_{k-1}^{(i)}\right) \tag{3}$$

It is often impossible to sample directly from the OIF. As a result, the particle filter samples from an approximation and uses an importance weight to compensate. It is important that the approximation yield particles from the high-probability regions of the OIF. To achieve this, one of two approximations is often made:

1. When the state transition model is more accurate than the observations, we can approximate that $x_k \perp z_k$, i.e., that for the purposes of proposal, the state is independent of the observations. The OIF then conveniently becomes the familiar $P(x_k|x_{k-1}^{(i)})$, i.e., the state transition model. The weight update then becomes $w_k^{(i)} = w_{k-1}^{(i)} \cdot P\left(z_k|x_k\right)$ (Thrun et al., 2005).
2. When the observation model is more accurate than the state transition model, we can approximate that $x_k \perp x_{k-1}^{(i)}$, and propose particles based only on observations. In this case, the OIF becomes $P\left(x_k|z_k\right)$. This proposal function is often complex, requiring estimation of a state from observations (e.g., via inverse kinematics). However, this choice enables the incorporation of observations during particle proposal.

The first approximation (1) is most typical, as it produces simple equations. However, because it relies only on the state transition model to propose particles, the next generation of particles is only as good as that model. In our case, a state transition model specific to the articulated object is unavailable, and we rely on generic zero-velocity and constant-velocity models. These models are relatively poor and result in particles with low observational probability.

Instead, our solution incorporates observations during particle proposal (2). Grisetti et al. (2005) successfully demonstrated such an approach for a mapping task. The state transition model for the robot's motion was often noisier than the sensor data, so the method used LIDAR observations, processed through a scan-matcher, to propose new particles. The result was that generated particles had higher probability and, thus, fewer particles were required. We adopt the same approach here, but do not require a function to convert directly from observations to an object configuration (i.e., we do not require a scan-matcher or equivalent).

A common performance metric for particle filters is the number of effective particles:

$$N_{\text{eff}} = \left( \sum_{i=1}^{N_s} \left( w_k^{(i)} \right)^2 \right)^{-1} \tag{4}$$

where, again, $w_k^{(i)}$ is the weight of the $i$-th particle at time $k$. When all particles have roughly the same weights and represent the distribution well, $N_{\text{eff}}$ will approach the number of particles, $N_s$. When the filter becomes degenerate, $N_{\text{eff}}$ will approach unity. If it falls below some threshold, we use $N_t = N_s/5$, resampling is performed.

## 3. Alternate Solutions

### *3.1. Optimization Only*

To motivate our method, we first consider several alternative solutions and discuss their shortcomings. Given that we have relatively accurate observations and a kinematic model, one possible solution is to simply optimize over configurations at each time step. (It is also possible to accumulate data over several time steps and then optimize. However, the system would then be a filter; we explore these ideas in later sections.)

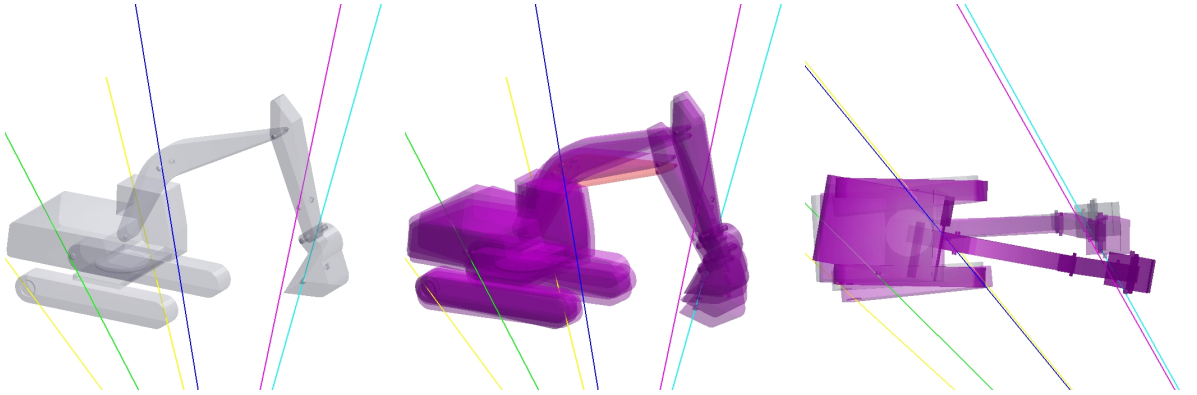
First, although this approach is simple, the system may be redundant (i.e., have multiple solutions). An optimization technique, which fails to account for the history of previous observations, may have difficulty converging to the correct solution. These redundancies may result from the kinematic structure; that is, the articulated object may have several configurations that will satisfy the observations. (A filtering or batch optimization technique would address this concern; see next sections.) Or, the system may not be observable using data from only the current time step. This effect may be the result of degeneracies or missing observations. Second, no estimate of uncertainty is provided and that knowledge is often useful for later processing. Third, joint limits are not inherently supported and may require additional constraints. Fourth, the optimization is often performed in the state space (and will experience parametrization dependence).

To highlight the multi-modal nature of the problem, consider the task of tracking an excavator at a construction site. Suppose the observations are detections made on an image (Fig. 1a) and correspond to rays in 3D space (Fig. 1b). An optimization-based solution would attempt to find a configuration of the excavator which most closely adheres to these observations.

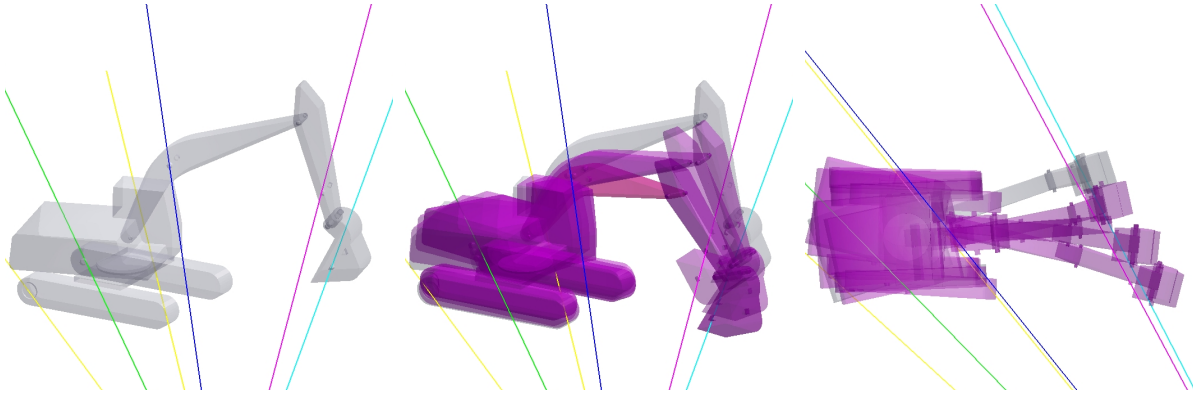
Fig. 3 shows three example configurations for an excavator. For each configuration, several thousand initial conditions were sampled and a Levenberg-Marquardt (Bertsekas, 1999) non-linear optimizer was used to find the nearest configuration. The rays corresponding to the observed pixels are depicted as lines emanating from the camera origin (not shown). For example, Fig. 3a shows the ground truth position of the excavator with observations; (b) and (c) show the modes of the optimized configurations. (Modes were obtained by performing EM on a Gaussian mixture.)

In Fig. 3(a)-(c), two discrete solutions are recovered. As the excavator's cab rotates clockwise, another configuration also explains the observations. Multiple solutions are found in (d)-(f). Here, the true position of the excavator is nearly parallel to the camera (to see this, note that the gray configuration in (f) is nearly perpendicular to the detection rays). The DOFs of the articulated object provide enough freedom to "slide" the excavator toward and away from the camera while still explaining the observations. In (g)-(i), the excavator's stick (link adjacent to the bucket) is not observed. Consequently, there are two possible configurations for the stick-bucket joint.

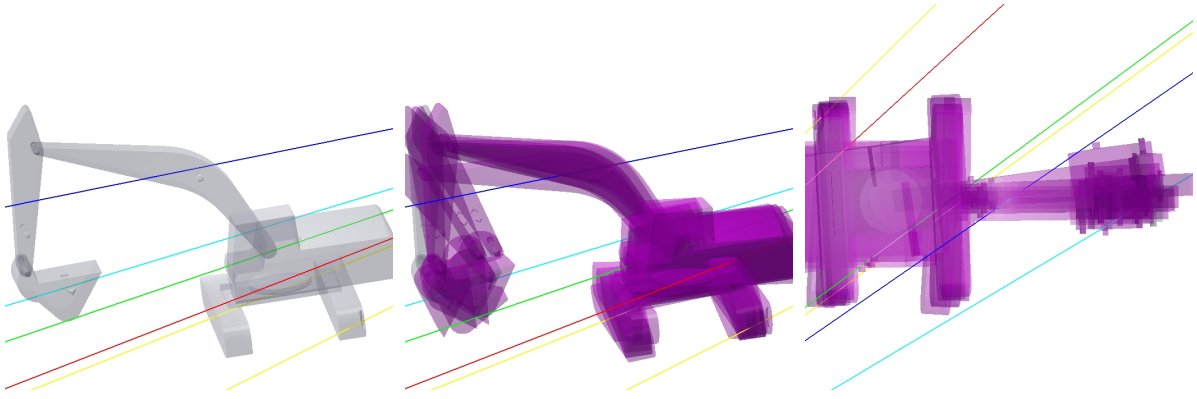
Clearly, we wish to incorporate previous observations and attempt to track the excavator. This led us to investigate filtering approaches. Even relying on previous states may be insufficient to avoid all local minima, however. As a result, we focus on particle filtering techniques which can maintain hypotheses at multiple modes and easily converge to a good solution when the configuration can be disambiguated.



(a) Scenario #1, Truth (b) Scenario #1, Alternate States, View 1 (c) Scenario #1, Alternate States, View 2



(d) Scenario #2, Truth (e) Scenario #2, Alternate States, View 1 (f) Scenario #2, Alternate States, View 2



(g) Scenario #3, Truth (h) Scenario #3, Alternate States, View 1 (i) Scenario #3, Alternate States, View 2

**Fig. 3.** Each row shows a sample scenario from different views (columns). The true pose is shown alone in the first column and in the background in other columns. The other renderings show alternative configurations which also explain the observations (rays). In all these examples, multiple modes explain the observations. Best viewed in color.

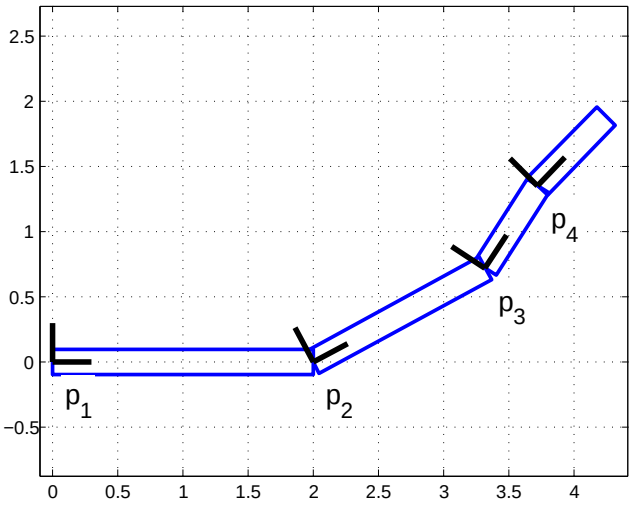



**Fig. 4.** An example planar articulated system with four rigid links and three revolute joints.

### *3.2. Baseline Particle Filter*

A filtering technique, which incorporates previous data, can alleviate some of the difficulties with optimization. A filter can use previous states to help estimate the current state. Here, we describe a "baseline" particle filter, which uses the simplest state transition proposal method. Unlike the optimization only strategy, the baseline PF can handle joint limits easily via rejection sampling, can handle partial observations, and provides a representation of uncertainty. It can also model multi-modal distributions. Although this solution is better than optimization, it exhibits several limitations discussed here. Later, this baseline will serve as a benchmark for our results.

The first problem with the baseline implementation is that it assumes a reasonably good state transition model. Although our system is provided with a kinematic model, we are not provided with a motion model. Instead, we are relegated to chose from generic motion models, such as zero-velocity or constant-velocity models, which may not predict state transitions well. As noted by Grisetti et al. (2007), this is particularly problematic when the observations are more accurate than the state transition model. When $P(x_k|x_{k-1})$ is more uncertain than $P(z_k|x_k)$, many particles must be sampled from $P(x_k|x_{k-1})$ to adequately sample the narrow $P(z_k|x_k)$. (A better solution, and part of our method, is to incorporate observations during proposal.)

A second problem with generic motion models is that they often suffer from parametrization dependence. That is, the performance of a filter which uses these generic models will depend on the mathematical expression of the state. To see this, suppose we wish to track the simple 2D kinematic linkage shown in Fig. 4 which is moving independently. A natural parametrization for the state space might be:

$$x = \begin{bmatrix} p_1 & \alpha_{12} & \alpha_{23} & \alpha_{34} \end{bmatrix}^T \tag{5}$$

where $p_1 \in SE(2)$ is the pose of the base link and $\alpha_{ij} \in \mathbb{R}$ are the joint angles between links $i$ and $j$. Further, the observations might be the $p_i \in SE(2)$ poses of each link:

$$z = \begin{bmatrix} p_1 & p_2 & p_3 & p_4 \end{bmatrix}^T \tag{6}$$

We propose from $P(x_k|x_{k-1}^{(i)})$. If we assume a zero-velocity model, we add Gaussian noise (AGN) to propagate the state, i.e., $x \sim N(\mu_x, \Sigma_x)$. An example of proposed particles is shown in Fig. 5a. We might also assume that our observations are corrupted by AGN, i.e., $z \sim N(f(\mu_x), \Sigma_z)$, where $f$ is the forward kinematics. The isocontours of such a distribution

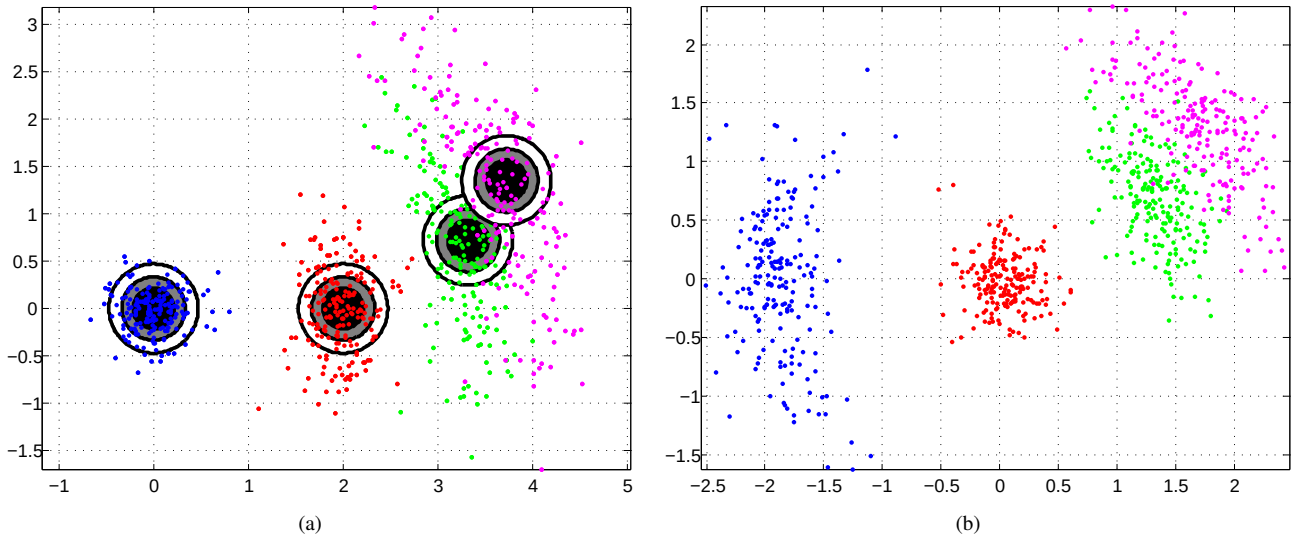



**Fig. 5.** (a)The dots show the link positions of the particles proposed from $P(x_k|x_{k-1}^{(i)})$ for links 1-4. The underlying contours illustrate the Gaussian $P\left(z_k|x_k\right)$. Notice that for link 4, for example, many particles are unlikely according to the observation model. (b) In the baseline method, altering the state parametrization also affects the proposed particles.

are also shown in Fig. 5a. It is clear that they are dissimilar — in the case of the system model, the distributions become increasingly "banana-like" as we proceed further out from the base link.

In the context of a particle filter, this dissimilarity means that many samples proposed from $P(x_k|x_{k-1}^{(i)})$ would have low weights due to the mismatch with $P\left(z_k|x_k\right)$ at the distal links. Wasted computation and degraded accuracy result.

Observe further that a slight change to the parametrization also affects particle proposal. To see this, consider an alternative formulation:

$$\hat{x} = \begin{bmatrix} \alpha_{21} & p_2 & \alpha_{23} & \alpha_{34} \end{bmatrix}^T$$

Here, the base link is taken as $p_2$, and AGN on this state produces the plot shown in Fig. 5b. Comparing Fig. 5a and Fig. 5b highlights that particle proposal in the baseline method depends on the state parametrization. To mitigate this problem, we will later see that we propose noise in the observation space, rather than the state space, because it is fixed in our application.

Many other particle filter formulations exist, in addition to the baseline method discussed later. Doucet et al. (2000) and van der Merwe et al. (2000), for example, apply an Extended Kalman Filter (EKF) and UKF (respectively) to each particle in the filter. Their formulations expand the state space for each particle, maintaining additional covariance parameters. The EKF approach is similar in spirit to our Jacobian tangent space; on the other hand, our formulation does not require a Gaussian approximation or require that a separate covariance be maintained for each particle.

### *3.3. Unscented Kalman Filter*

The UKF is another possible technique to track an articulated object. As in § 3.2, the state might consist of a base pose and the joint angles. The UKF calculates sigma points around the current state, predicts motion according to the state transition model, and updates the estimate given the new observations. The UKF models the uncertainty as a Gaussian, so the current state estimate is always unimodal and associated with a covariance matrix. The UKF also handles partial observations and, as Boyd & Vandenberghe (2004) describe, an additional optimization step can be used to handle joint limit verification.

Although the UKF may be a viable filtering technique for some objects, in general, the distribution of configurations for an articulated object will be multi-modal and cannot be captured by a single Gaussian. Since the UKF cannot handle such multi-modal distributions, we use a particle filter.

Like the baseline particle filter, the UKF also exhibits a state parametrization dependence. This occurs because the sigma point calculation (Julier, 2002) adds small perturbations in the state space. To see this, again consider the four-link kinematic chain in Fig. 4. A state parametrization of:

$$x = \begin{bmatrix} p_1 & \alpha_{12} & \alpha_{23} & \alpha_{34} \end{bmatrix}^T \tag{7}$$

yields sigma points as shown in Fig. 6a. On the other hand, a state parametrization of:

$$x = \begin{bmatrix} p_4 & \alpha_{12} & \alpha_{23} & \alpha_{34} \end{bmatrix}^T \tag{8}$$

yields the different sigma points shown in Fig. 6b. One hundred iterations of the same 50-frame simulation reveal that this difference affects the RMSE performance of the two parametrizations, as shown in Fig. 6c. As mentioned for the baseline particle filter, we propose noise in the observation space to address this dependence. It is important to note that this effect can be minimized by using a reliable state transition model. Again, however, we do not assume a reliable state transition model is available.

## 4. Our Method

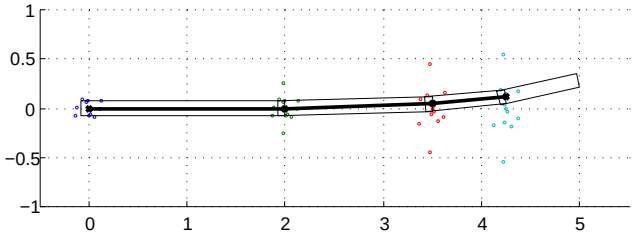
The previous section highlighted some of the shortcomings of alternative methods to track an articulated object. In this section, we describe an algorithm which addresses these issues. Because we need to track a state with a multi-modal distribution, we use a particle filter. Because our state transition model is inaccurate, relative to our observations, we need to incorporate the observations during particle proposal. Finally, because we wish to minimize the effects of state parametrization, we perform operations in the fixed observation space where possible, using the state space only when unavoidable.

Brookshire & Teller (2014) derived the particle proposal equations assuming both the state space and observation space were Euclidean. In the general case, however, the state space base pose (and possibly the observations) will be $SE(3)$, and Euclidean operations will be inappropriate. In this work, we make several changes to accommodate 3D articulated objects. First, for convenience, we standardize our observations to be 6-DOF poses with a possibly singular precision matrix (see § 5). Second, we re-derive our tangent approximation using the Lie algebra, and show that our conclusions for the Euclidean case are still applicable.

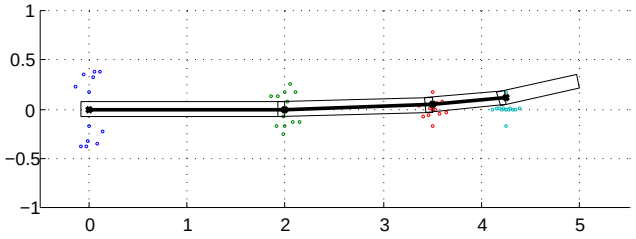
### *4.1. State and Observation Spaces*

During particle proposal, the algorithm considers the valid configurations of the articulated object as a manifold $\mathcal{M}$ in a higher-dimensional space. We chose this high dimensional space to be the observation space so that proposed particles correspond to observation noise and because this space is fixed in our application. However, other choices are possible (e.g., link poses could be used for noise proposal regardless of the observation or state representation). Regardless, the manifold is defined by the forward kinematics and observation function, $f(x)$, which converts from an $M \times 1$ state vector, $x$, to a point in the higher dimensional space. A complete observation $z$ is an $N$ dimensional point in the observation space.

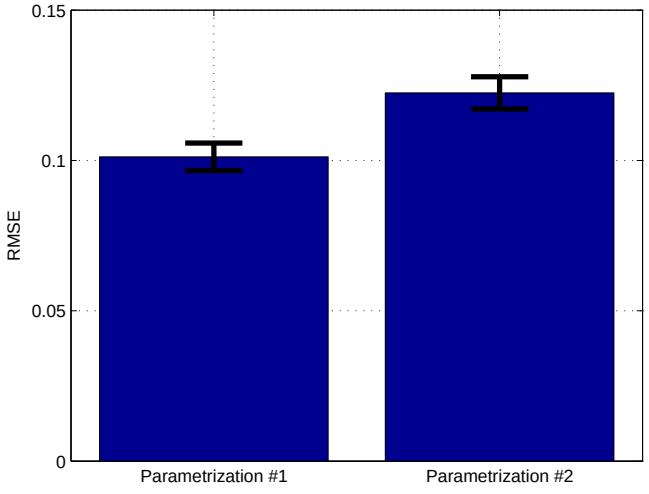
In Brookshire & Teller (2014), the observation space was considered to be $\mathbb{R}^N$. Here, our observations are 6-DOF poses and the observation space is a compounded $SE(3)$ space. Although any representation of $SE(3)$ is suitable for the derivation, we will use dual quaternions, $\mathbb{H}$, and their Lie algebra, $\mathfrak{h}$. The $\boxplus$ and $\boxminus$ operators express addition and subtraction, respectively, in the tangent space of $\mathbb{H}$. (See § 2.4 and Brookshire & Teller (2012).) Observations are $N \times 1$, $\mathcal{M} \subseteq \mathbb{H}^{N/8}$ (a DQ has 8 elements).




(a) Sigma points for parametrization #1 with base pose as $p_1$, the left-most link.




(b) Sigma points for parametrization #2 with base pose as $p_4$, the right-most link.




(c)

**Fig. 6.** Different state parametrizations result in different sigma points for the UKF (a), (b). The sigma points for links 1-4 are shown. In this situation, different tracking error (c) results. Error bars indicate one standard deviation. Compare with Our Method in Fig. 13.

The state is also not $\mathbb{R}^M$; rather, we treat the state as $\mathbb{H} \times \mathbb{R}^{M-8}$ encoding a 6-DOF base pose and the joint angles. For example, if $b \in \mathbb{H}$ is the base pose and $q_i \in \mathbb{R}$ are the joint angles, then the state is:

$$x = \begin{bmatrix} b & q_0 & \cdots & q_{M-8} \end{bmatrix} \tag{9}$$

A small perturbation in the state can be represented by a Lie algebraic velocity, $c \in \mathfrak{h}$, and joint velocities, $r_i \in \mathbb{R}$:

$$\delta = \begin{bmatrix} c & r_0 & \cdots & r_{M-8} \end{bmatrix} \tag{10}$$

where $\delta$ is $(\mathfrak{M} = 6 + M - 8) \times 1$. ($c$ could be calculated as the difference between two DQ's as described in § 2.4 or created as a noise perturbation in the tangent plane.) We then define an additional operator $\oplus$ such that:

$$x \oplus \delta = \begin{bmatrix} b \boxplus c & q_0 + r_0 & \cdots & q_{M-8} + r_{M-8} \end{bmatrix} \tag{11}$$

In this way, the base pose is updated by compounding transforms and the joint angles are added in a Euclidean fashion.

The particle proposal algorithm is presented in Function 4.1. Generally, it proceeds by:

1. Using available observations, $z_k$, to find the nearest valid configuration/state (i.e., the closest point on $\mathcal{M}$).
2. Approximating the OIF (Eq. 1) using a set $\mathcal{X}^{(i)}$ of discrete samples, generated by:
   (a) Adding noise in the observation (not state) space, then using a first-order Taylor approximation and projection to map the noise to the state space,
   (b) Adding noise in the state space only when the Jacobian is singular, and
   (c) Rejection sampling to satisfy joint limits.

As a result of (1), we can exploit observations during the proposal stage without requiring an inverse kinematic/observation function. As we will see, we can extract information from observations which are either redundant ($\dim(z_k) > M$) or incomplete ($\dim(z_k) < M$). As a result of (2a), the particle proposals are less dependent on the state parametrization because particle perturbations are created in the fixed observation space. Attribute (2b) handles degenerate observations by relying on the state transition model. In (2c), we ensure that joint limits are satisfied. Because rejection sampling can be expensive, we sample directly from the discrete approximation $\mathcal{X}^{(i)}$ when choosing $x_k^{(i)}$ (rather than re-approximating with a Gaussian, as by Grisetti et al. (2005)).

As mentioned, the algorithm is not provided with a state transition model; therefore, we select a model which will generalize to many different kinematic chains. In all our examples, we found a zero-velocity model sufficient. That is, we use the state transition model $P(x_k|x_{k-1}) \sim N(x_{k-1}, \Sigma_x)$. If another transition model were more appropriate, e.g., a constant-velocity model, the following techniques could be modified by adding the constant velocity, in addition to diffusion noise, in the null space. However, it is important to note that if the velocity is part of the state, calculation of the Jacobian will require the derivative of velocity terms. Higher order derivatives may be more difficult to calculate and risk amplifying noise.

### *4.2. Incorporating Observations During Particle Proposal*

The algorithm begins by considering each particle $x_{k-1}^{(i)}$ from the previous time step. For notational simplicity, let $x = x_{k-1}^{(i)}$. We wish to update this particle using whatever observations are available. Thus, we find a new point $m_k$ which is both near the previous particle and near the observations:

$$m_k = x \oplus \widehat{dx} \tag{12}$$

where $\widehat{dx}$ is a small change in the state space. We wish to find $m_k$ by minimizing the Mahalanobis distance between the observation and the configuration manifold. Further, the observations, $z_k$, may be incomplete. Let $Q$ be an $\mathfrak{N} \times \mathfrak{N}$ diagonal

matrix whose diagonal entries are one if the observation is made at time $k$ and zero otherwise. ($\mathfrak{N} = 6N/8$, because elements in $\mathbb{H}$ and $\mathfrak{h}$ are 8 and 6 elements, respectively.) Thus, $Q$ selects the valid observations.

$$\widehat{dx} = \underset{dx}{\operatorname{argmin}} \left[z_k \boxminus f\left(x \oplus dx\right)\right]^T Q^T \Sigma_{obs}^{-1} Q \left[z_k \boxminus f\left(x \oplus dx\right)\right] \tag{13}$$

where $\Sigma_{obs}^{-1}$ is a possibly singular precision matrix associated with the observations. Note that this precision matrix expresses a Gaussian uncertainty in the Lie algebra. Let $L^T L = \Sigma_{obs}^{-1}$ be an SVD factorization of $\Sigma_{obs}^{-1}$. Then

$$\begin{aligned}\widehat{dx} &= \underset{dx}{\operatorname{argmin}} \left[z_k \boxminus f\left(x \oplus dx\right)\right]^T Q^T L^T L Q \left[z_k \boxminus f\left(x \oplus dx\right)\right] \\ &= \underset{dx}{\operatorname{argmin}} \left|LQ\left(z_k \boxminus f\left(x \oplus dx\right)\right)\right|^2 \end{aligned} \tag{14}$$

The algorithm then finds a nearer point on the manifold by projecting onto a tangent plane (Fig. 7). We assume that $\mathcal{M}$ is well-approximated by a first-order Taylor series in a neighborhood corresponding to the observation noise. That is, we assume that:

$$f\left(x \oplus dx\right) = f\left(x\right) \boxplus J_x \cdot dx \tag{15}$$

where $J_x$ is the Jacobian of $f$ at $x$. We also note that $z_k = f\left(x\right) \boxplus dz$, where $dz$ is the vector between the previous particle and the current observation.

$$\widehat{dx} = \underset{dx}{\operatorname{argmin}} \left| LQ \left[ \left( \underbrace{f\left(x\right)}_{N\times 1} \boxplus \underbrace{dz}_{\mathfrak{N}\times 1} \right) \boxminus \left( f\left(x\right) \boxplus \underbrace{J_x\, dx}_{\mathfrak{N}\times 1} \right) \right] \right|^2 \tag{16}$$

Operations in $SE(3)$ do not commute and we cannot simplify to cancel $f(x)$, as in the Euclidean case. That is $(b \boxplus c) \boxminus (b \boxplus d) \neq c - d$. (In Eq. 16, the underline notations indicate the dimension of the vector.)

However, as shown by Brookshire (2013), it is still the case that:

$$\widehat{dx} = \underset{dx}{\operatorname{argmin}} \left|LQdz - LQJ_x\, dx\right|^2 \tag{17}$$

The pseudo-inverse, $(\cdot)^\dagger$, solves this problem:

$$m_k = x_{k-1}^{(i)} \oplus \left(LQJ_x\right)^\dagger LQ \left(z_k \boxminus f\left(x_{k-1}^{(i)}\right)\right) \tag{18}$$

(The pseudo-inverse may be calculated by several means; we use a modified version of the SVD calculation (Brookshire, 2013).)

Thus, $m_k$ is a configuration of the articulated object which is near the most recent observations (at least on the local tangent plane). The corresponding line 6 in Function 4.1 takes a single Gauss-Newton optimization step toward the observations. This process is illustrated in Fig. 10; in (a), an observation (dotted line) is made and there are four possible explanations for this observation (intersection of $\mathcal{M}$ and $z_k$). Each particle is optimized (with a single Gauss-Newton step) towards a local minimum with the new observations, (b). In this way, the particles are shifted toward higher probability regions.

It is important to note that the matrix product $(LQJ_x)$ may be singular for several reasons: (1) it is singular because the same dimensions are always in the null space; (2) it is singular due to a specific configuration; or (3) it is singular due to missing observations. We assume that (1) is not the case, as the system would be unobservable and unable to be reliably tracked. If (2) or (3) is the case, then there is a non-trivial null space. In that situation, there are dimensions of

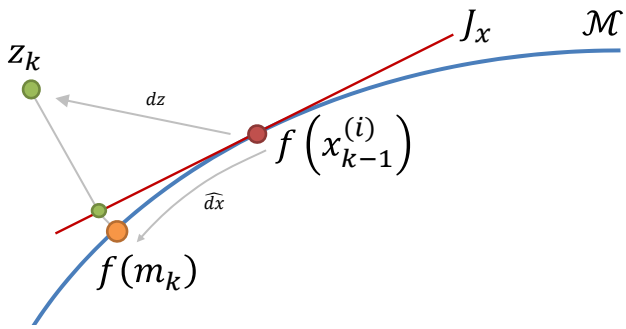



**Fig. 7.** The method updates the previous particle, $x_{k-1}^{(i)}$, with the observations, $z_k$, using the Taylor approximation and subsequent projection.

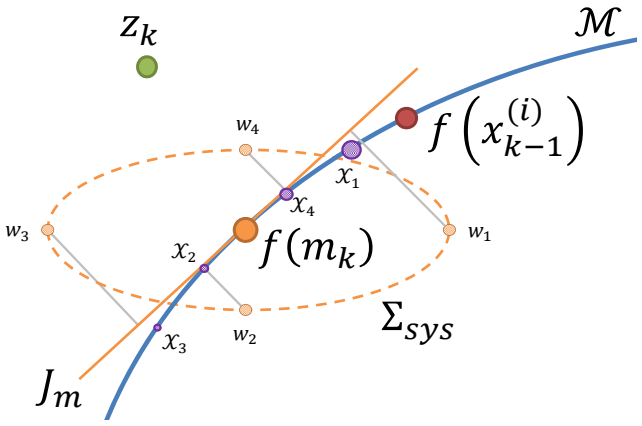



**Fig. 8.** A Taylor approximation and a projection relate noise in observation space to noise in state space.

$x_{k-1}^{(i)}$ which are unaltered in $m_k$. On the other hand, the dimensions that lie in the range are updated and "pulled" as close as possible to the observation. This has the desired result that all information is extracted from the observations, while the unobserved dimensions remain centered at $x_{k-1}^{(i)}$. This performance is consistent with our zero-velocity state transition model; alternative models could update $x_{k-1}^{(i)}$ by manipulating the null space. When $z_k$ has redundant observations and over-constrains $x_{k-1}^{(i)}$, Eq. 18 computes $\widehat{dx}$ by projecting onto $J_x$.

### *4.3. Diffusion During Particle Proposal*

We discretely approximate the OIF by randomly selecting points about $m_k$ and weighting. The algorithm creates a set $\mathcal{X}^{(i)} = \left\{ \begin{array}{ccc} \mathcal{X}_1 & \cdots & \mathcal{X}_P \end{array} \right\}$ of points centered around $m_k$ (lines 7-16). These $P$ points discretely approximate the OIF and are sampled in line 23 to select $x_k^{(i)}$, the next generation particle.

Because the observation space remains fixed in our application, our algorithm proposes noise in it, rather than the state space. Along some dimensions, however, it may not be possible to convert uncertainty in the observation space to uncertainty in the state space. This can occur when the Jacobian is singular due to a specific joint configuration or missing observations. When this occurs, the algorithm proposes using state uncertainty in this null space.

It is interesting to note that the observation space can be thought of as an over-parametrization of the state space. To see this, consider a noise-free set of observations. Assuming $N > M$, then the observations use more parameters to describe the state than are strictly necessary. An analogous situation was faced by Brookshire & Teller (2012), where an 8-element DQ was used to represent a 6-DOF pose. In that situation, noise was expressed in the Lie algebra, a locally tangent space. Here, we project the noise into a local tangent approximation.

As shown in Fig. 8, we sample AGN perturbations from $w_j \sim N\left(0, \Sigma_{sys}\right)$. $\Sigma_{sys}$ is an $\mathfrak{N} \times \mathfrak{N}$ matrix which expresses the uncertainty of the state in the observation space. In our experiments (§ 6) we estimate $\Sigma_{sys}$ using *ad hoc* methods, but it could also be determined by passing sigma points through the forward kinematics (Brookshire & Teller, 2012). Again, using a Taylor approximation:

$$\underbrace{\mathcal{X}_j}_{M\times 1} = \underbrace{m_k}_{M\times 1} \oplus \underbrace{J_m^\dagger}_{\mathfrak{M}\times\mathfrak{N}} \underbrace{w_j}_{\mathfrak{N}\times 1} \tag{19}$$

The equation uses the pseudo-inverse of the Jacobian, $J_m$, evaluated at $m_k$, to convert uncertainty in the observation space to uncertainty in the state space. Then $w_j$ is simply projected onto $\mathcal{M}$ to produce the points $\mathcal{X}_j$.

The conditions under which $J_m$ is singular must again be considered. Either (1) $J_m$ is singular because the same dimensions are always in the null space or (2) $J_m$ is singular due to a specific configuration. Since the system is observable, (1) is not the case. If (2) is the case, then there exists a non-trivial null space and $\mathcal{X}_j$ will equal $m_k$ along the null dimensions of $J_m$. This is a problem because no noise has been added along the null dimensions. In other words, $w_j$ does not define a unique change in $m_k$ and the pseudo-inverse extinguishes perturbations along those dimensions.

To see this, consider again the four-link chain shown in Fig. 9a with all links aligned along the horizontal axis. If a bearing-only sensor is positioned at $(-1, 0)$ looking along the horizontal axis, the Jacobian is degenerate; the entire linkage can slide along the horizontal axis without changing the bearing observations. Using Eq. 19, the particles shown in Fig. 9b are generated. Note that along the (horizontal) null dimension, there is no perturbation in the particles for the leftmost link. Increasingly to the right, there is some displacement in the horizontal dimension due only to motion in the constraint manifold. Thus, Eq. 19 does not meaningfully distribute the particles, which is undesirable.

The solution is to use the state space to propose points in the null space of $J_m$. Revising Eq. 19:

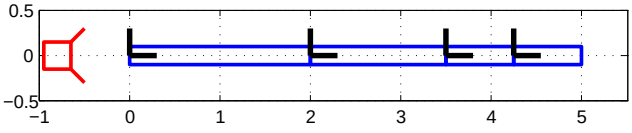
$$\mathcal{X}_j = m_k \oplus \left( J_m^\dagger \, w_j + \underbrace{\mathcal{N}\left(J_m\right)}_{\mathfrak{M}\times\mathfrak{M}} \underbrace{v_j}_{\mathfrak{M}\times 1} \right) \tag{20}$$

where $v_j \sim N(0, \Sigma_x)$ and $\Sigma_x$ is the $\mathfrak{M} \times \mathfrak{M}$ covariance matrix for the state. The $\mathcal{N}\left(J_m\right) = I - J_m^\dagger \, J_m$ matrix is the null space projector matrix. When using Eq. 20, particles are well distributed even when singularities exist in the Jacobian (see Fig. 9c). Notice that because we make use of the state space in these situations, our method is not entirely independent of the state parametrization. However, it is used sparingly, relative to other methods.

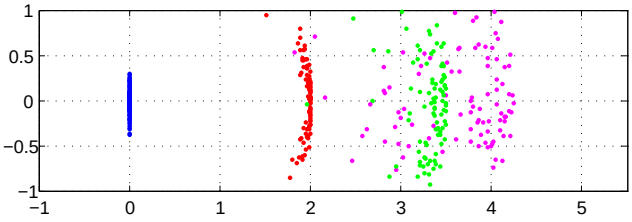
Our method draws samples from the high-dimensional observation space and projects them onto a (typically, lower-dimensional) plane tangent to the configuration manifold. The samples are then transferred onto the manifold itself according to the Taylor approximation. An alternative could be to sample directly on the tangent plane, for example from a distribution suitably constructed from the sigma points (Julier, 2002) of $\Sigma_{sys}$ (Fig. 8).

### *4.4. Rejection Sampling*

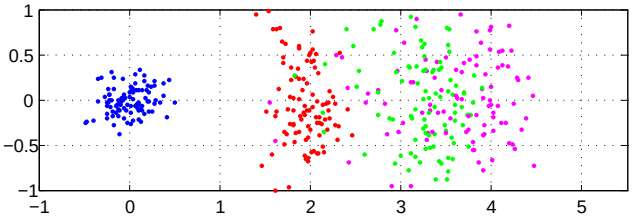
Although the samples produced by Eq. 20 are guaranteed to satisfy the kinematic equations (i.e., lie in the manifold), they may not satisfy the joint limits. As a result, Function 4.1 iterates Eq. 20 in line 9-line 16, generating a sample and evaluating it against the joint limits. If the particle satisfies the joint limits, it is added to $\mathcal{X}^{(i)}$; otherwise, it is discarded. This rejection sampling (Koller & Friedman, 2009) is appropriate only for joint limits which define a volume in the state space. Joint limits defining equality constraints can be incorporated into the kinematic constraints.




(a) A bearing-only sensor at $(-1, 0)$ observes the four-link kinematic chain. In this configuration, the system is singular because the chain can be moved along the horizontal axis without affecting the observations.




(b) Particles proposed with Eq. 19 have little motion along the singular dimension.




(c) Particles proposed with Eq. 20 use the state space to propose only along the singular dimension, reducing effects of the state parametrization.

**Fig. 9.** Proposing particles at singularities

### *4.5. OIF Sampling*

Finally, the OIF is evaluated using Eq. 1 and a discrete version of Eq. 2. Lines 17-22 calculate likelihood of each $\mathcal{X}_j$. For each particle, $\mathcal{X}^{(i)}$ forms a discrete approximation for the OIF distribution $P\left(x_k \left| x_{k-1}^{(i)}, z_k \right.\right)$ and their likelihoods can be summed to approximate $P\left(z_k | x_{k-1}^{(i)}\right)$ (c.f., Eq. 2). Fig. 10d illustrates this — each particle has a discrete approximation to the OIF. Since all $\mathcal{X}_j$'s satisfy the joint limits and kinematic constraints, we can select $x_k^{(i)}$ by drawing from $\mathcal{X}^{(i)}$ according to the OIF probabilities.

### *4.6. Algorithm*

The algorithm is shown in `Proposal3D`, Function 4.1. To calculate the `Jacobian3D` and the forward kinematics, $f$, we relied on the Orocos Kinematics Dynamics Library (KDL) (Bruyninckx et al., 2003). The mechanical description of the object is supplied in a URDF file, parsed, and loaded as a KDL kinematic tree. Using that tree, the forward kinematics and Jacobian can be easily calculated. Since KDL internally uses 6-element twists to represent rotational and translational velocities, which are also members of the Lie algebra of $SE(3)$, we also use them here. As a result, `Jacobian3D` returns a matrix with rows corresponding to twists of observations and columns corresponding to a twist in the base pose and joint velocities. All covariance and precision matrices also use twists to express incremental changes.

## 5. Implementation Notes

We desired an interface such that many different kinds of detectors could provide input without requiring code modification to the algorithm. For example, an image-based detector would provide observations in the form of $\{x, y\}$ pixel locations. An ICP-based fitting algorithm operating on point cloud data might provide a complete 6-DOF pose. Or, an optimization routine based on a planar LIDAR might provide a 3-DOF pose. In principle, the algorithm can be configured to accept any such type of observation, but might require re-coding or re-compiling the code each time.

Instead, we chose to standardize our observation input to always be 6-DOF poses. To allow for detectors which would not provide complete information, we associated each pose with a precision matrix. Importantly, we allowed this precision matrix to be singular to represent unobserved DOFs. For example, an image-based detector would observe only two DOFs. The remaining four DOFs would be unobserved and singular in the associated precision matrix. Such an example is shown in Fig. 11. A TLD-based detector (Kalal et al., 2010) tracks six features in video of an excavator (Fig. 1a). Each of these detections defines a ray in 3D space (Fig. 1b). The precision matrix is then defined to be singular along that ray. Fig. 11 shows example poses drawn using the resulting precision matrix. Any pose along the ray, regardless of orientation, is considered to explain the pixel detection.

Internally, we use a precision matrix to represent uncertainty rather than its inverse, the more common covariance matrix. A covariance matrix cannot easily encode an infinite covariance, but a precision matrix can encode a zero (singular) precision. In § 4, we used $\Sigma^{-1}$ to mean the precision matrix, with the understanding that explicit inversion was not performed.

Although this encoding for the observations has advantages, it also imposes limitations. First, this encoding may require additional computation. Rather than computing two DOFs for an image detection, we are now processing six DOFs (four of which are zero). Additionally, we now maintain a $6 \times 6$ precision matrix for each observation, rather than a $2 \times 2$. Second, the technique is limited to observations which can be expressed with Gaussian uncertainty on a 6-DOF pose. (This encoding would be insufficient to represent a multi-modal uncertainty.)

**Function 4.1:** Proposal3D($x_{k-1}$, $w_{k-1}$, $R_k$, $z_k$, $\Sigma_{z_k}$, $Q$)

**inputs** : $x_{k-1}$, $w_{k-1}$ previous $N_s$ particles and weights
$R_k$ state covariance $\mathfrak{M} \times \mathfrak{M}$ matrix in state space
$z_k$ the $N \times 1$ observations at time $k$, $z_k \in \mathbb{H}^{N/8}$
$\Sigma_{z_k}^{-1}$ precision matrix for the $z_k$ observations
$Q$ observation selection $\mathfrak{N} \times \mathfrak{N}$ matrix specifying valid observations
**params**: $f$ forward kinematic/observation function
$P$ number of discrete elements used to approximate OIF
$\Sigma_{sys}$ state covariance $\mathfrak{N} \times \mathfrak{N}$ matrix in observation space
**outputs**: $\{x_k, \alpha\}$ updated $N_s$ particles and weight scales

$[U, S, V] = \texttt{svd}\left(\Sigma_{z_k}^{-1}\right)$ // NB: $U = V^T$
$L \leftarrow \sqrt{S}V^T$ // $L^T L = \Sigma_{z_k}^{-1}$; cannot use Cholesky because may be singular
$x_k \leftarrow \emptyset$; $\alpha \leftarrow \emptyset$
**for** $x_{k-1}^{(i)} \in x_{k-1}$ **do**
  $J_x \leftarrow \texttt{Jacobian3D}\left(f, x_{k-1}^{(i)}\right)$
  $m_k \leftarrow x_{k-1}^{(i)} \oplus (LQJ_x)^\dagger \, LQ \left(z_k \boxminus f\left(x_{k-1}^{(i)}\right)\right)$
  // create $N_s$ samples via rejection sampling where
  // $\mathcal{X}^{(i)}$ is discretely distributed according to $P(x|x_{k-1}^{(i)}, z_k)$
  $J_m \leftarrow \texttt{Jacobian3D}\left(f, m_k\right)$
  $\mathcal{N}\left(J_m\right) \leftarrow I - J_m^\dagger \, J_m$ // null space projector matrix
  $p \leftarrow P$; $\mathcal{X}^{(i)} \leftarrow \emptyset$
  **while** $p > 0$ **do**
    $w_j \sim N\left(0, \Sigma_{sys}\right)$ // propose state noise in observation space
    $v_j \sim N(0, R_k)$ // propose state noise in state space
    $\mathcal{X}_j \leftarrow m_k \oplus \left(J_m^\dagger \, w_j + \mathcal{N}\left(J_m\right) v_j\right)$
    **if** $\texttt{JointLimitsSatisified}(\mathcal{X}_j)$ **then**
      $\mathcal{X}^{(i)} \leftarrow \{\mathcal{X}^{(i)}, \mathcal{X}_j\}$
      $p \leftarrow p - 1$
  // determine OIF probability of each candidate particle in $\mathcal{X}^{(i)}$
  **for** $\mathcal{X}_j \in \mathcal{X}^{(i)}$ **do**
    $P\left(z_k|\mathcal{X}_j\right) \leftarrow N\left(z_k; f\left(\mathcal{X}_j\right), \Sigma_{z_k}\right)$
    $P\left(\mathcal{X}_j|x_{k-1}^{(i)}\right) \leftarrow N\left(\mathcal{X}_j; x_{k-1}^{(i)}, \Sigma_{sys}\right)$
  $P\left(z_k|x_{k-1}^{(i)}\right) \leftarrow \sum_{\mathcal{X}_j \in \mathcal{X}^{(i)}} P\left(z_k|\mathcal{X}_j\right) \cdot P\left(\mathcal{X}_j|x_{k-1}^{(i)}\right)$
  **for** $\mathcal{X}_j \in \mathcal{X}^{(i)}$ **do**
    $P\left(\mathcal{X}_j \left| x_{k-1}^{(i)}, z_k\right.\right) \leftarrow \frac{P\left(z_k|\mathcal{X}_j\right) \cdot P\left(\mathcal{X}_j|x_{k-1}^{(i)}\right)}{P\left(z_k|x_{k-1}^{(i)}\right)}$
  // sample from $\mathcal{X}^{(i)}$ according to OIF probabilities
  $x_k \leftarrow \{x_k, \texttt{RandomSample}\left(\mathcal{X}^{(i)}, P(\mathcal{X}^{(i)}|x_{k-1}^{(i)}, z_k)\right)\}$
  $\alpha \leftarrow \{\alpha, P\left(z_k|x_{k-1}^{(i)}\right)\}$ // append weight to $\alpha$
**return** $\{x_k, \alpha\}$

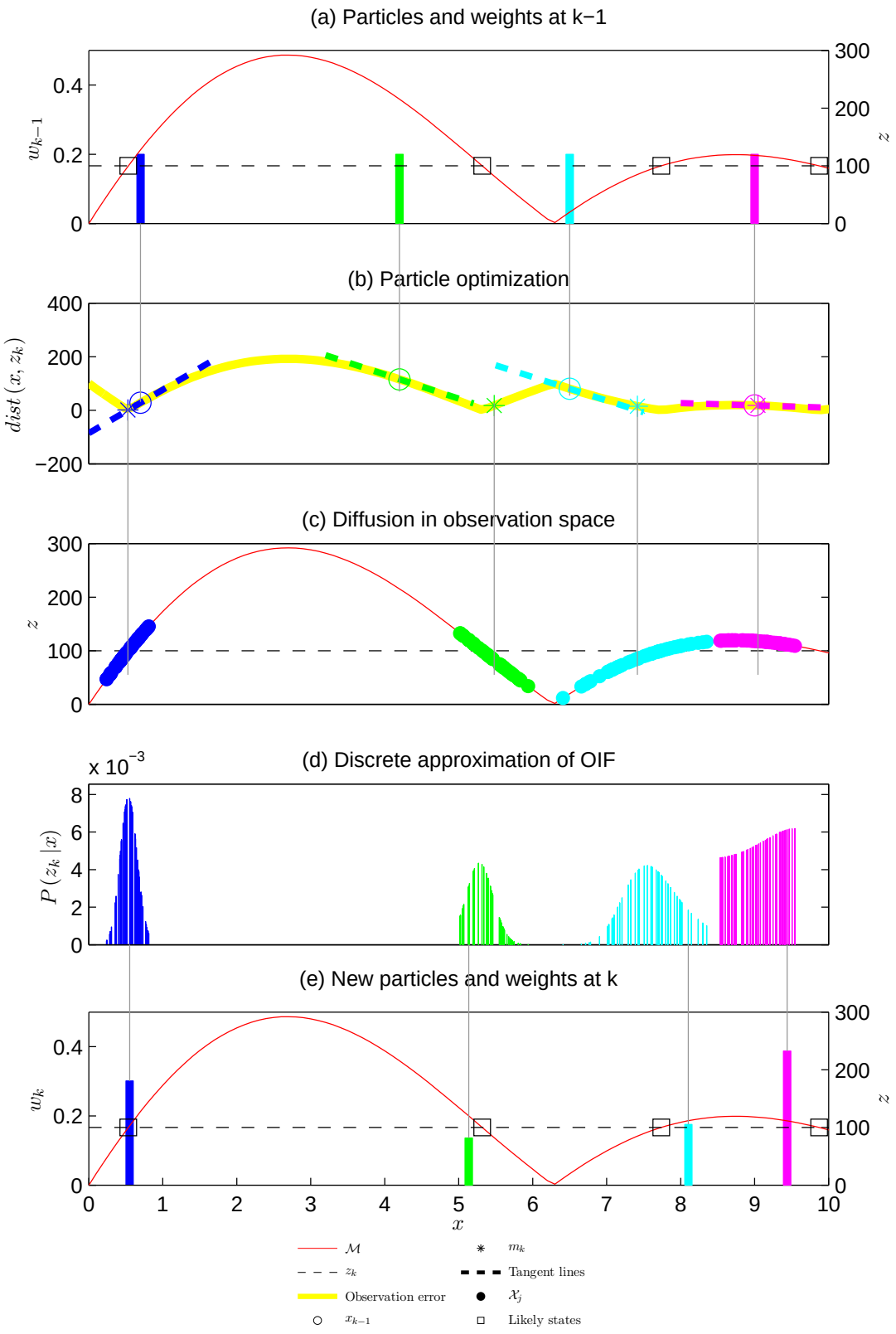



**Fig. 10.** An observation (dotted line) is obtained in (a); intersections with $\mathcal{M}$ are likely configurations (black squares). The particles ($x_{k-1}$) are then optimized (b) toward the likely configurations ($m_k$, color asterisks). Random perturbations are added in the observation space (c). For each particle, $\mathcal{X}^{(i)}$ in (d) approximates the OIF. $\mathcal{X}^{(i)}$ is then sampled to select the next generation of particles (e).

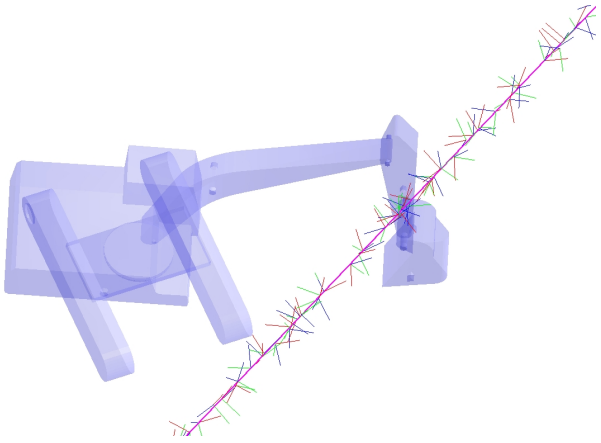
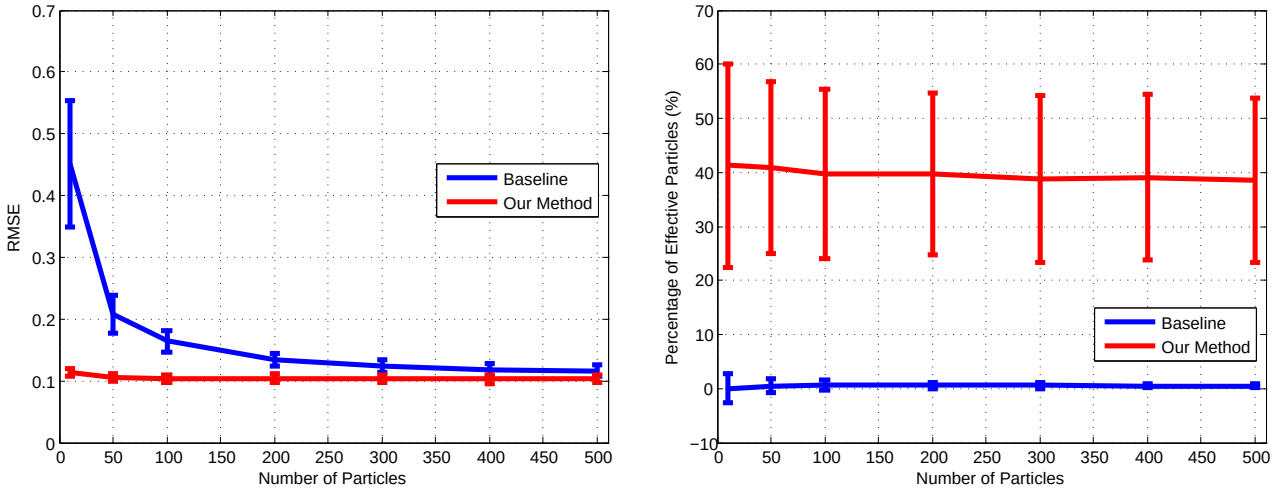


**Fig. 11.** A precision matrix is defined such that samples drawn from the associated Gaussian distribution lie along the ray. Here, we show samples for the detection on the excavator's stick (the ray). The "barbed" lines along the ray are axes corresponding to sampled poses.




(a) RMSE

(b) Number of effective particles

**Fig. 12.** Results for the kinematic chain. Error bars show one standard deviation about the mean.

## 6. Experiments

### *6.1. Planar Simulation*

We compared the baseline approach and our method on two different examples in simulation: a four-link kinematic chain with pose observations and a dishwasher with a door and two drawers. For each system, we (1) varied the number of particles, (2) performed 100 Monte Carlo simulations, and (3) evaluated the root mean squared error (RMSE) of the tracker.

*Four-link Kinematic Chain with Pose Observations* The kinematic chain in Fig. 4 was simulated with a state vector consisting of the $SE(2)$ pose of the leftmost link and each of the joint values (c.f. Eq. 5). The observation vector consisted of an $SE(2)$ pose for each of the links (c.f. Eq. 6).

As shown in Fig. 12a, our method achieves the same tracking accuracy (RMSE) with 10 particles as does the baseline method with ~500 particles. As seen in Fig. 12b, our method also demonstrated a higher average $N_{\text{eff}}$ indicating more high probability particles.

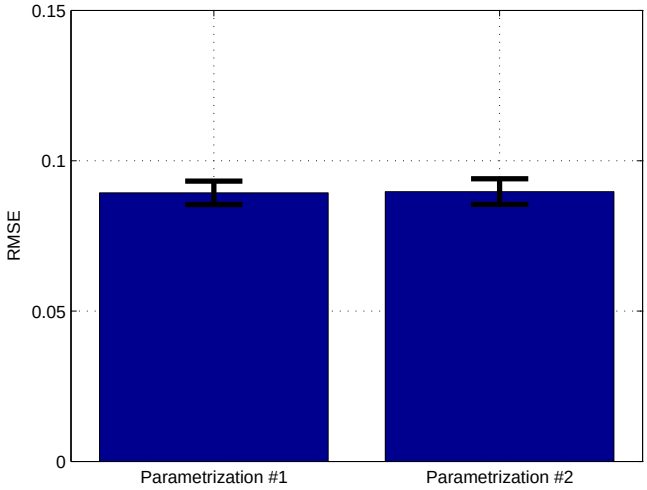



**Fig. 13.** Different state parametrizations do not significantly affect the RMSE for the four-link kinematic chain. The error bars indicate one standard deviation. This was not the case for the UKF (see Fig. 6).

*Parametrizations and the Four-link Kinematic Chain* In § 3.3, we examined two different parametrizations of the four-link kinematic chain and demonstrated that the choice of parametrization affects the RMSE for the UKF (see Fig. 6c). We performed the same experiment, using the parametrizations in Eq. 7 and Eq. 8 for our method. As shown in Fig. 13, the RMSE is nearly the same for both state parametrizations using our method. Our method will generally exhibit reduced dependence on the state parametrization. However, it is not strictly independent of the state parametrization. Dependencies may arise when degeneracies exist in the observation space and the method begins to propose from the state transition model.

*Dishwasher* A dishwasher is a simple articulated object which a household robot is likely to encounter (see Fig. 14a). The state for the dishwasher consists of a pose $p \in SE(2)$ for the dishwasher's basin and joint values for its doors and drawers:

$$x = \begin{bmatrix} p & \theta_{door} & x_{upper} & x_{lower} \end{bmatrix}^T \tag{21}$$

The observations consist of poses for the dishwasher door and each of the two drawers, $z \in SE(2)^3$:

$$z = \begin{bmatrix} p_{door} & p_{upper} & p_{lower} \end{bmatrix}^T \tag{22}$$

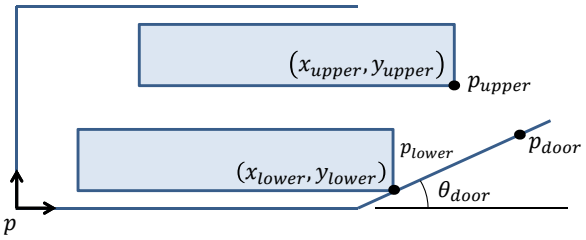
Unlike the previous example, the dishwasher joints are subject to joint limits. The drawers have limited travel in the horizontal direction and the door cannot close into the drawers. These constraints are satisfied via rejection sampling.

$$0 \leq \theta_{door} \leq \min\left(\operatorname{atan}\left(\frac{y_{upper}}{x_{upper}}\right), \operatorname{atan}\left(\frac{y_{lower}}{x_{lower}}\right)\right) \tag{23}$$

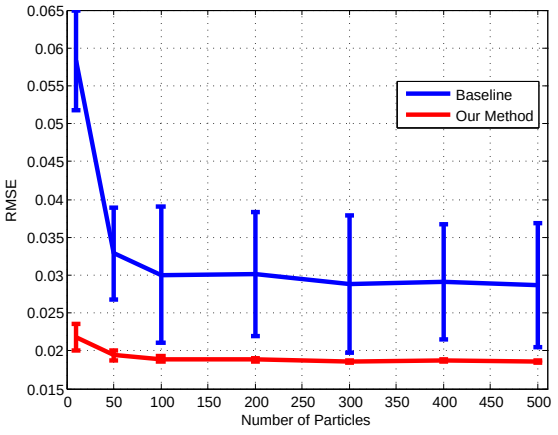
$$0 \leq x_{upper} \leq x_{max} \tag{24}$$

$$0 \leq x_{lower} \leq x_{max} \tag{25}$$

The RMSE for the dishwasher, using our method (Fig. 14b), was relatively constant at ~0.02 for 10-500 particles. The baseline method did not achieve that performance with fewer than ~500 particles. Additionally, our method averaged 40-60% effective particles, compared to less than 5% with the baseline method.




(a) The dishwasher consists of two articulated drawers and one door. Joint limits prevent configurations in which the door and drawers would overlap or extend beyond physical limits. $y_{upper}$ and $y_{lower}$ are static parts of the kinematic model.




(b) Results for the dishwasher simulation are shown. Error bars show one standard deviation about the mean.

**Fig. 14.** Simulated dishwasher

*Comparison with UKF* As mentioned in § 3.3, the UKF exhibits a state parametrization dependence. In this example, we demonstrate that, when the articulated object's motion is favorable to the parametrization, both our method and the UKF perform similarly. However, when the encoding does not correspond to the motion well (i.e., such that the UKF's sigma points do not cover likely configurations well), our method has lower RMSE than the UKF.

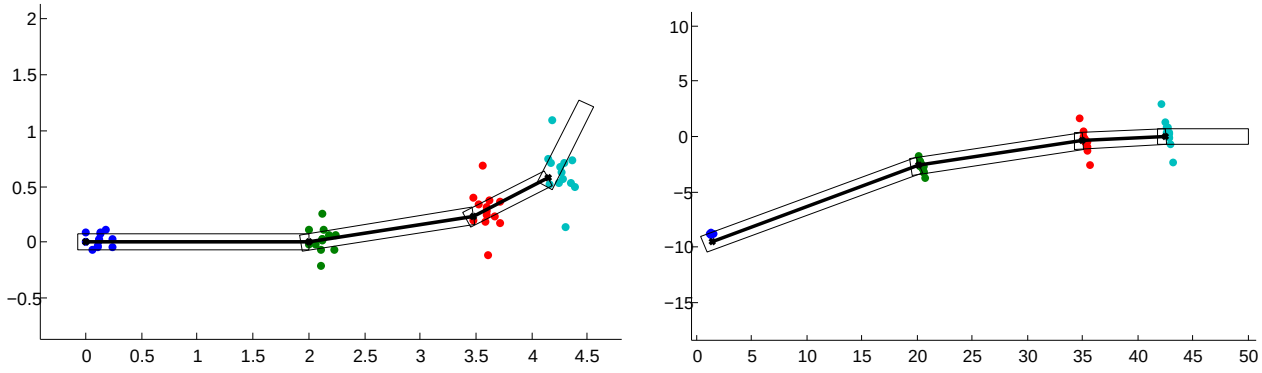
Fig. 16 show sample frames from two simulations for two different kinematic chains. Both simulations use the parametrization of Eq. 7, including the base pose of the left-most link and the three joint angles. Because both simulations use the same parametrization, both propose similar sigma points (see Fig. 15). In particular, note that sigma points near the base pose (left) are tightly grouped, while sigma points near the right-most link are spaced further apart. Again, this difference in spacing is due to the non-linear nature of the parametrization. In other words, this difference results because the UKF uses Euclidean addition in the state space to add to the mean state.

If a state transition model is particularly good, then the small differences in sigma points may be negligible. However, our work does not assume that a good state transition model is available. For example, in Simulation #1, the left-most link (base pose) remains fixed and the right-most link moves significantly; the sigma points cover the motion well. On the other hand, for Simulation #2, the left-most link moves significantly and the right-most link remains fixed; the sigma points do not cover the motion well.

As seen in Fig. 17a for Simulation #1, the UKF and our method perform similarly well. However, in 17b for Simulation #2, our method exhibits lower RMSE than the UKF. This is because our method is less dependent on the state parametrization. In situations where a good state transition model was available or the "best" parametrization could be guaranteed, the UKF might be a reasonable alternative.

## 6.2. Dishwasher

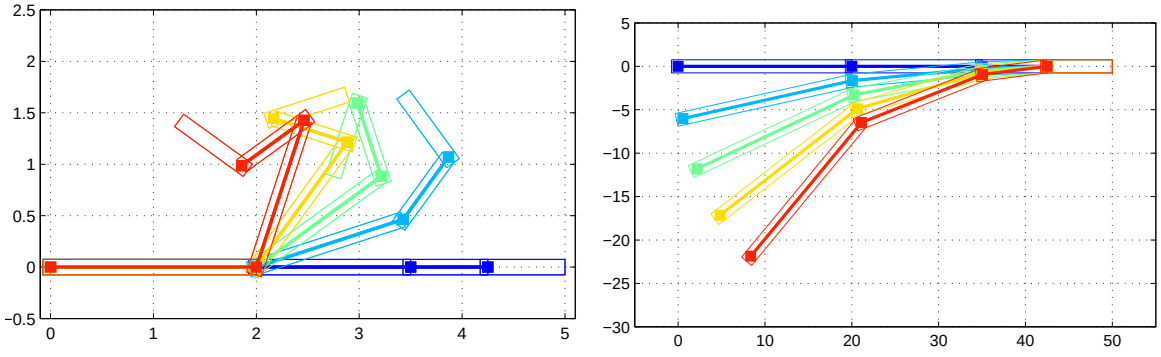
Approximately 1300 frames of RGB-D data were collected of a dishwasher (see Fig. 18) being opened and closed. Ground truth and the kinematic model were established via manual annotation. Three independent TLD (Kalal et al., 2010) trackers were manually initialized on the door and drawers on the first frame they came into view. As a result, there are many frames with missing observations. For example, in the first frame, the drawers are not tracked because they are obscured by the door. Some observations are also missing as the TLD tracker occasionally lost track. Nonetheless, our algorithm is still able to use these partial observations during particle proposal, c.f. Eq. 18.



(a) Sample sigma points for Simulation #1

(b) Sample sigma points for Simulation #2

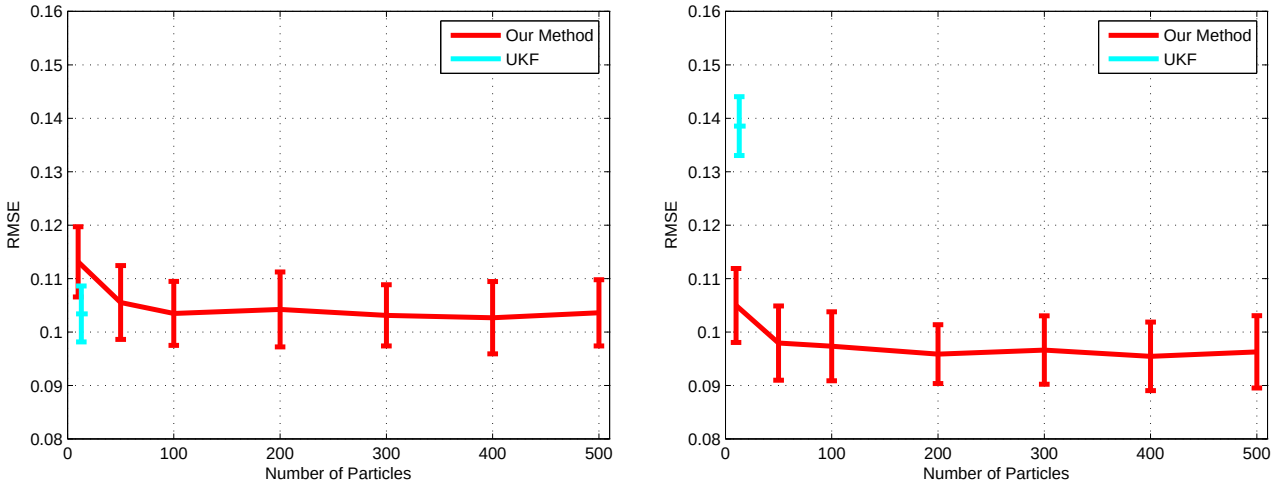
**Fig. 15.** Sigma points for UKF simulations.



(a) Simulation #1

(b) Simulation #2

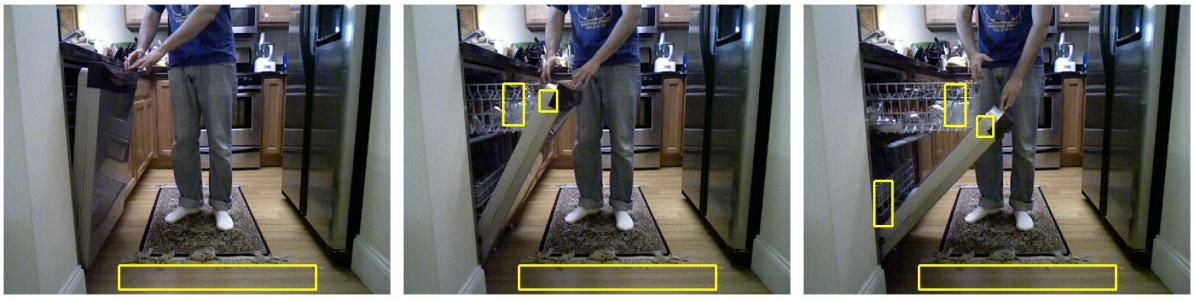
**Fig. 16.** The same parametrization is used for two different simulations, resulting in different UKF performance, relative to our method (see Fig. 17). The kinematic chains shown here are simulated to move through the counter-clockwise poses.




(a) RMSE for Simulation #1

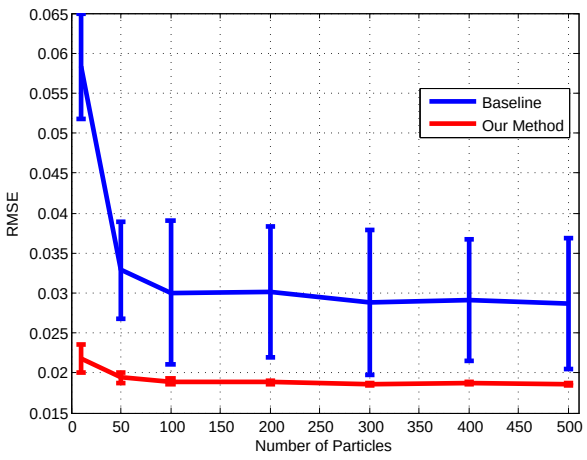
(b) RMSE for Simulation #2

**Fig. 17.** The RMSE of our method and the UKF are similar for Simulation #1, where the parametrization produces favorable sigma points. This is not always the case, however, as illustrated by Simulation #2. The x-axis location of the UKF results corresponds to the number of sigma points. Error bars show one standard deviation about the mean.

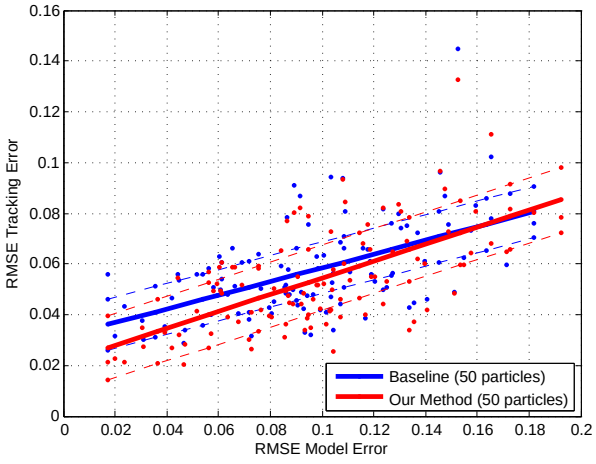


(a) (b) (c)

**Fig. 18.** A TLD tracker provided positions of the dishwasher's articulated links as input. A vertical was also extracted.




(a) In addition to lower RMSE, our method demonstrated less variation in accuracy while tracking the dishwasher, because it quickly recovered when missing observations resumed. Error bars show one standard deviation about the mean.




(b) The baseline and our method are affected similarly by model noise. Dotted lines show one standard deviation.

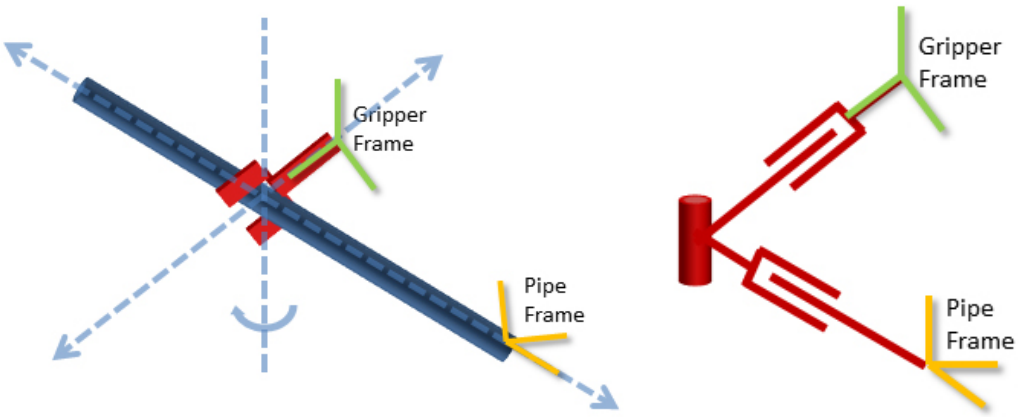
**Fig. 19.** Dishwasher results.

Unlike the simulation (which uses poses), these observations consist of the positions of the door and drawers and a vertical estimated from a static patch on the floor. Although the vertical is not necessary for observability, we found both the baseline and our method benefited from the additional information.

Since only the positions of the dishwasher door and drawer are available, a singularity exists when the door is closed (vertical). In this configuration, observation movement in the horizontal direction can be explained both by movement of the base pose or by a slight opening of the door and movement of the drawers. The null space term in Eq. 20 is crucial so that meaningful particles are proposed during the initial frames.

Fig. 19a shows results for the dishwasher. In addition to higher accuracy, our method also exhibits substantially less variation. This is primarily due to periods when the observations were insufficient to make the Jacobian full rank (which did not occur in the other examples). Both methods would "wander" during these singularities, each proposing unconstrained particles in the null space. However, when observations became available again, our method was able to quickly recover and begin proposing particles near those observations. Finally, our method maintained ~50% effective particles, while the baseline had only about 5%.

We also consider the effect of model noise on performance. As mentioned, we anticipate that most models will be imperfect. We simulated uniform noise of $\pm 0.05\mathrm{m}$ on the dishwasher housing width, height, and the vertical locations of the two drawers. A comparison of the model error to RMSE tracking performance is shown in Fig. 19b with best fit lines.




(a) (b)

**Fig. 20.** The PR2's grip on the pipe is not rigid, but still constrains some movement. The system can be modeled as a 3-DOF kinematic chain.

The lines have similar slopes, indicating that, for the dishwasher example, both the baseline approach and our method behave comparably under increasing model noise. (We also simulated Gaussian noise with a standard deviation of $0.05\mathrm{m}$ on the same parameters and achieved similar results.)

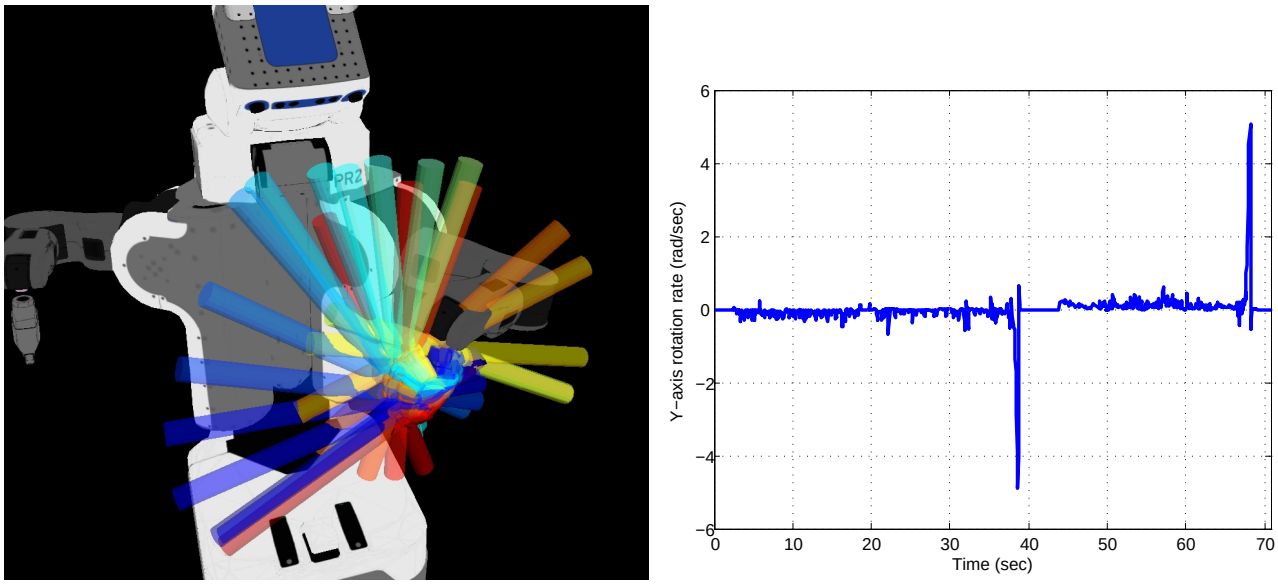
### *6.3. PR2*

In this experiment, we consider a PR2 robot holding a 60cm PVC pipe, and the goal is to estimate the pose of the pipe. This task is interesting because PR2 cannot grip the pipe rigidly due to its weight and length. As a result, the pipe tends to slip in the gripper; however, this slip occurs only along certain directions. As suggested in Fig. 20, the pipe may translate in the gripper along the pipe's long axis or along the grip direction. The pipe can also rotate about the suggested axis.

We might wish to know the pose of the pipe so that it can be accurately manipulated — when, for example, inserting into a mating coupling. An obvious solution might be to track the pipe in RGB-D data using a RANSAC cylinder fit. A complication with this approach, however, is that the absolute pose of the pipe is not observable. Rotation about the pipe's long axis cannot be observed (by any of the PR2's sensors) due to the rotational symmetry. (In practice, because the end of the pipe is small, we also found that the position estimate along the long axis was also inaccurate.) Thus, recovering even a relative pose of the pipe is not possible by observing only the pipe itself.

Fortunately, more information is available. Although the gripper does not hold the pipe rigidly along all six DOFs, it does provide rigid support along three DOFs. Thus, 6-DOF information from the pose of the gripper (as provided by the PR2's telemetry and forward kinematics) and 4-DOF information about the partial pose of the pipe (as provided by a RANSAC cylinder fit on RGB-D data) can be combined to track the pipe in 6-DOF. (Note that we do not track the orientation about the pipe's long axis absolutely; this rotation is tracked relative to the initial orientation.) It is clear that the RANSAC estimate of the pipe's location will have errors; but the pose of the gripper is also subject to errors. Mechanical slop, imperfect PR2 models, and errors in the Kinect calibration all lead to errors in the Kinect-to-gripper transform.

We achieve the pipe tracking by explicitly modeling the DOFs between the gripper and the pipe. As shown in Fig. 20b, the system can be modeled as a 6-DOF pose of the gripper and a kinematic chain with three joints (two prismatic and one rotational). Two sequences were collected with the PR2. The "pipe-swing" scenario in Fig. 21 demonstrates significant pipe rotation in the gripper as the gripper rotates. Fig. 21b shows the rotational velocity; notice two significant spikes in velocity prior to 40 and 70 seconds. These spikes correspond to events when the pipe passed a vertical orientation and




(a) (b)

**Fig. 21.** In this sequence, the PR2 rotates the pipe. (a) shows the pipe poses over time; the pipe proceeds through red-yellow-blue poses (best viewed in color). The two large velocity spikes in (b) correspond to times when the pipe underwent significant slip in the gripper.

slipped in the gripper. During these two events, the pipe moved over 90 degrees in just 2-3 frames, and the state transition model was particularly poor especially during these frames.

In the second "axis-rotate" scenario, the PR2 moved to rotate the pipe primarily along its long axis (see Fig. 22). Fig. 22b shows the rotational velocity as the gripper makes several forward and backward rotations with the pipe. Here, we track the pipe's pose (relative to the start pose); this would not be possible by observing only the pipe. Information from the gripper's pose is combined (via the kinematic chain constraints) to estimate a full 6-DOF pose of the pipe. It is important to note that this ability is a result of the model in Fig. 20b and not of our particular tracking technique.

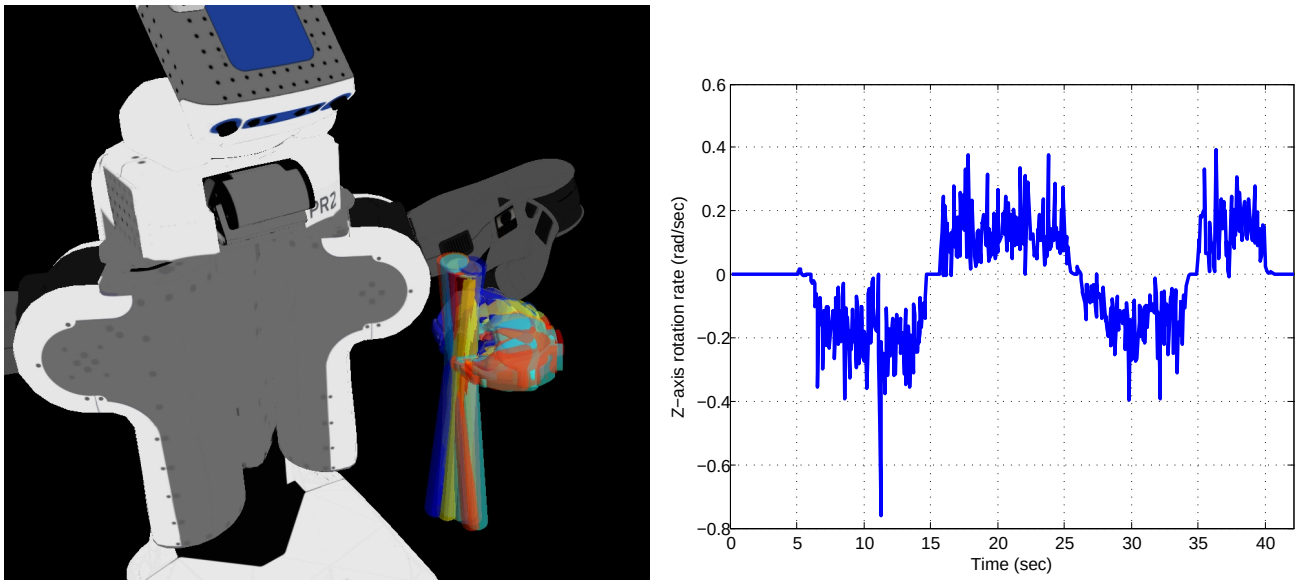
Fig. 23 and Fig. 24 show the RMSE and number of effective particles for the pipe-swing and the axis-rotate experiments, respectively. In both cases, our method achieves error levels at around 20 particles lower than those achieved by the baseline method with fewer than 500 particles. The $N_{\text{eff}}$ is also higher, indicating that our particles are located in more high-probability regions.

We again perturbed the kinematic model by adding uniform random noise of $\pm 0.05\text{m}$ and $\pm 0.05\text{rad}$ to each translational and rotational parameter, respectively. In Fig. 25a, the baseline method's error during tracking dominates the results. In Fig. 25b, both methods experience similar increases in tracking error.

## *6.4. Excavator*

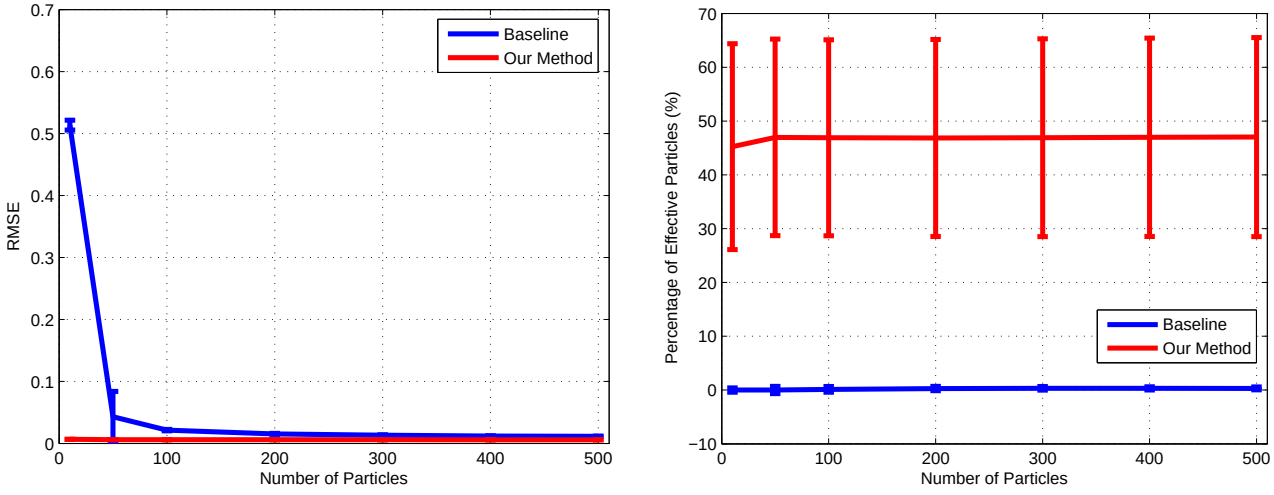
In this experiment, we visited the Heavy Construction Academy (HCA) in Brentwood, NH (see Fig. 26). We observed an excavator operating and collected data with a PointGrey CMLN-13S2C camera and Velodyne HDL-32E 3D LIDAR. (The Velodyne was not used in the following experiments, but did provide a check during ground truth processing.) We also collected video using a Sony HF10 camcorder. Both the PointGrey camera and Sony camcorder were calibrated using a standard checkerboard technique (Bouguet, 2013).

The excavator is a CATERPILLAR 322BL (CAT322BL), and includes a tracked base and rotating operator cab/engine. The heavy lift arm consists of three links, referred to as the boom, the stick, and the bucket (proceeding outward from the cab). We constructed a model of the CAT322BL excavator using third-party datasheets (RITCHIESpecs, 2013). The datasheets were not complete, but provided adequate measurements to make reasonable extrapolations. We do not expect



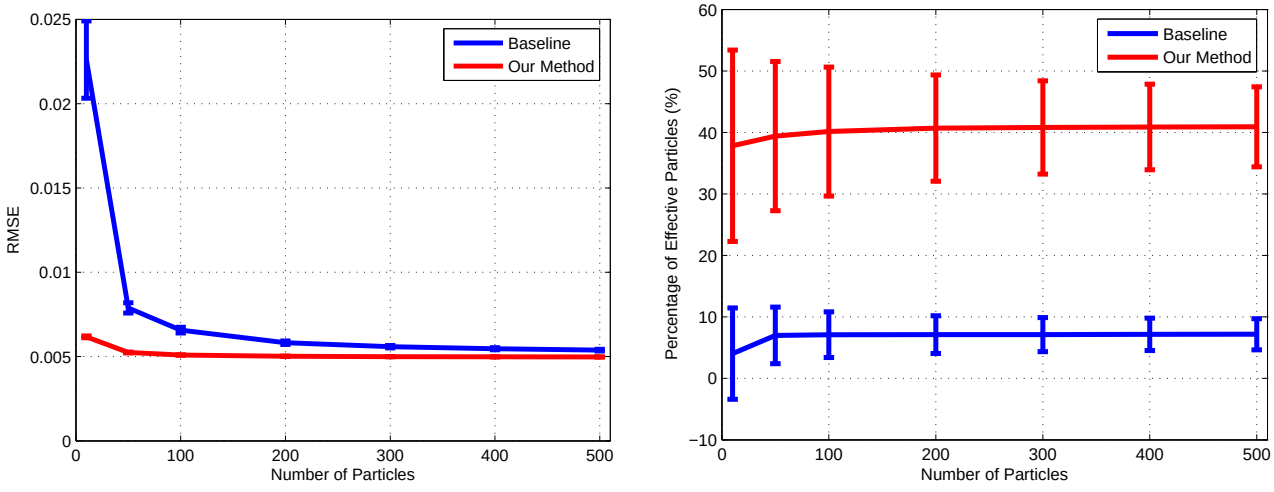

(a) (b)

**Fig. 22.** In this sequence, the PR2 moved its gripper so as to rotate the pipe approximately along its long axis (here, nearly vertical).



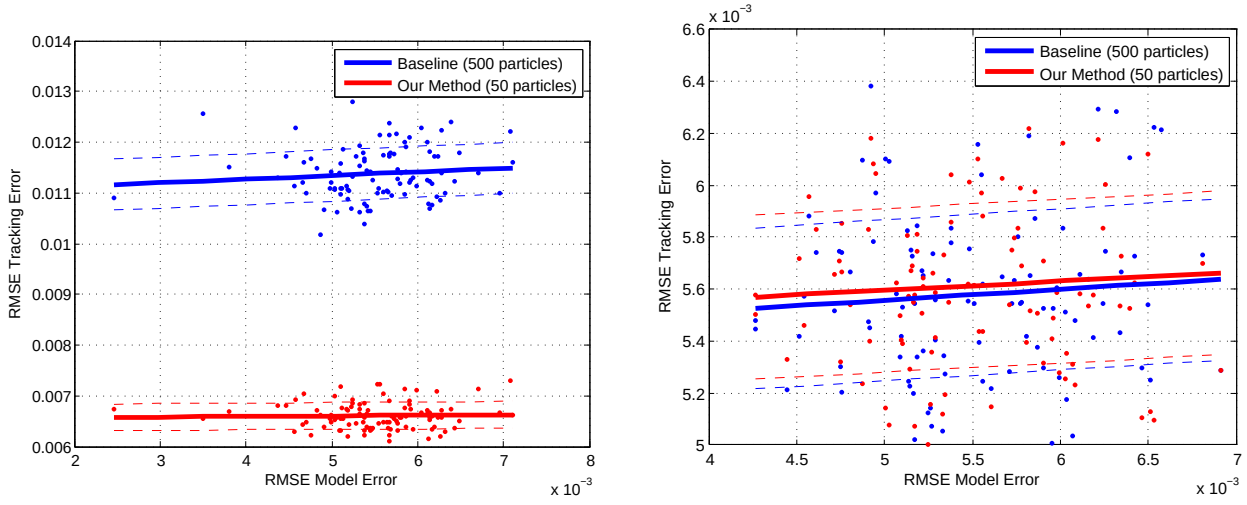

(a) (b)

**Fig. 23.** RMSE and number of effective particle performance for the pipe-swing sequence in Fig. 21 are shown. Error bars show one standard deviation about the mean.




(a) (b)

**Fig. 24.** RMSE and number of effective particle performance for the axis-rotate sequence in Fig. 22 are shown. Error bars show one standard deviation about the mean.




(a) Model noise for the pipe-swing experiment

(b) Model noise for the axis-rotate experiment

**Fig. 25.** PR2 model noise results

**Fig. 26.** We captured operation of an excavator with a PointGrey camera (mounted left) and 3D LIDAR (mounted right).

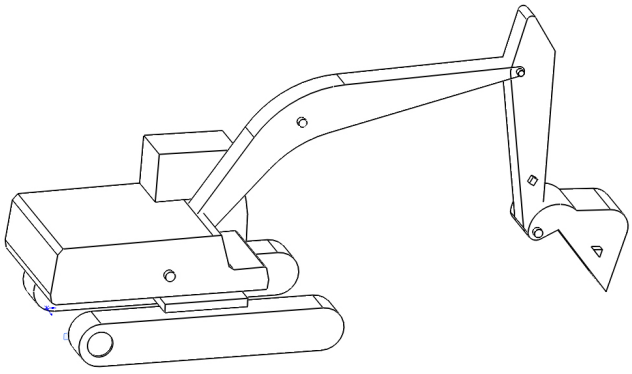


**Fig. 27.** The CATERPILLAR 322BL excavator was modeled in SolidWorks using specifications available online (RITCHIESpecs, 2013).

that our model of the CAT322BL is perfect; but it does demonstrate that even approximate models are sufficient to achieve quality results.

The image-based observations were made using the seven independent TLD (Kalal et al., 2010) trackers. The trackers were manually initialized during the first frame on the points shown in Fig. 1a. The observations included two points on the tracks, two points on the cab (the point on the back of the cab is not visible in this frame), a point on the boom, a point on the stick, and a point on the bucket.

The excavator has 10-DOFs: six DOFs for the base pose and four joints (track-cab joint, cab-boom joint, boom-stick joint, and stick-bucket joint). The seven observations create 14 constraints, and it can be shown that these constraints are sufficient to observe the 10-DOF excavator configuration. As described in § 5, the pixel locations for the observations are encoded as 6-DOF poses with singular precision matrices.

*Results for Dump Truck Loading* The excavator loads a dump truck with three bucket fulls, repeatedly moving from roughly parallel to the camera's image plane to facing away from the camera. Ground truth was established by manually entering key-frames. Between key-frames, DQ SLERP (Brookshire & Teller, 2012) was used to interpolate the base pose and linear interpolation was used for the joint values. The location of key-frames was established by aligning against both the camera images and the 3D LIDAR data. The ground truth is shown in Fig. 28.

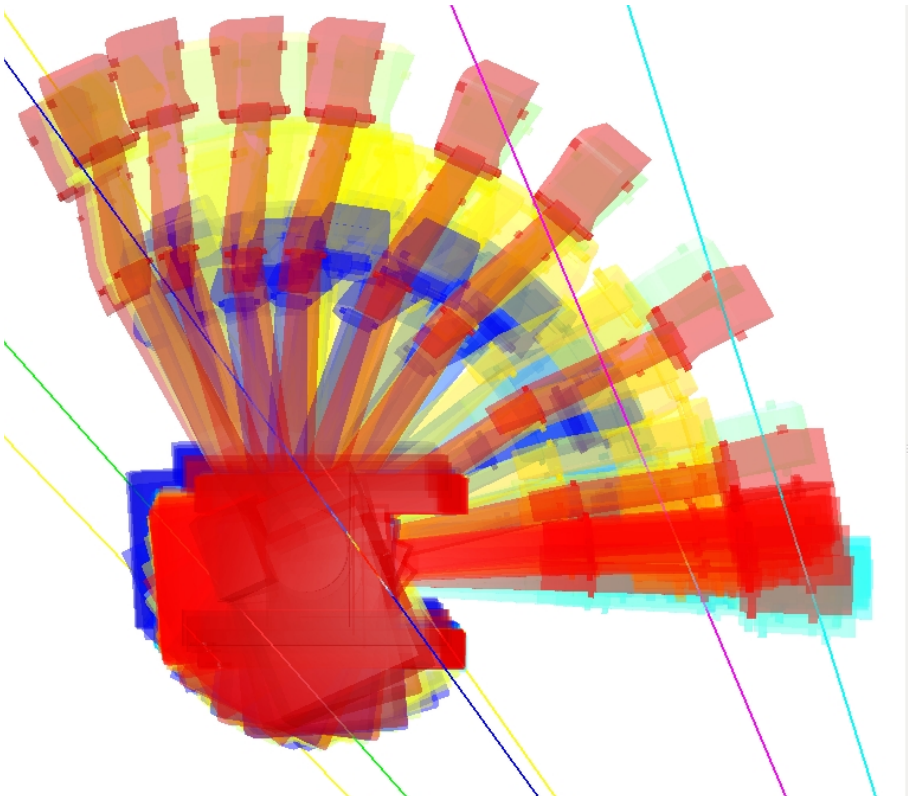


**Fig. 28.** The excavator loads a dump truck three times. Viewed from above, this graphic shows the entire 1645-frame sequence. The three loads correspond to blue, yellow, and red, in that order (best viewed in color).

The plots in Fig. 29 show the RMSE and the effective number of particles of our method versus several alternatives. We evaluate our method against a baseline PF and UKF. We also experimented with a different state transition model: our method uses a zero-velocity model, as do the baseline and UKF methods. However, to give the baseline and UKF methods a possible benefit, we also evaluate them with a constant-velocity state transition model.

1. Baseline. This is the standard particle filter, as used in previous comparisons, which proposes from the state transition model.
2. Baseline with constant velocity. This method combines the baseline particle filter with a constant velocity state transition model. The excavator's state is expanded to include velocity components for all DOFs, creating a 20 DOF state vector.
3. UKF. The UKF (see § 3.3) maintains a single hypothesis for the 10-DOF configuration of the excavator. For a 10-DOF system, the UKF creates 21 sigma points which are somewhat analogous to particles in a PF; we plot UKF results with a "Number of Particles" (x-axis) equal to 21.
4. UKF with constant velocity. Similar to the baseline with constant velocity, this method uses a UKF with a state expanded to include velocity terms. The results are plotted with a "Number of Effective" particles of 41, again corresponding to the number of sigma points.

For the particle filters (baseline and our method), the filter is considered to have failed if all particles have zero probability. The baseline methods were simulated with 400-2000 particles (using fewer than 400 particles resulted in failure every time). Fig. 29a and 29b show the RMSE results, at different scales. In the following, we discuss a few interesting aspects to these graphs, and a video comparing the tracking methods is available in Extension 1.

First, our method performs better than the UKF and UKF with constant velocity methods. As noted in § 6.1 where we considered the UKF in simulation, the UKF and our method can achieve the same RMSE depending on the circumstance.

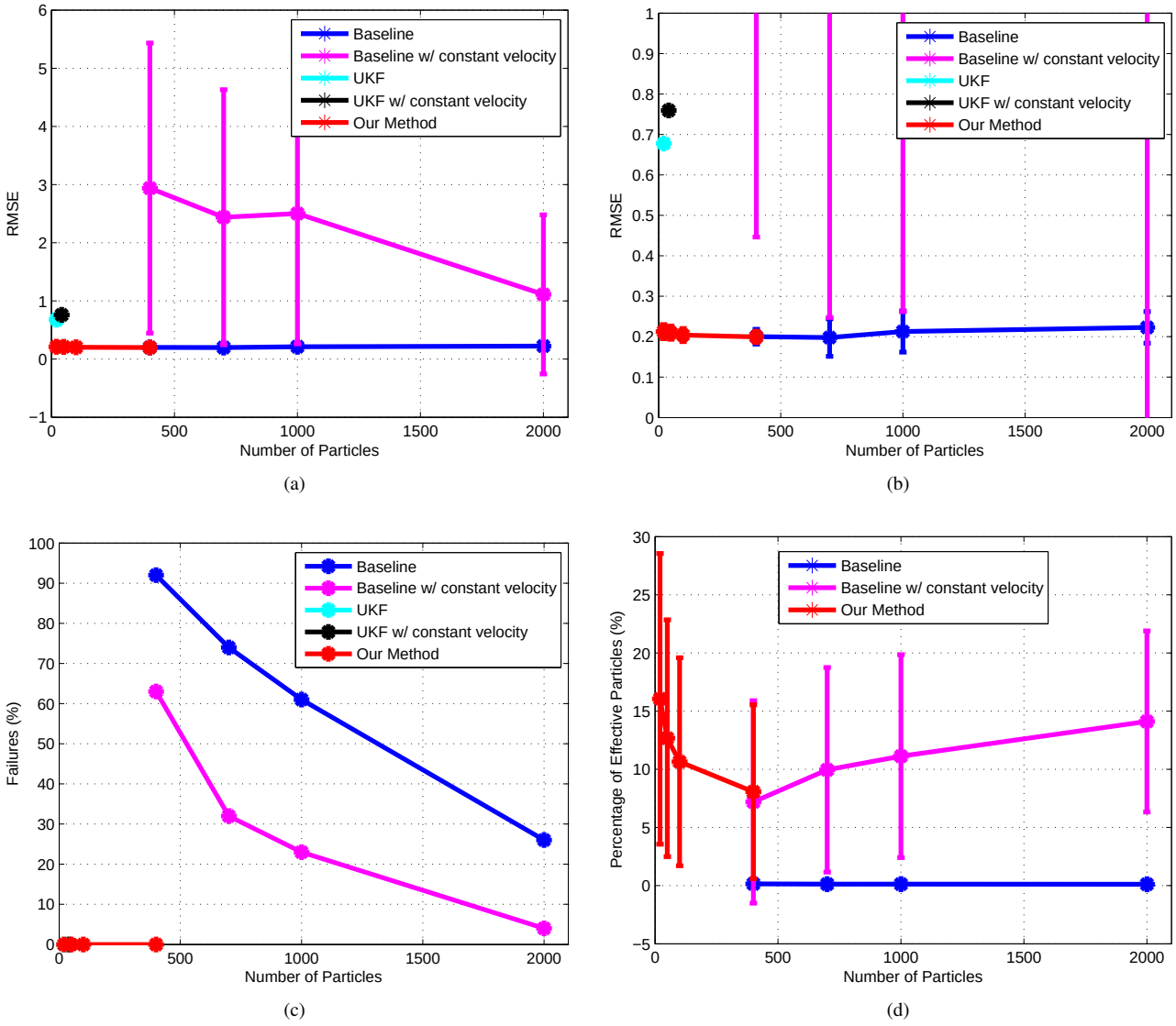



**Fig. 29.** These plots show errors, failures, and number of effective particles for the excavator example. The top row shows the sample RMSE plot at different zoom scales. Error bars show one standard deviation about the mean.

However, for the excavator, the multi-modal nature of the distribution of configurations (see Fig. 3 for examples) impairs the UKF. Although the UKF will occasionally track the correct mode, this kind of event happened often enough in the sequence to impair the RMSE performance.

Second, examining only the RMSE plots, it appears that at 400 particles our method exhibits similar RMSE to the baseline method. Simultaneously, the RMSE error of the baseline method seems to increase with the number of particles. These two surprising results can be explained by examining the failures in Fig. 29c. The baseline method exhibits over 90% failures at 400 particles. The handful of successes indicate the filter randomly selects lucky particles and happens to find a good solution. As more particles are added, the probability of finding a good solution increases and the number of failures decreases. Despite the occasional lucky performance for the baseline method, our method achieves a 100% success rate for all number of particles while still maintaining low RMSE.

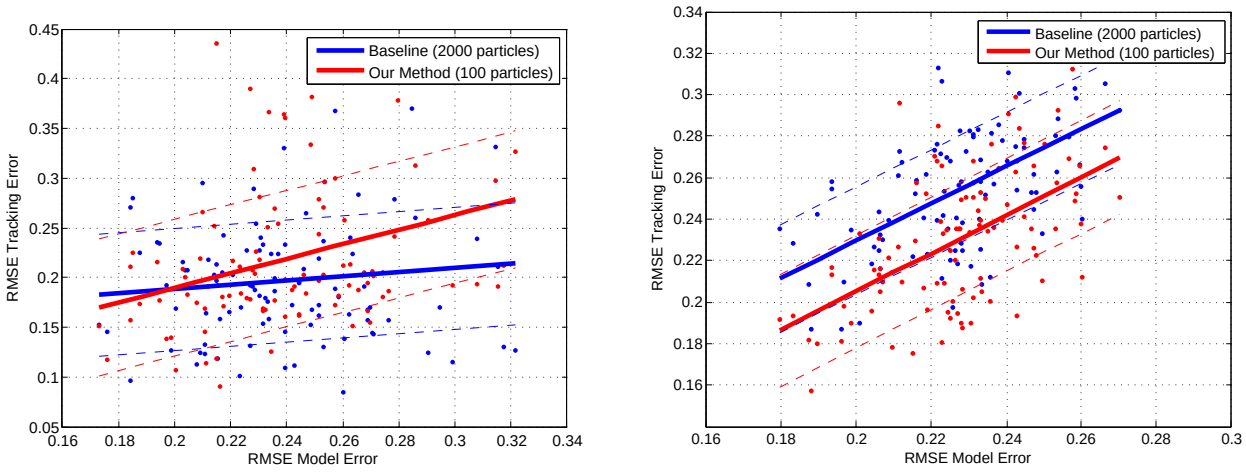
Third, the constant velocity state transition model did not help reduce RMSE for the baseline and UKF filters. Occasionally, a constant velocity model can provide a good first order approximation to motion, but this depends on the smoothness of the motion relative to the frame rate. In this scenario, the motion was sufficiently non-linear (or, alternatively, the frame rate was sufficiently low) that a constant velocity assumption caused increased RMSE. This occurred because the excavator would make sudden stops or starts and the constant velocity model would tend to overshoot the true position. We do not evaluate a constant velocity assumption with our method because it requires the Jacobian of the state. This would have required calculating accelerations for the kinematic tree and is beyond the scope of this work. Regardless, the state model is used sparingly by our method, only along singular dimensions, and would not have significantly affected the results.

Generally, we do not expect the kinematic model to be perfect. (Indeed, the model of the excavator used here is approximated from online datasheets.) Imperfections in the model will cause errors in the forward kinematics and Jacobian calculations. In order to characterize the effects of model noise, we distorted our model of the excavator by adding Gaussian and uniform noise to the kinematic parameters. In particular, we added zero-mean translational and rotational noise, with standard deviations of $0.05\mathrm{m}$ and $0.05\mathrm{rad}$, respectively, to all kinematic parameters. We then compared the tracking error of the baseline PF to our method, as shown in Fig. 30a, for a short segment of the dump truck loading. For the baseline and our method, we used a number of particles where RMSE tracking error was similar, 2000 and 100 particles, respectively. We repeated the experiment with uniform noise of of $\pm 0.05\mathrm{m}$ and $\pm 0.05\mathrm{rad}$ in Fig. 30b.

Imperfections in the excavator model cause the tracking error for our method to increase at a greater rate than for the baseline. This is because our method uses the model for particle proposal and weighting, whereas the baseline method uses it for only weighting. The errors in the model cause non-linear effects: small rotational errors are magnified by the large size (several meters) of the cab, boom, and stick. Because our method depends on the model more, it is more susceptible to these imperfections when larger errors are introduced with Gaussian noise. By contrast, this effect was less noticeable in the model error analysis for the dishwasher (see Fig. 19b) and PR2 (see Fig. 25a). There, the structures were relatively small and did not magnify the errors. Although our method is more sensitive to model errors, it still out performed the baseline method with a reasonable (yet imperfect) model (see Fig. 29) generated from datasheets.

*Results for Climbing* We also collected video of an excavator climbing a hill using a hand-held camcorder. This video series differed from the previous dump truck loading scenario in that there was significant camera and track movement. In this sequence, the operator maneuvered the equipment so that the arm was used to "pull" the excavator up the hill. This maneuver highlights one of the difficulties with selecting a state transition model: most models would have assumed relatively stationary tracks and a more mobile bucket.

Ground truth was not reliable for this sequence, because the absence of 3D LIDAR data made it difficult to manually resolve ambiguities. Nevertheless, our method is able to track the excavator well. Fig. 31 shows three frames (rows) from the sequence for the UKF, baseline, and our method. In each frame, the estimated excavator's pose is projected into the




(a) Gaussian noise

(b) Uniform noise

**Fig. 30.** We simulated errors in the excavator model by adding random noise to all kinematic parameters. Each point represents a different simulation. The solid lines show a best fit; the dotted lines show one standard deviation.

camera frame. When tracked correctly, the excavator should be darkened by the estimated pose. For the first frame of all methods (first column), the bucket is not detected and its position is not able to be estimated.

The UKF again fell into local minimums for the first and second frames. In the first frame, the cab is significantly tilted forward; in the second, the stick/bucket joint is in the wrong position. The baseline method did better at 400 particles, but tended to misalign the tracks. Our method demonstrates good alignment throughout the sequence.

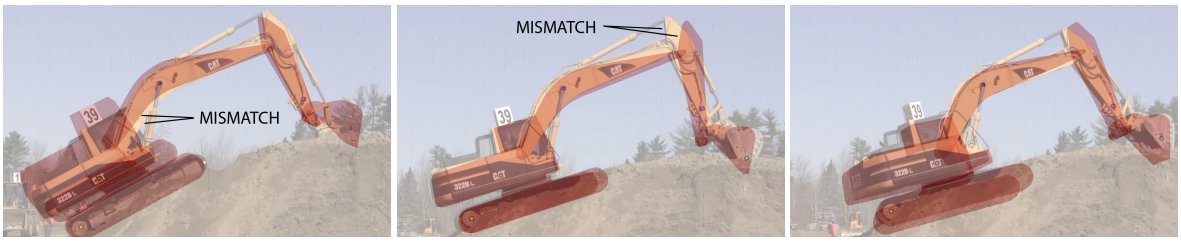
### *6.5. Frame rate*

In this section, we have demonstrated at least an order-of-magnitude reduction in the number of required particles to track at similar levels of RMS error. This has a corresponding increase in our method's frame rate. Fig. 32 compares the frame rate for the Dishwasher, PR2, and Excavator examples. In each case, our method exhibited a factor of $4$ to $8$ speed up over the baseline method. Since the frame rate of the particle filter is linearly proportional to the number of particles, we might have expected a factor of $10$ improvement. This did not happen, however, because of the discrete approximation required to calculate the particle weights in our method.

## 7. Conclusion

Our approach is most appropriate when the object's motion cannot be well predicted (relative to the quality of the observations). When the motion can be well predicted, a UKF or baseline particle filter approach might be a more suitable choice. It is also interesting to consider an increasing number of missing observations. When observations are unavailable, the algorithm relies on the state transition model. As less data is available, the performance converges to that of the baseline method (which always proposes with the state transition model). In other words, when a good state transition model is available and there are missing observations, the algorithm will perform similarly to the baseline particle filter.

It is also interesting to consider articulated structures with axes of symmetry (e.g. "mirrored" objects). The pose distribution of these objects would be similar to the multi-modal ones of the experimental structures considered here. In Brookshire (2013) we explored a multi-modal kinematic chain and found that the particle filter correctly maintained the two modes until the measurements allowed for disambiguation. A symmetrical object would behave similarly, except that multiple modes would always remain.




(a) UKF


(b) Baseline

(c) Our Method

**Fig. 31.** The excavator's estimated position is projected into the camera frame (darkened overlay) for three methods. Each column shows one frame from the series. Ideally, the darkened virtual excavator should just cover the excavator in the image. For example, the middle frame for the UKF shows a mismatch for the stick and bucket.

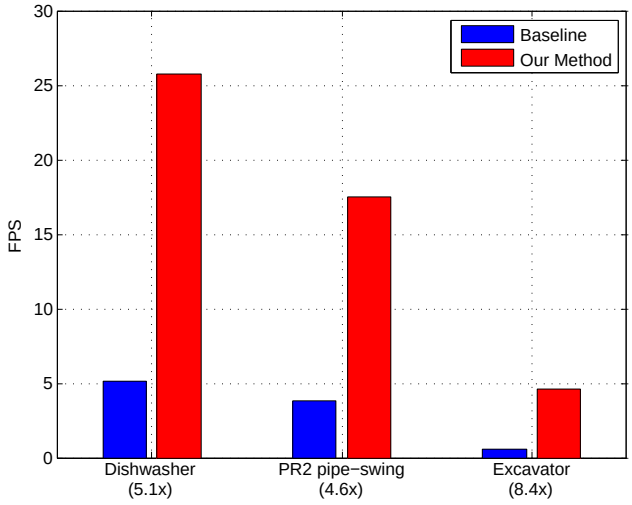



**Fig. 32.** The improved particle generation in our method resulted in a factor of 4-8 speed up over the baseline method. Results were generated on a 3.4 GHz processor running Matlab 2013.

We give an algorithm that augments observations with a model describing the constraints between them. It estimates the joint positions of an articulated object, under assumptions that the configuration manifold is locally planar, relative to the state and observation uncertainty. The algorithm does not rely solely on the state transition model during proposal and, instead, incorporates observations. It proposes noise in observation space, and incorporates an estimator that is independent of state parametrization. We demonstrate more than an order-of-magnitude reduction in the number of particles over a baseline implementation, and a corresponding increase in frame rate.

## Funding

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

### References

D. P. Bertsekas (1999). *Nonlinear Programming*. Athena Scientific, Belmont, MA.

J.-Y. Bouguet (2013). 'Camera Calibration Toolbox for Matlab'.

S. Boyd & L. Vandenberghe (2004). *Convex Optimization*. Cambridge University Press.

J. Brookshire (2013). *Articulated Pose Estimation via Over-parametrization and Noise Projection*. Ph.D. thesis, Massachusetts Institute of Technology.

J. Brookshire & S. Teller (2012). 'Extrinsic Calibration from Per-Sensor Egomotion'. In *Robotics Science and Systems*.

J. Brookshire & S. Teller (2014). 'Articulated Pose Estimation via Over-parametrization and Noise Projection'. In *Robotics Science and Systems*.

H. Bruyninckx, P. Soetens, & B. Koninckx (2003). 'The real-time motion control core of the Orocos project'. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, vol. 2, pp. 2766 – 2771.

T.-J. Cham & J. Rehg (1999). 'A multiple hypothesis approach to figure tracking'. In *Computer Vision and Pattern Recognition*, vol. 2, p. 244.

C. Choi & H. Christensen (2011). 'Robust 3D visual tracking using particle filtering on the SE(3) group'. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pp. 4384–4390.

A. Comport, E. Marchand, & F. Chaumette (2007). 'Kinematic sets for real-time robust articulated object tracking'. *Image and Vision Computing* **25**(3):374–391.

A. P. Dempster, N. M. Laird, & D. B. Rubin (1977). 'Maximum likelihood from incomplete data via the EM algorithm'. *Journal of the Royal Statistical Society* **39**(1):1–38.

J. Deutscher, A. Blake, & I. D. Reid (2000). 'Articulated body motion capture by annealed particle filtering'. In *Computer Vision and Pattern Recognition*, pp. 126–133.

A. Doucet, S. Godsill, & C. Andrieu (2000). 'On Sequential Monte Carlo Sampling Methods for Bayesian Filtering'. *Statistics and Computing* **10**(3):197–208.

D. W. Eggert, A. Lorusso, & R. B. Fisher (1997). 'Estimating 3-D rigid body transformations: a comparison of four major algorithms'. *Applied Machine Vision* **9**(5-6).

P. Felzenszwalb, D. McAllester, & D. Ramanan (2008). 'A discriminatively trained, multiscale, deformable part model'. In *Computer Vision and Pattern Recognition*.

V. Govindu (2004). 'Lie-algebraic averaging for globally consistent motion estimation'. In *Computer Vision and Pattern Recognition*.

G. Grisetti, S. Grzonka, C. Stachniss, P. Pfaff, & W. Burgard (2007). 'Efficient estimation of accurate maximum likelihood maps in 3D'. In *Intelligent Robots and Systems*.

G. Grisetti, C. Stachniss, & W. Burgard (2005). 'Improving Grid-based SLAM with Rao-Blackwellized Particle Filters by Adaptive Proposals and Selective Resampling'. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*.

C. Hertzberg, R. Wagner, U. Frese, & L. Schröder (2011). 'Integrating Generic Sensor Fusion Algorithms with Sound State Representations through Encapsulation of Manifolds'. *Information Fusion* .

M. Isard & A. Blake (1998). 'CONDENSATION - conditional density propagation for visual tracking'. *International Journal of Computer Vision* **29**.

A. Jain & C. Kemp (2010). 'Pulling open doors and drawers: Coordinating an omni-directional base and a compliant arm with Equilibrium Point control'. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*.

S. Julier (2002). 'The scaled unscented transformation'. In *IEEE American Control Conference*, vol. 6, pp. 4555–4559.

Z. Kalal, K. Mikolajczyk, & J. Matas (2010). 'Forward-Backward Error: Automatic Detection of Tracking Failures'. *International Conference on Pattern Recognition* .

R. E. Kalman (1961). 'On the General Theory of Control Systems'. In *First International Congress of IFAC, Proceedings of the*.

K. Kanatani (1990). *Group Theoretical Methods in Image Understanding*. Springer-Verlag New York, Inc.

D. Katz, M. Kazemi, J. Bagnell, & A. Stentz (2012). 'Interactive Segmentation, Tracking, and Kinematic Modeling of Unknown Articulated Objects'. Tech. Rep. CMU-RI-TR-12-06, Robotics Institute.

L. Kavan, S. Collins, J. Zara, & C. O'Sullivan (2008). 'Geometric Skinning with Approximate Dual Quaternion Blending'. In *ACM Transactions on Graphics*, vol. 27. ACM Press.

D. Koller & N. Friedman (2009). *Probabilistic Graphical Models: Principles and Techniques*. MIT Press.

J. Kwon & K. M. Lee (2010). 'Monocular SLAM with locally planar landmarks via geometric rao-blackwellized particle filtering on Lie groups'. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pp. 1522–1529.

J. McCarthy (1990). *An Introduction to Theoretical Kinematics*. MIT Press.

W. Meeussen, M. Wise, S. Glaser, S. Chitta, C. McGann, et al.P. Mihelich, E. Marder-Eppstein, M. Muja, V. Eruhimov, T. Foote, J. Hsu, R. Rusu, B. Marthi, G. Bradski, K. Konolige, B. Gerkey, & E. Berger (2010). 'Autonomous Door Opening and Plugging In with a Personal Robot'. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*.

P. Pan & D. Schonfeld (2008). 'Adaptive resource allocation in particle filtering for articulated object tracking'. In *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, pp. 729–732.

W. Qu & D. Schonfeld (2007). 'Real-Time Decentralized Articulated Motion Analysis and Object Tracking From Videos'. *Image Processing, IEEE Transactions on* **16**(8):2129–2138.

J. M. Rehg & T. Kanade (1995). 'Model-based tracking of self-occluding articulated objects'. In *Computer Vision and Pattern Recognition*.

J. M. Rehg, D. D. Morris, & T. Kanade (2003). 'Ambiguities in Visual Tracking of Articulated Objects Using Two- and Three-Dimensional Models'. In *International Journal of Robotics Research*.

RITCHIESpecs (2013). 'CATERPILLAR 322BL Hydraulic Excavator'.

T. Ruhr, J. Sturm, D. Pangercic, M. Beetz, & D. Cremers (2012). 'A generalized framework for opening doors and drawers in kitchen environments'. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*.

J. Sturm, C. Stachniss, & W. Burgard (2011). 'A Probabilistic Framework for Learning Kinematic Models of Articulated Objects'. *Journal of Artificial Intelligence Research* **41**.

S. Thrun, W. Burgard, & D. Fox (2005). *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. MIT Press.

R. Urtasun, D. J. Fleet, & P. Fua (2006). '3D People Tracking with Gaussian Process Dynamical Models'. In *Computer Vision and Pattern Recognition*, pp. 238–245.

R. van der Merwe, A. Doucet, N. de Freitas, & E. A. Wan (2000). 'The Unscented Particle Filter'. In T. Leen, T. Dietterich, & V. Tresp (eds.), *Advances in Neural Information Processing Systems 13*.

Willow Garage (2013). 'Unified Robot Description Format (URDF)'.

J. Ziegler, K. Nickel, & R. Stiefelhagen (2006). 'Tracking of the Articulated Upper Body on Multi-View Stereo Image Sequences'. In *Computer Vision and Pattern Recognition*, pp. 774–781.