Utilizing Redundancy in Modular Robots to Achieve Greater Accuracy

Tom Larkworthy[1] and Gillian Hayes[1]

***Abstract*—Modular robot configurations typically feature many kinematically redundant loops. We believe that the information in these loops can be utilized in order to increase the accuracy of localization, in particular, at an end effector. We hope it will be possible to design an algorithm that can design configurations able to manipulate to a user specified level of accuracy. In order to do so, it would be necessary to predict the level of accuracy of a configuration from *a priori* information.**

In this work we provide experimental evidence that increased accuracy is easily achieved through redundancy. We then formulate a representation of accuracy as a distribution of location across space. We use Bayesian statistics to prototype three different models of the experimental system and test their ability to predict the increase in accuracy observed. We find that all three of the statistics prototyped were conservative estimators, leading us to the conclusion that our kinematic model of the system was too constrained.

I. Introduction

Modular robotics offers many advantageous over monolithic robotic solutions including flexibility, robustness *etc.* [1] One advantage is that redundant spatial information caused by redundant kinematic pathways can be fused to increase the overall accuracy of localization. This effect has been well utilized in monolithic parallel robotic architectures such as the Stewart platform [2], [3] to greatly improve accuracy over their serial counterparts. We believe that redundant kinematic pathways found commonly in modular robots can be used similarly to increase the accuracy of tool manipulation.

Accuracy is the ability of a forward model model to predict the position of a robot from its motor commands or proprioceptive sensors. Work on monolithic parallel actuation concentrated on calibrating forward models in order to increase the precision of a forward model [4] or analyzing the error workspace with respect to design parameter vectors [5]. Modular robots [6], [7], [1] offer a different challenge because the topology of the robot may not be known in advance. Therefore, any calibration step can only be applied to individual modular components rather than an assembled whole. The problem then for modular robots is predicting the accuracy of arbitrary robot configurations from *a priori* information.

Accuracy is related to localization. Substantial work has been carried out on localization of wireless sensor networks, driven by the desire for low cost spatially aware services [8]. In sensing networks it is usually assumed that a number of units have been distributed around an environment. They are capable of estimating the distance to a few of their nearest neighbors, and possibly the direction of their neighbors. The problem is then to determine the absolute locations of the units given the noisy relative sensor readings. Most work to date has concentrated on algorithms that can determining a point estimate of the positions of units when sensor noise is large. A point estimate of positions, however, is not useful as a measure of accuracy. Accuracy is the probability distribution of position across space. Nevertheless, these methodologies highlight the difficulty of inference across cyclic graphs [9], a technical issue that is preserved across the two problem domains.

Our long term aim is to develop a modular robot designer algorithm that utilizes kinematic redundancy in order to achieve a user specified level of accuracy. We expect to build on existing localization algorithms in order to achieve this. However, as the problem of inference on general graphs for localization is difficult, this preliminary work utilizes a very basic configuration of robotic components. This is so we can check the accuracy of different statistics against a real robot, without becoming bogged down in technical issues regarding calculation of those statistics.

We use a Bayesian statistical formulation for representing accuracy as a probability distribution of location. The Bayesian approach is particularly useful in this domain because it is itself modular. In section II we present experiments on a mock modular robot that demonstrate that accuracy is indeed increased through kinematic redundancy. Section III describes the non-parametric methodologies that we utilize in order to estimate probability densities from data gathered in section II. These general methodologies are applied to construct three different statistical models for predicting accuracy in section IV. We apply those statistical models and compare with experimental behavior in section V. Finally, general conclusions about utilizing redundancy in order to achieve increased accuracy, and the issues regarding predicting increases in accuracy, are presented in section VI.

II. Experimental setup

Our experimental setup attempts to reproduce the most salient features of a modular reconfigurable robot. It is comprised of two planar robots, whose end effector accuracy can be measured. They can be mechanically linked together to create an end effector that is controlled redundantly by the two robots. With this setup we can study the effect of kinematic redundancy on accuracy. The planar robots are mounted such that their workspaces are affected by gravity, an important source of inaccuracies in most robot installations.

[1] Institute of Perception, Action and Behaviour,
School of Informatics,
University of Edinburgh.

ROBOCOMM 2009, 31st Mar – 2nd Apr 2009, Odense, Denmark.
ISBN 978-963-9799-51-6
DOI 10.4108/ICST.ROBOCOMM2009.5819

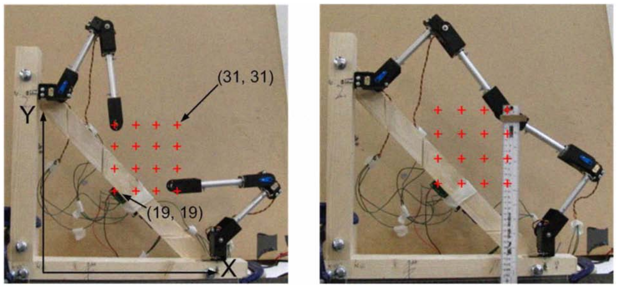

Figure 1. Left: A picture of the two 2DOF arms with the testing workspace superimposed (coordinates are in cm). Right: the arms linked with the Y axis measuring device attached.

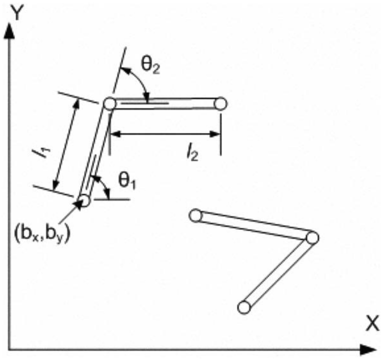

Figure 2. Kinematic schematic

Figure 1 shows the experimental setup. Each robot has two rotational degrees of freedom driven by Hitec HSR-TI5995TG servos. We selected 16 points from a rectangular area of the workspace for our experiments (Figure 1 left). Our measurement equipment could locate the distal endpoint of each arm to an accuracy of approximately 1mm (Figure 1 right). The arm attached to the vertical strut will be referred to as the top arm, and the arm attached to the base of the equipment the bottom arm.

The distal endpoints of each arm could be linked together with a rod to form a combined robotic setup. Kinematically the rod added an additional passive rotary joint to the combined kinematics. The combined distal end point thus became a function of both arms' state. When the equipment is configured as such it will be referred to as the combined arm.

We modeled each individual arm as two rigid rods connected by two rotary joints (Figure 2). In this scheme the distal end point of an arm was defined by the following forward model:-

$$\boldsymbol{d} = \begin{matrix} b_x + l_1 cos(\theta_1) + l_2 cos(\theta_1 + \theta_2) \\ b_y + l_1 sin(\theta_1) + l_2 sin(\theta_1 + \theta_2) \end{matrix} = f(\boldsymbol{\theta})$$

The angles θ_1 and θ_2 used in the forward model are calculated from the motor commands. The technical sheet of the servos states that the angle is proportional to the width of the pulses sent to the servo. Thus a linear transformation converts between motor command, $\boldsymbol{u}$(pulse widths, ms) to angles, $\boldsymbol{\theta}$ (rad):-

$$\boldsymbol{\theta} = \begin{matrix} a_1 u_1 + b_1 \\ a_2 u_2 + b_2 \end{matrix} = g(\mathbf{u})$$

We estimated the parameters of the forward models f and g from training data. We moved each arm approximately to points on a 4x4 grid in distal space (Figure 1 left). We recorded the motor commands sent, and the distal locations reached. There were 16 training data points in total for each arm. We also applied the exact same 16 motor commands a second time to each arm. We reserved the second set of data points as a validation set.

The quality of a set of training parameters on a data set of N motor commands $\mathbf{u}$ and N observed distal locations $\mathbf{d}$ was defined to be the summed Euclidean distance between the predicted position and observed position over the set.

$$J = \sum_{i=1}^{N} ||f(g(\boldsymbol{u}_i)) - \boldsymbol{d}_i||$$

A basic hill climbing algorithm was used to improve an initial guess of the kinematic parameters on the training set. Overfitting was prevented by halting the algorithm when the cost function rose when applied to the validation set. The initial guesses for the parameters were determined by simply measuring the lengths of the links and base positions, and reading the servo technical specification.

Once the forward parameters were learned, it was possible to compare the agreement between the forward model and the observed behavior. Figure 3 top left and top middle shows the difference between the arms commanded position and the actual location reached. The forward models have an average error of roughly zero, demonstrated by the fact that the errors point in all directions, but are systematically biased across the workspace. This systematic bias manifests itself on the workspace error diagrams non-random patterning. For the top arm, the biases appear as a smooth swirl of the error arrows. For the bottom arm, the majority of the errors form a swirling pattern, but the lower and upper datapoints on the leftmost column break from the continuum. This indicates a significant change in the kinematics which we attribute to the center of mass, which is concentrated in the servo of the elbow joint, acting in a different torque direction at the base joint when the arm stretches to reach the far away positions in the workspace. When the center of mass is resolved to an opposite torque direction, backlash in the base joint acts in a different direction and so the distal end point moves significantly for a small displacement in joint space.

After training the kinematic models we linked the arms and commanded both arms redundantly to reach each of the 16 desired distal workspace locations. To determine the correct motor commands to send to each arm, the forward model functions were inverted, denoted $f^{-1}$and g^{-1}. An analytical solution for f^{-1} was determined using the law of cosines. There was an ambiguity that arose in the derivation caused by the fact that two angular solutions exist for a single given distal point. The correct solution is easily determined by picking the solution that lies closest to a training example.

With inverse kinematics defined we commanded both arms to each of the 16 test locations and recorded the actual locations reached. This formed our combined arm data set (Figure 3 top right).

Digital Object Identifier: 10.4108/ICST.ROBOCOMM2009.5819
http://dx.doi.org/10.4108/ICST.ROBOCOMM2009.5819

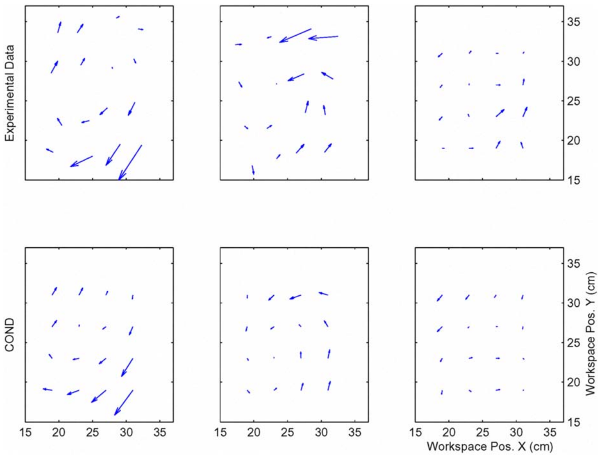

Figure 3. Plots of accuracy errors over the workspace for the top arm (left), bottom arm (middle) and the combined arms (right). The top row is the discrepancy between the forward models and the experimental validation set. The lower row is a summary of the expected error over the workspace for the COND model. The start of each arrow denotes the the prediction of the trained forward model for the motor commands issued, and the arrow head points in the direction of the observed location (top row), or expected location (bottom row). The length of the arrow is proportional to the magnitude of the observed error (top row), or expected error (bottom row).

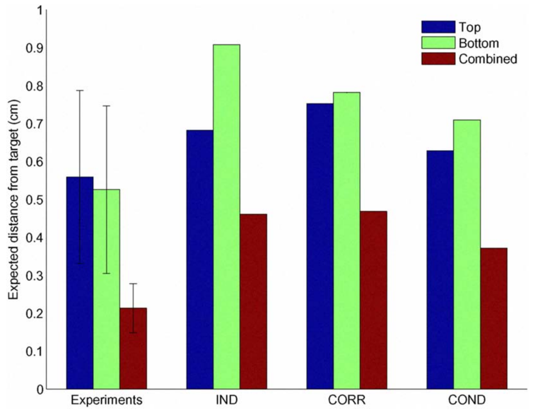

Figure 4. First column, the mean accuracy of the experiments. Error bars denote the 95% confidence interval for the estimation of the mean. Middle and right columns, predictions of accuracy by each of the candidate statistical models.

To quantify the overall accuracy of the different setups, we averaged the Euclidean distance between the predicted distal point for the motor commands against the observed distal point. The results are summarized in Figure 4 (leftmost column). We note both arms individually have similar accuracy performance, but the combined arm setup is significantly more accurate.

III. Preliminaries

We have modeled accuracy as a probability density function (PDF) of location across space, which is dependent on the motor commands sent $p(\boldsymbol{d}|\boldsymbol{u})$. In section V we will construct an estimate of this distribution by modeling noise at the joints of the robots as PDFs and fusing them under different assumptions. The underlying PDFs are estimated from data gathered in section II. We use non-parametric techniques to estimate the PDF, gathered from data which are less prone to skewing the data compared to parametric techniques. This section describes, in brief, the density estimation techniques used.

For density estimation of PDFs of the form $p(\boldsymbol{d})$ we have used Parzen windowing [10] with leave-one-out cross validation (see [11] for a good explanation) using spherical Gaussian kernel functions (SGF). Spherical Gaussian kernels are bad estimators when data is spread unevenly across dimensions. We reduce the impact of this problem by first normalizing the data into a dataset $\tilde{D}$ by dividing by the componentwise standard deviation. Let x_i denote single training example encoded as a real valued vector. Then the training set D is:-

$$D = (\boldsymbol{x}_1^T ... \boldsymbol{x}_N^T)^T$$

$$\mu_i = \frac{1}{N}\sum_j^N x_{ji} \text{ , } \sigma_i = \sqrt{\frac{1}{N}\sum_j^N (x_{ji} - \mu_i)^2}$$

$$\tilde{D} = (\tilde{\boldsymbol{x}}_1^T ... \tilde{\boldsymbol{x}}_N^T)^T \text{where } \tilde{x}_{ji} = \frac{x_{ji}}{\sigma_i}$$

A spherical Gaussian of width h is then placed at each datapoint as an estimation of the PDF. To obtain the value of the estimated PDF at a (unnormalized) test point $\boldsymbol{x}$, the test point first must be normalized, then the spherical Gaussian kernel functions summed, and then the result adjusted to account for the change of variable in the normalization process:

$$SGF(\boldsymbol{x}|\boldsymbol{\mu}, h) = \frac{1}{\sqrt{(2\pi h^2)^{dims}}} e^{(-\frac{1}{2h^2}(\boldsymbol{x}-\boldsymbol{\mu})^T(\boldsymbol{x}-\boldsymbol{\mu}))}$$

$$pdf(\boldsymbol{x}|D, h) = diag(\sigma_1 ... \sigma_N) \sum_i^N SGF(\tilde{\boldsymbol{x}}|\tilde{\boldsymbol{x}}_i, h)$$

We obtain a good value for h by minimizing a leave-one-out cross validation cost function. Let D_{-j} denote the D dataset less row j (i.e. the complete dataset with datapoint j missing). Then the leave-one-out error is calculated as:-

$$J_{LOO}(h) = \sum_i^N \sum_j^N (pdf(\boldsymbol{x}_i|D, h) - pdf(\boldsymbol{x}_i|D_{-j}, h))^2$$

We minimize this quantity to obtain a good value for h.

$$h_{LOO} = \underset{h}{arg\, min}\, J_{LOO}(h)$$

We will denote the Parzen leave-one-out PDF value at point $\boldsymbol{x}$ which was learned from a dataset D as $pdf_{LOO}(\boldsymbol{x}|D)$.

For estimation of conditional distributions of the form, $p(\boldsymbol{x}|\boldsymbol{z})$, we first estimated the joint distribution $p(\boldsymbol{x}, \boldsymbol{z})$ using Parzen leave-one-out estimator as above. To condition a variable we converted the Parzen estimate of the joint distribution into a multivariate histogram. Hyper-rectangular bins were placed evenly over the region of interest, and assigned a value taken from the estimated joint distribution based on the position of the bin's center. The estimate of $p(\boldsymbol{x}|\boldsymbol{z})$was then:

$$p(\boldsymbol{x}|\boldsymbol{z}) \approx \frac{value\, in\, bin\, (\boldsymbol{x}, \boldsymbol{z})}{\sum_{\forall k} value\, in\, bin\, (\boldsymbol{k}, \boldsymbol{z})}$$

IV. Statistical Models

For the modular robot accuracy prediction problem, we need to predict the accuracy of a configuration using data gathered from the modular components in isolation. In this

Digital Object Identifier: 10.4108/ICST.ROBOCOMM2009.5819
http://dx.doi.org/10.4108/ICST.ROBOCOMM2009.5819

work, each arm constitutes a single modular component, and the combined arm setup the target of our accuracy predictions. We model distal error in the individual arms as the result of random variables in the kinematic formulation. The distal position of the end effector then becomes a random variable itself, conditioned on the motor commands sent. We used these distal location distributions under a conditional independence assumption to form a predicted distal location distribution when the arms were linked and controlled cooperatively.

We modeled each arms' individual inaccuracies as noise at the actuated joints. The forward model was adjusted so that an angular error term alters the resultant position of the distal point:

$$\mathbf{d} = f(g(\mathbf{u}) + \boldsymbol{\theta}_{err})$$

By defining a probability density function of the angular error term we can introduce uncertainty into the kinematic model. We tried three different noise models (described below), one model where sensor noise at individual joints is assumed independent of the entire system (IND), one where the two joint sensor noises are assumed independent of the system but can be correlated (CORR), and one where the sensor noises are conditioned on the motor commands issued (COND).

We converted the training data sets in section II that were of the form $(\boldsymbol{u}, \boldsymbol{d}_{obs})$ into a set of motor commands $\boldsymbol{u}$ and a set of angular errors $\boldsymbol{\theta}^{err}$. $\boldsymbol{\theta}^{err}$was calculated by $\boldsymbol{\theta}_i^{err} = f^{-1}(\boldsymbol{\theta}_i) - g(\boldsymbol{\theta}_i)$.

For the first noise model, IND, each $\boldsymbol{\theta}_i^{err}$ was assumed to be generated independently from an unknown density to be estimated. We estimated the probability density using a Parzen window technique using leave-one-out cross validation.

$$\begin{aligned} pdf_{INF}(\theta_{err1}, \theta_{err2}) \quad &\propto \quad pdf_{LOO}(\theta_{err1}|(\theta_{11}^{err}...\theta_{N1}^{err})^T) \times \\ &\quad\; pdf_{LOO}(\theta_{err2}|(\theta_{12}^{err}...\theta_{N2}^{err})^T) \end{aligned}$$

The Parzen estimation of these variables is shown in Figure 5. Note that in both the top arm and the bottom arm, the angular noise is greatest for the joint nearest the base, as would be expected if gravity had a significant effect. For the top arm, both curves appear quite Gaussian. For the bottom arm data, distinct non-Gaussian artifacts are seen. The cause of these we attribute to the fact that most of the mobile mass for the bottom arm is concentrated at the servo controlling the elbow joint. This mass is over the opposite side of the base joint w.r.t gravity for movements to a distal location at the lower left of the test workspace, evident in experimentation (Figure 3 top middle, note the distinct lack of continuity at the lower left of the diagram).

For the second noise model, CORR, the angular noises are permitted to be correlated. We compute the correlation between the angular error variables by estimating a joint distribution using a multivariate Parzen estimator.

$$pdf_{CORR}(\boldsymbol{\theta}_{err}) = pdf_{LOO}(\boldsymbol{\theta}_{err}|\boldsymbol{\theta}^{err})$$

The estimated joint distribution is shown in Figure 6. Note that correlation between the variables is strongest for the bottom arm. This is probably also because of the effect of the center of mass switching sides. When the dynamics change slightly, the kinematic model attributes some error to both joints, thus some form of correlation becomes apparent when viewed as joint error.

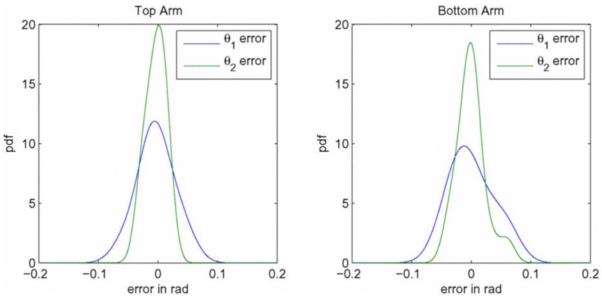

Figure 5. Parzen density estimation of angular noise under the IND model. Noise is assumed to be generated independently from two uni-variate distributions.

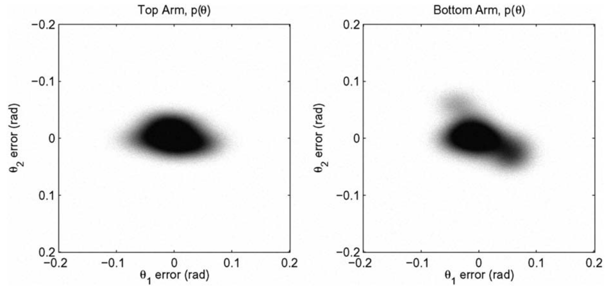

Figure 6. The multivariate estimated PDF function of sensor noise for the CORR model. The bottom arm data suggests a negative correlation is present between the angular error variables.

Finally for the third model, noise is conditioned on the state of the motor commands. We first construct an estimate of a joint distribution for all the variables using a multivariate Parzen window estimator.

$$pdf_J(\boldsymbol{\theta}_{err}, \boldsymbol{u}_C) = pdf_{LOO}(\boldsymbol{\theta}_{err}, \boldsymbol{u}_C|\boldsymbol{\theta}^{err}, \boldsymbol{u})$$

The use of the Parzen window allows us to obtain a smooth estimate of the probability density function over the entire range of motor commands and sensor errors. For predicting the expected angular error when an arm is controlled, we need to convert the joint distribution into a conditional density of the sensor noise given the motor commands. We converted the Parzen estimate of the joint into a histogram of the joint, see section III, with 10 bins across each of the four dimensions. Each bin hyper-rectangle was .01 rads by 0.1 rads by 50 ms by 50 ms.

$$pdf_{COND}(\boldsymbol{\theta}_{err}|\boldsymbol{u}) = histCond(\boldsymbol{\theta}_{err}|\boldsymbol{u}, pdf_J)$$

We avoid displaying the estimated pdf_{COND} in the form of angular error and motor commands, because it is more naturally displayed as the resultant distal error conditioned on the desired target location, as in Figure 7. Will look at this specifically in section V.

All the models make assumptions about how the forward model can be inaccurate. The first model, IND, is extremely

Digital Object Identifier: 10.4108/ICST.ROBOCOMM2009.5819
http://dx.doi.org/10.4108/ICST.ROBOCOMM2009.5819

simplistic, while the third model, COND, is capable of learning that the noise profiles change over the workspace. The third model has more scope for predicting forward model errors caused by dynamic effects such as gravity. However, it must be noted that the effects of gravity can only be expressed in the equations as angular noise, whereas in the real situation gravity effects will probably alter the whole geometry of the arms. The second model lies somewhere between the first model and third model in terms of being able to express dynamic effects.

V. Predictions

With the models defined we can make predictions about the accuracy of the arms individually and when combined over the workspace. For a given target distal location $\boldsymbol{d}_{targ}$ we computed the motor command $\boldsymbol{u}$ using the inverse models $\boldsymbol{u} = g^{-1}(f^{-1}(\boldsymbol{d}_{targ}))$.We then compute the probability of the end point actually being at another location $\boldsymbol{d}_{tst}$ by computing the angular difference, θ_{err} between the kinematic models of $\boldsymbol{d}_{targ}$ and $\boldsymbol{d}_{tst}$ by $\boldsymbol{\theta}_{err} = f^{-1}(\boldsymbol{d}_{targ}) - f^{-1}(\boldsymbol{d}_{tst})$ and consulting the relevant probability distribution, $pdf_{MODEL}(\boldsymbol{\theta}_{err}|\boldsymbol{u})$ ($\boldsymbol{u}$ has no effect for the first two models, so this simplifies to $pdf_{MODEL}(\boldsymbol{\theta}_{err})$).

To predict the distal behavior of the combined arm setup, we made the assumption that each arms' angular errors are independent. This implies the probability of the combined end effector being at a position $\boldsymbol{d}_{tst}$ when the motor commands were aiming for d_{targ} as:-

$$\begin{aligned} pdf(\boldsymbol{d}_{tst}|\boldsymbol{d}_{targ}) \quad \propto \quad & pdf_{MODEL}(\boldsymbol{\theta}_{err}^{TOP}|g_{TOP}^{-1} f_{TOP}^{-1}(\boldsymbol{d}_{tst})) \times \\ & pdf_{MODEL}(\boldsymbol{\theta}_{err}^{BOT}|g_{BOT}^{-1} f_{BOT}^{-1}(\boldsymbol{d}_{tst})) \end{aligned}$$

Each arms' sensor noise profiles are, of course, not independent. If the two arms were given motor commands to reach two different distal locations, the mechanically linked distal end effector would prevent separation and the actual location reached would be determined by a tug-of-war between the arms. So the final set of angular errors is very much a function of both arms' commands. In experimentation however, we always gave commands that aimed to move both arms to the same distal location. We note though, that the forward models were biased over the workspace (Figure 3 top left, middle), so the validity of this assumption is compromised by the accuracy of the forward models.

For all the models and setups, a test point $\boldsymbol{d}$ far away from $\boldsymbol{d}_{targ}$will have a near zero value in the PDF. So by histogramming over a reasonable bounds around $\boldsymbol{d}_{targ}$we can normalize once in binned histogram form. For all graphics presented in this article, the histogrammed region was a rectangle ±3cm around the $\boldsymbol{d}_{targ}$ with 100x100 bins. The expected position of the end effector for a motor command $\boldsymbol{u}$ is $\boldsymbol{d}_{exp}^{u} = \sum_{\forall \boldsymbol{d}_{tst}} \boldsymbol{d}_{tst} pdf(\boldsymbol{\theta}_{err}|\boldsymbol{d}_{targ})$

We will only show the PDF variation for different target location for the COND model, as it is the only model that is capable of interesting variation. Figure 7 shows the probability of the end effector being in a location given the desired location for all 16 different target test points for the COND model on all three setups.

We used this data to generate the expected location of the end point given a motor command and compared it with the actual target locations predicted by the forward model. This provides us with a method for predicting the expected errors in the forward model for the experiments in section II, and is the basis for Figure 3.

To assess the general accuracy we summed and normalized the relative distal error histograms over the 16 test locations. This provides us with an expected distal error over the entire test workspace. We can compare this with data from the experiments in section II. Figure 8 shows the resultant distal error probability densities for each of the three models alongside the experimental data. Only the COND model displays contour variation that agrees well with the observed data for the individual arms. All the models predict an increase in accuracy for the combined arm setup, evident by the reduced spread around the zero distal error location.

We can further summarize the results by computing the expected Euclidean distal error distance over all 16 workspace locations, to give us a single numerical value for the accuracy. These overall results are shown in Figure 4. All three models predict an increase in accuracy when the arms are combined. All of the statistical models appear to underestimate the accuracy in all cases. It is interesting that all the models predict the bottom arm as less accurate than the top arm, even though the experiments suggest the opposite. We attribute this to the fact that the bottom arms' dynamics are more complex due to backlash. The statistical models cannot account for the discontinuities caused, and so integrate it erroneously as increased noise over the system. Overall the COND model is the better predictor of accuracy.

VI. Conclusion

Despite the forward models ignoring a large source of bias from gravity, we observed an increase in accuracy of the forward models when the arms were physically linked. This encourages us that modular robots could utilize redundancy quite simply in order to become more accurate.

As none of our models performed well at predicting the accuracy of the individual arms, we conclude that the use of a kinematic model as a forward model introduced significant error. The increased accuracy of the COND model suggests that some of the forward model inaccuracies could be corrected for. However, its predictive power was limited by the fact that corrections could only be expressed through angular noise.

For predicting the accuracy of the combined arm setup we fused the individual arm models under an independence assumption. Because the individual arm models did not fit the experimental data well it is difficult to evaluate the validity of this assumption. That said, the relative increase accuracy as shown in Figure 4 appears to be about right, so this assumption may be a pragmatic approximation for later works.

For our future work we will add dynamics and backlash into the forward models and obtain data from a complex modular robot. A modular system with a variety of possible kinematic

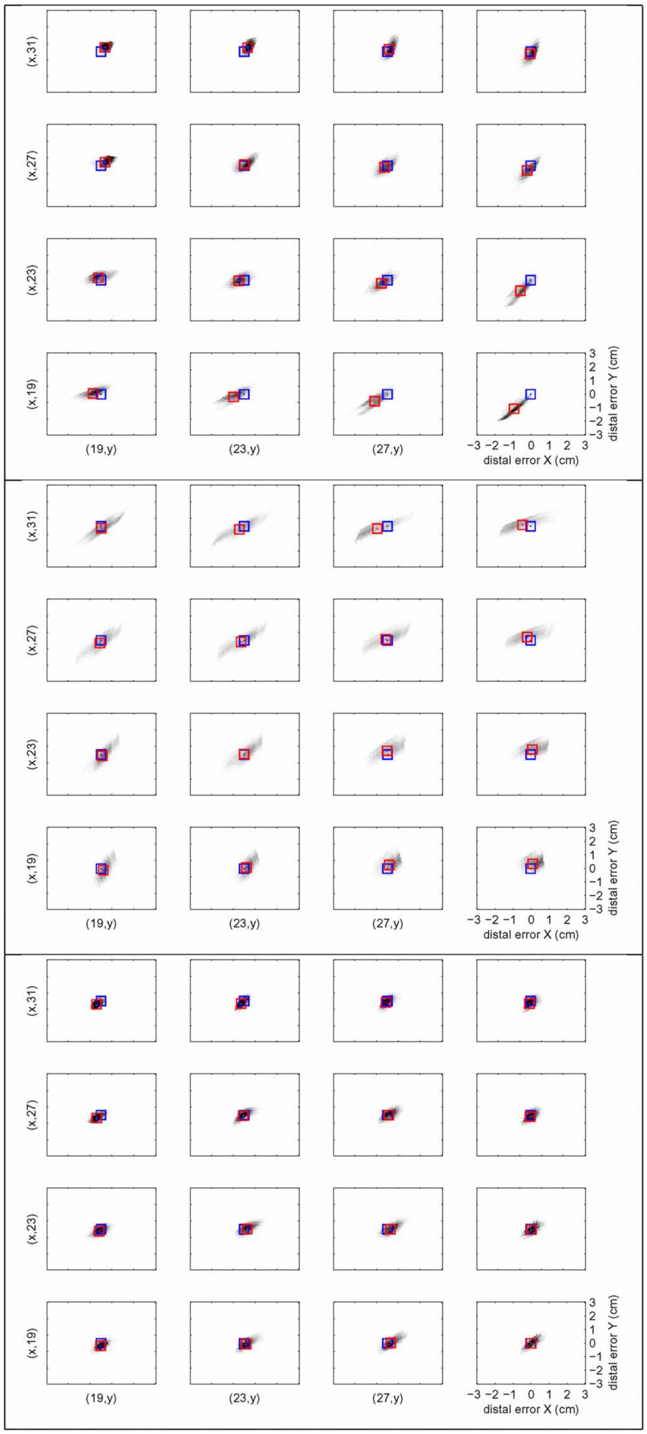

Figure 7. The resultant PDF of $p(\mathbf{d}|\mathbf{u})$ for the COND model over the 16 test workspace locations, for the top arm (top), bottom arm (middle) and the combined arms (bottom). Blue squares denote the zero distal error location in the plot, and the red square denotes the expected error (the vector between the squares is the same quantity as in Figure 3).

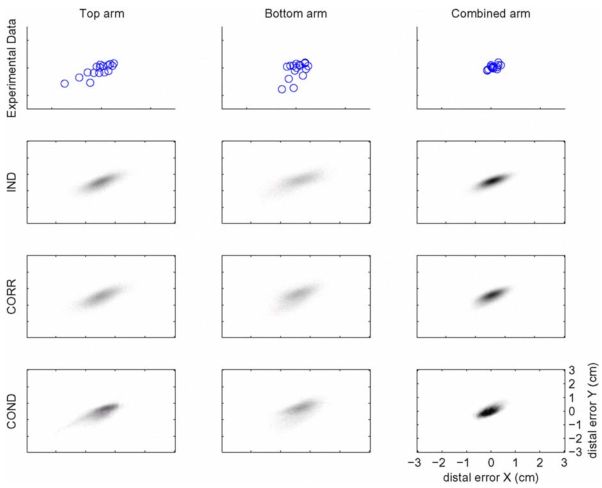

Figure 8. Top, distal error between validation set data and forward model prediction on the top arm (top left), bottom arm (top middle) and combined arm (top right). Remaining rows, expected distal error integrated over the 16 test points for each of the statistical models.

configurations will allow us to properly estimate the dynamics parameters and verify general purpose accuracy prediction methods. We hope this will enable us to design modular robot configurations autonomously that could manipulate tools to a user defined level of accuracy.

VII. References

References

[1] M. Yim, W.-M. Shen, B. Salemi, D. Rus, M. Moll, H. Lipson, E. Klavins, and G. S. Chirikjian, "Modular self-reconfigurable robot systems – challenges and opportunities for the future," *IEEE Robotics and Automation Magazine*, vol. March, pp. 43–53, 2007.

[2] D. Stewart, "A platform with six degrees of freedom: A new form of mechanical linkage which enables a platform to move simultaneously in all six degrees of freedom developed by elliott-automation," *The Institution of Mechanical Engineers, Proceedings*, vol. 180 Part 1, no. No. 15, pp. 371–386, 1965.

[3] B. Dasgupta and T. S. Mruthyunjaya, "The stewart platform manipulator: A review," *Mechanism and Machine Theory*, vol. 35, no. 1, pp. 15–40, 2000.

[4] O. Masory, J. Wang, and H. Zhuang, "Kinematic modeling and calibration of a stewart platform," *Advanced robotics*, vol. 11, pp. 519–539, 1997.

[5] T. Ropponen, T. Arai, "Accuracy analysis of a modified stewart platform manipulator," *Robotics and Automation, 1995. Proceedings.*, vol. 1, pp. 521–525, 1995.

[6] A. Lyder, R. Garcia, and K. Stoy, "Mechanical design of odin, an extendable heterogeneous deformable modular robot," in *International Conference on Intelligent Robots and Systems (IROS2008)*.

[7] M. Jorgensen, E. Ostergaard, and H. Lund, "Modular atron: Modules for a self-reconfigurable robot," in *International Conference on Robots and Systems, (IROS2004)*, 2004.

[8] N. Patwari, J. Ash, S. Kyperountas, A. H. III, R. Moses, and N. Correal, "Locating the nodes: cooperative localization in wireless sensor networks," *Signal Processing Magazine, IEEE*, vol. 22, no. 4, pp. 54 – 69, July 2005.

[9] S. Funiak, P. Pillai, M. P. Ashley-Rollman, J. D. Campbell, and S. C. Goldstein, "Distributed localization of modular robot ensembles," in *Proceedings of Robotics: Science and Systems*, June 2008.

[10] E. Parzen, "On the estimation of a probability density function and mode," *Annals of Mathematical Statistics*, vol. 33, pp. 1064–1076, 1962.

[11] C. Archambeau, M. Valle, A. Assenza, and M. Verleysen, "Assessment of probability density estimation methods: Parzen window and finite gaussian mixtures," *Circuits and Systems, 2006, Proceedings. 2006 IEEE International Symposium on*, vol. 1, p. 4, 2006.

Digital Object Identifier: 10.4108/ICST.ROBOCOMM2009.5819
http://dx.doi.org/10.4108/ICST.ROBOCOMM2009.5819