The 2010 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 18-22, 2010, Taipei, Taiwan

Statistical Visual-Dynamic Model for Hand-Eye Coordination

Daniel Beale, Pejman Iravani and Peter Hall

Abstract— **This paper introduces a new statistical method for combining vision and robot dynamics to generate trajectories to intercept a moving object. Previous methods only use information from the kinematics without considering the forces needed to move along the trajectory. Using robot dynamics allows extra measures, such as energy efficiency, to be optimised alongside maximising the likelihood of intercepting the target. We derive a statistical model for a vision system and a Lagrangian dynamical model of a robotic arm, showing how to relate joint torques to the vision. The method is tested by applying it to the problem of catching a simulated moving object.**

I. INTRODUCTION

Hand-eye coordination is an important problem in robotics, enabling a robot equipped with a vision system to manipulate visually sensed objects. This is particularly important for robots operating in dynamic environments, where the robot must reach a certain pose without a map of its workspace, in potentially noisy conditions.

The problem can be broken down into two distinct areas *static* and *timed* hand-eye coordination. In the static case, the robot must be able to move towards a point which is static in the scene, for example the picking and placing of fixed objects within the task space. A robot equipped with timed hand-eye coordination can move to match a trajectory of points through time.

A significant amount of research has been done in the area of static hand-eye coordination; the image Jacobian can be estimated, relating velocities in the joints of the robot to points in the image [Chaumette and Hutchinson, 2006], [Chaumette and Hutchinson, 2007]. In a partitioned approach [Corke and Hutchinson, 2001] several points are tracked on the end-effector, ensuring the manipulator reaches the correct pose, taking care of the redundant degrees of freedom in the kinematics.

Timed hand-eye coordination, introduced in [Allen et al. 1993], presents a much more difficult problem, since the target object needs to be tracked using a reliable motion model. Progress has been made towards a solution using the image Jacobian, extending it to match the velocity of a visual trajectory [Chaumette and Hutchinson, 2007]. Dynamical systems can also be used to generate timed trajectories which make an interception [Santos and Ferreira, 2009], while allowing the robot to learn natural grasping movements. The area of *active vision* [Hong and Slotine, 1997] introduces moving cameras to track and catch an object in real-time, reducing the need for complex vision algorithms and wide field-of-view cameras. This is extended into the field of mobile robotics, to intercept a moving ball. Methods in this area use statistics to find a distribution of possible ball positions [Guerrero et al. 2008], [Santos and Lima, 2009]; and then use controllers to intercept it [Capparella et al. 2005].

Both static and timed coordination have been solved using kinematics, giving positions, velocities and accelerations for the robot to move to. It is also important for the robot to be given the torques needed to make the movement. Force and vision have been combined in this way to better estimate the control the robots position [Beaten and Schutter, 2004], [Lippiello et al. 2007], and allow the robot to apply a force to an object while maintaining a particular pose [Nelson et al. 1995]. This gives the robot the ability to use vision to position the arm and force sensors to undertake tasks which require more precision.

In this paper a method is explored for timed hand-eye coordination that can generate trajectories which are optimal in terms of the vision and dynamics of the robot. The paper contributes to the current literature with,

- A novel statistical model for hand-eye coordination, which allows priors based on *robot dynamics* to be considered for intercepting a moving object, where previously only kinematics can be used.

This applies particularly to robots which are force controlled and designed to be compliant, trajectories can be chosen which minimise energy or are robust to external disturbances. Current methods generate trajectories which are based entirely on the kinematics of the arm and target object, negating the use of criteria based on joint torques.

We test our method to show that it is robust to noise in the vision system, and flexible to a range of different criteria in generating trajectories.

II. VISUAL MODEL

In this section a statistical model of the robots vision system is built. This allows us to find points that are most accurate, giving a basis to infer which are the best places to perform an action.

A. Geometric Uncertainty

It is conceptually clear that accuracy in depth perception is dependent on the configuration of any visual sensors, and also the point being viewed. This is demonstrated by deriving

This work was not supported by any organization

Daniel Beale, Department of Computer Science, University of Bath, Claverton Down, Bath, BA2 7AY. d.beale@bath.ac.uk

Pejman Iravani, Department of Mechanical Engineering, University of Bath, Claverton Down, Bath, BA2 7AY. p.iravani@bath.ac.uk

Peter Hall, Department of Computer Science, University of Bath, Claverton Down, Bath, BA2 7AY. pmh@cs.bath.ac.uk

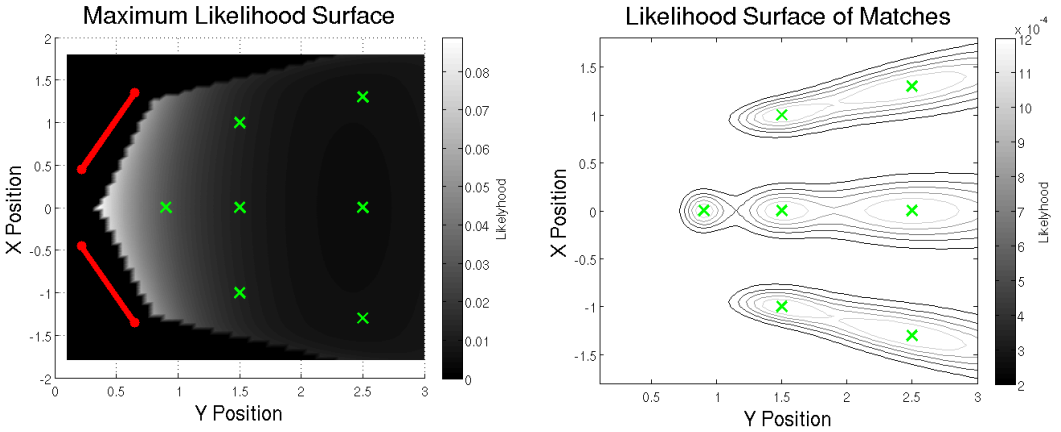

Fig. 1: On the left, the maximum-likelihood surface in triangulating a matching point across the whole workspace is shown. The red lines denote the camera positions and the corresponding likelihood is quantified by the saturation. This is the maximum value of distribution 1, for each point $\mathbf{x}$. The diagram on the right shows the probability distributions given a selection of triangulated points (marked by crosses), this is a direct calculation of equation (4).

a Bayesian model of the sensors, and observing the resulting error distribution.

The 3D location of a point is represented as $\mathbf{x} \in \mathbb{RP}^3$ and cameras modeled by two projection matrices P_1, P_2 [Zisserman, 2000]. The corresponding points in each camera are given by x_1, x_2 where,

$$x_i = P_i\mathbf{x} + \varepsilon_i, \tag{1}$$

with $\varepsilon_i \sim N(0, \Sigma_i)$ and $i \in \{1, 2\}$.

Letting $S_i := (P_i, \Sigma_i)$ be the parameters of each camera we have $p(x_i|\mathbf{x}, S_i) = N(x_i|P_i\mathbf{x}, \Sigma_i)$, the likelihood of the three dimensional point given the camera parameters.

It remains to find the probability distribution of points in three dimensions. Each camera reading is assumed to be statistically independent, and so with an application of Bayes theorem and the assumption that $p(\mathbf{x}|S_i)$ is uniformly distributed, the following holds,

$$p(\mathbf{x}|C) = p(\mathbf{x}|x_1, x_2, S_1, S_2) \tag{2}$$

$$\propto \prod_{i\in\{1,2\}} p(\mathbf{x}|x_i, S_i) \tag{3}$$

$$\propto \prod_{i\in\{1,2\}} p(x_i|\mathbf{x}, S_i). \tag{4}$$

This gives the likelihood that the triangulated point is correct, given the measurements from both cameras. Figure 1 shows how the accuracy in calculating the position of a world point changes depending on where it is with respect to each of the cameras.

B. Predictive Uncertainty

When tracking a moving object we would like to be able to predict how it will move in the future. Doing this will allow enough time to make a decision and perform an action, before the object reaches a particular state.

In making this kind of prediction, the uncertainty grows with every time step taken without new data. Assuming that the object follows linear dynamics, the movement is modelled using a Kalman filter. This gives an estimated position and velocity for the object, and allows us to predict its state arbitrarily into the future.

A covariance matrix representing the corresponding measure of uncertainty in estimating the state is also calculated, complementary to the probabilistic model used so far.

For each measured state $\mathbf{s}_t = \begin{pmatrix} z_t \\ v_t \end{pmatrix}$ at discrete time $t \in \mathbb{N}$, where z_t is the position and v_t is the velocity, the following relation holds,

$$\mathbf{s}_t = A\mathbf{s}_{t-1} + \varepsilon_t, \tag{5}$$

where A is the state-transition matrix and $\varepsilon_t \sim N(0, Q)$ is zero mean Gaussian noise. Letting $H_{(i,j)} = [\mathbf{s}_i, \mathbf{s}_{i+1}, ..., \mathbf{s}_j]$, the following matrix equation is solved finding a least-squares solution for A.

$$AH_{(1,N-1)} = H_{(2,N)} \tag{6}$$

$$A = H_{(2,N)}(H_{(1,N-1)})^*. \tag{7}$$

Where $(H_{(1,N-1)})^*$ is the Moore-Penrose pseudo inverse of the matrix $H_{(1,N-1)}$. This requires that N states are measured before any prediction can take place.

For known states the position and variance is estimated using the predict-update equations of a Kalman filter, for future time intervals the position is predicted using equation 5, and covariance predicted as follows,

$$M_t = AM_{t-1}A^T + Q. \tag{8}$$

This ensures that the covariance increases appropriately with the amount of prediction.

To incorporate this information in to the final optimisation a probability distribution $p(\mathbf{x}|t, K)$ is required where $t \in \mathbb{N}$ is the time and $K := H_{(1,N)}$ denotes the set of past measurements.

The probability distribution is given as follows,

$$p(\mathbf{x}|t, K) = p(\mathbf{x}|t, H_{(1,N)}). \tag{9}$$

$$= N(\mathbf{x}|\mathbf{s}_t, M_t) \tag{10}$$

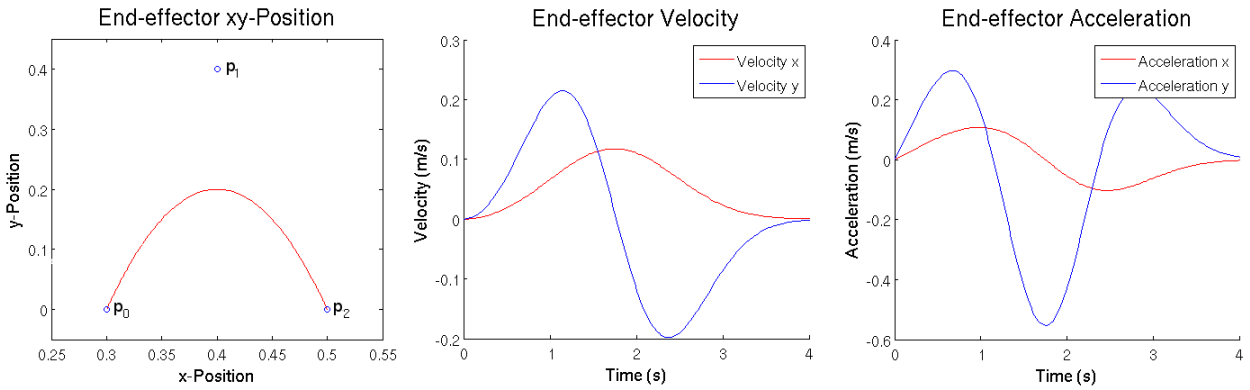

Fig. 2: The position, velocity and acceleration are shown for a 2-dimensional Bezier curve, that has been modified by shaping the time input. The graphs demonstrate that in each dimension the trajectory has zero velocity and acceleration at the end points.

Here $\mathbf{x}$ denotes the true position of the target and $\mathbf{s}_t$ the measured state at time t. This distribution incorporates knowledge about the most accurate places in the vision system to catch the object, and also the best point in time.

C. Combining geometric and predictive uncertainty

We now have a probabilistic model governing the geometric relationship of the camera system, showing how the error in measuring a point is distributed in three dimensions (equation (4)). A Kalman filter allows us to measure the position of an object and predict its future position and error distribution (equation (9)). It remains to show how these two distributions can be combined, to predict the position of a moving object while triangulating its position using multiple cameras.

Using the independence of the camera C and the Kalman filter with time (K, t), the likelihood of the position $\mathbf{x}$, given data from the tracked target $\beta := (C, K)$ is calculated.

$$p(\mathbf{x}|\beta, t) = p(\mathbf{x}|C, K, t) \tag{11}$$
$$\propto p(\mathbf{x}|C)p(\mathbf{x}|K, t) \tag{12}$$

This probability distribution gives a complete likelihood of the estimated position of a moving object, also allowing for future positions to be predicted. The formulation is general enough for alternative tools to be used if needed. For example, where the assumption of linear dynamics in the Kalman filter may be too limiting in some circumstances, it is possible to be easily exchanged with another particle filter which also gives information about the uncertainty in prediction.

III. ARM DYNAMICS

There are an infinite number of movements a robot can make in order to achieve a particular task. In this section we analyse a set of trajectories by considering the inverse dynamics of the robot, i.e. the torques needed to make the movement. This is then used as a basis to decide on which trajectory is the best.

To determine the inverse dynamics, a Katana robot is modeled in the program DySim [Sahinkaya, 2004]. Katana is a 6DOF small sized robotic manipulator, with a reach of 517mm and weighing 5.2kg.

DySim uses the Lagrange equation, considering the difference between kinetic and potential energy in the system to find the torque in each joint needed to make a given movement.

A. The Set of Trajectories

A small set of trajectories are selected to analyse on the robot. Bezier curves are used to make the movement, but with the additional constraint that the beginning and end of the trajectory have zero velocity and acceleration. This prevents the controllers from putting too much force on to the motors, making them stall or break.

A Bezier curve is defined by the following equation,

$$\mathbf{b}(t) = (1-t)^2\mathbf{p}_0 + 2(1-t)t\mathbf{p}_1 + t^2\mathbf{p}_2 \quad t \in [0, 1] \tag{13}$$

Then $\mathbf{b}(t)$ passes through the points $\mathbf{p}_0$ and $\mathbf{p}_2$ with curvature controlled by the point $\mathbf{p}_1$. To ensure that the trajectory terminates with zero velocity and acceleration a shaping function $f : \mathbb{R}^+ \to [0, 1]$ is used [Sahinkaya, 2001], defined by the following equations,

$$f(t) = 1 - e^{-(\alpha t)^3} \tag{14}$$
$$\dot{f}(t) = 3(\alpha t)^2 e^{-(\alpha t)^3} \tag{15}$$
$$\ddot{f}(t) = (6\alpha t - 9(\alpha t)^4)e^{-(\alpha t)^3} \tag{16}$$

The parameter α controls the magnitude of acceleration. Setting $\alpha := 1.66$ ensures that the movement stops near $t = 1$.

The Bezier curve is then modified by shaping the variable t, to give a new function $\Phi : \mathbb{R} \to \mathbb{R}^n$, defined by the following,

$$\Phi(t) = \mathbf{b}(f(t)) \tag{17}$$
$$\dot{\Phi}(t) = \dot{\mathbf{b}}(f(t))\dot{f}(t) \tag{18}$$
$$\ddot{\Phi}(t) = \ddot{f}(t)\dot{\mathbf{b}}(f(t)) + \ddot{\mathbf{b}}(f(t))\dot{f}^2(t) \tag{19}$$

Substituting (14) - (16) in to (17) - (19) gives a smooth function with the correct boundary conditions.

A set of trajectories is then defined as $\psi \in \Psi$ with $\psi = (\theta, l, \gamma)^T$, relating to the Bezier curve defined by $\mathbf{p}_0 = (0, 0.7, 0)^T$ the robots initial position, $\mathbf{p}_2 = l(\sin(\theta), \cos(\theta), 0)^T$ and $\mathbf{p}_1 = \frac{1}{2}\mathbf{p}_2 + \gamma\mathbf{p}_2^{\perp}$, where $\mathbf{p}_2^{\perp}$ is the vector perpendicular to $\mathbf{p}_2$. This constrains the manipulator to move in the plane, to simplify degrees of freedom in the model. With the parameter γ, giving the degree of freedom termed *curvature*. Figure (2) shows a Bezier curve along with the corresponding velocity and acceleration, demonstrating the zero terminal boundary conditions.

B. Energy Efficiency

We take the set of trajectories as defined in section III-A and calculate the energy required for each one. This gives a good measure to determine one trajectory from another.

The average energy in the k^{th} joint is defined as follows,

$$E_k(\psi) = \int_{t=0}^{\upsilon} |f_k(t)| dt \tag{20}$$

This is the integral of the absolute sample mean torques over time, where $f_k(t)$ denotes the torque in joint k at time $t \in [0, \upsilon)$. The total energy consumption for the action is taken to be the the sum of energy over each of the joints.

To test that the calculation of energy is correct, the same trajectories are performed on the Katana robot and the total current needed to make the movement is measured. Figure 4 shows the results of this experiment for a range of trajectories, the simulated results are similar to those from the dynamic simulation, validating the model.

C. Precision

Another measure of quality between trajectories is precision. A trajectory is called precise if adding noise to any of the motor torques affects the final position of the end-effector the least. This is useful for robots which do not use a high gain for position feedback, an example of this would be a torque controlled robot which is designed to be compliant.

To calculate precision, random Gaussian noise is added to the desired end position of the robot. The trajectory needed to reach this point is then calculated. This process is repeated 500 times, giving a sample set of the torques required to make the move in each joint. Counter intuitively, precise trajectories are chosen to be ones which have the lowest signal to noise ratio, ensuring that small perturbations in the motor torques will affect the end-position the least.

Two examples of this are shown in 3, with each colour representing a different motor. The bottom set of torque trajectories have a lower signal to noise ratio and are therefore more precise.

The signal to noise ratio for the k^{th} joint is then given by,

$$\rho_k(\psi) = \int_{t=0}^{\upsilon} \frac{\overline{\mu}_k(t)}{\overline{\sigma}_k(t)} dt \tag{21}$$

Where $\overline{\mu}_k(t)$ and $\overline{\sigma}_k(t)$ are the mean and variance of the torques in the k^{th} joint at time t. ψ denotes the parameters of the trajectory.

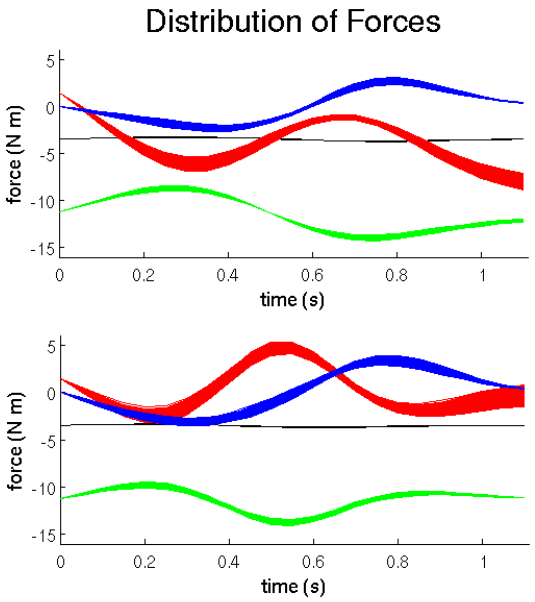

Fig. 3: The resulting torques calculated from a distribution of end-effector positions. The top graph shows a the torques needed to generate a Bezier curve with parameters $\theta = 0$, $l = 0.15$ and $\gamma = 0$. In the bottom graph the trajectory has parameters $\theta = -1.5$, $l = 0.07$ and $\gamma = -0.3$

This can then be used to estimate how the covariance of the end-effector position changes, given that there is noise present in the torques.

$$\Sigma_\psi \propto (\sum_{k=1}^{4} \rho_k(\psi))^{-1} I_3. \tag{22}$$

If a robot with high precision controllers are used, a small constant covariance matrix can be used instead.

IV. COMBINING VISION AND DYNAMICS

In this section, the arm dynamics and probabilistic models of the vision system are combined in order to perform hand-eye coordination.

Given the statistics for the arm α and object β being tracked, and the time passed $t \in \mathbb{R}^+$, a distribution over the set of trajectories can be found,

$$p(\psi|\alpha, \beta, t) \propto p(\alpha, \beta|\psi, t)p(\psi) \tag{23}$$

$$\propto \iint_{\mathbf{x}^b, \mathbf{x}^a} p(\alpha, \beta|\psi, \mathbf{x}^a, \mathbf{x}^b, t)p(\mathbf{x}^a, \mathbf{x}^b|t, \psi)p(\psi) \tag{24}$$

$$\propto \iint_{\mathbf{x}^b, \mathbf{x}^a} p(\alpha|\psi, \mathbf{x}^a, t)p(\beta|\mathbf{x}^b, t)p(\mathbf{x}^a, \mathbf{x}^b|t, \psi)p(\psi) \tag{25}$$

Where $\mathbf{x}^a$ and $\mathbf{x}^b$ denote the positions of the arm and object respectively, and ψ represents the parameters of the trajectory.

The Dirac-delta distribution is then used, so that a touch is only probable when the position of the object and position of the hand coincide.

$$p(\mathbf{x}^a, \mathbf{x}^b|t, \psi) = \delta(\mathbf{x}^a - \mathbf{x}^b) \tag{26}$$

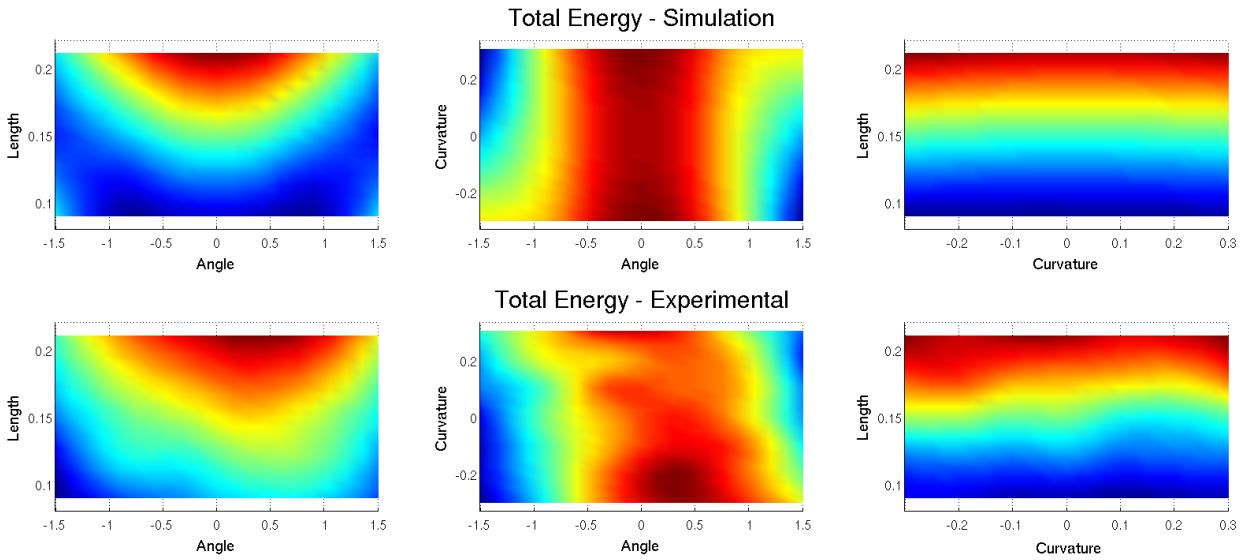

Fig. 4: The total energy required to perform a particular trajectory. The top row shows the results from a dynamic simulation, and the bottom row shows the results of measuring the total current used in the motors of the Katana.

Using Bayes theorem and equation (26) with equation (25) then gives that,

$$p(\psi|\alpha,\beta,t) \propto \int_{\mathbf{x}} p(\mathbf{x}|\alpha,t,\psi)p(\mathbf{x}|\beta,t)p(\psi) \qquad (27)$$

$p(\mathbf{x}|\beta,t)$ is the distribution of error in the visual estimate of the moving object, given in equation (12), with β defined at the end of section II-C; $p(\mathbf{x}|\alpha,t,\psi)$ is the distribution of error in the estimate of the arm position, taken to be normally distributed, with covariance as in equation (22) and a mean corresponding to the measured arm position at time t.

The distribution $p(\psi)$ is known as the prior, and gives the likelihood of a trajectory independent of any other information. The prior is set to be proportional to the total energy required to make the movement,

$$p(\psi) \propto (\sum_{k=1}^{4} E_k(\psi))^{-1} \qquad (28)$$

To find an action ψ the negative log-likelihood of (27), $-ln(p(\psi|\alpha,\beta,t)$ is optimised using the Gauss-Newton technique. The initial value is estimated by sampling the whole domain sparsely, and then choosing the maximum.

V. SIMULATIONS

The method is tested by dynamically simulating a Katana robot, and a linearly moving object in DySim. To ensure that the simulation environment is as accurate as possible, the same set of movements are performed on the robot as in simulation, and compare the total energy outputs using equation (20). The trajectories are defined by Bezier curves, as in section III-A, with the following defining parameters, $l \in [0.07, 0.21]$, $\theta \in [-1.5, 1.5]$, $\gamma \in [-0.3, 3]$. Sampled to give a total of 252 trajectories.

In order to compute the total energy for the robot, the trajectories were performed 10 times, and the current required in each of the joints recorded and averaged, this gives a value roughly proportional to the torque. This mean current is then used as $f_k(t)$ in equation (20), to calculate the total energy exerted by the robot throughout the movement. The results of this experiment are given in figure 4, showing a strong correlation between simulated and empirical results.

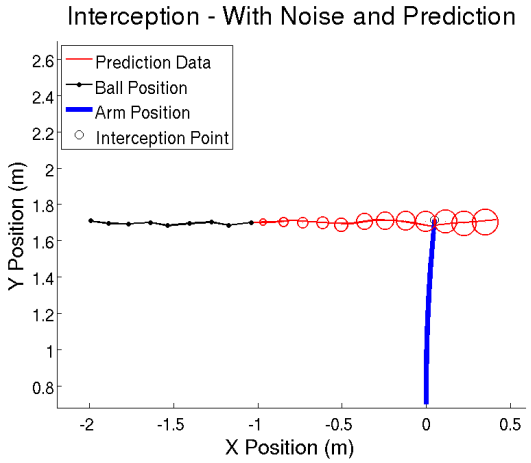

Fig. 5: Intercepting a moving target, the uncertainty in prediction is represented proportionally to the marker size.

The validation of the dynamics of the robot allow us to move on to use only simulation for experimentation. This simplifies the amount of technology needed and allows us to concentrate on the functionality of the method. We plan to extend these results in the future, using real cameras and a robot.

The ability of the robot to intercept the target is shown in figure 5. The method starts predicting when the trajectory becomes red, and the size of marker is proportional to the covariance as estimated by the Kalman filter. This demonstrates how uncertainty in the objects position increases with the amount of prediction. The robot is optimising for both energy

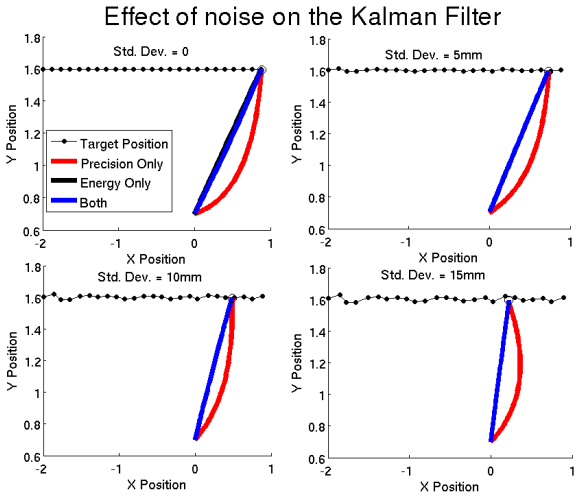

Fig. 6: Different actions the robot takes under varying noise and priors. In the final three graphs the trajectories for energy efficiency and combined energy and precision coincide, this shows a prior with a stronger preference for a trajectory dominating the other.

reduction and precision in the trajectory, and as a result, moves in a straight line towards the point of interception.

In order to show the effect of using different priors, the method is run and optimised for both energy minimisation and precision; energy minimisation alone with the end-effector covariance taken as 0.01; and precision alone with the prior $p(\psi)$ set to be uniformly distributed.

The magnitude of the noise is increased, and as the standard deviation increases the method tends to choose interception points earlier (figure 6). Comparing this with figure 5 it can be seen that as the variance increases, the robot starts to choose interception points earlier along the objects path, minimising the uncertainty in the Kalman Filter.

The response to noise was tested by adding Gaussian noise to the position of the moving object with a standard deviation between 0 and $3cm$, this was done for 50 trajectories and the average distance between the object and end-effector was consistently below $3mm$. This is just above the minimum tolerance of $2.5mm$, which is calculated as the average distance when zero noise is added to the objects position. As a result, if the target object has a diameter of $6cm$, then assuming that the tracker estimates its position within the boundary, the standard deviation will be less than $3cm$, allowing the object to be intercepted with a high probability.

VI. CONCLUSIONS AND FUTURE WORK

A. Conclusions

We have introduced a new method for combining robot dynamics and vision to perform hand-eye coordination, and shown it to be successful at intercepting a moving object.

Because of the Bayesian approach to integrating the different distributions and modelling the vision, the method is robust to noisy data when estimating the position of the object. It also allows a degree of flexibility to change prior information about the trajectories.

A restriction of the method is that it is open loop with respect to the cameras, so changing the calibration would affect the results. To remedy this, a visual feedback controller (such as defined in [Chaumette and Hutchinson, 2007]) can be used after a decision has been made on the correct trajectory to use. This is beyond the scope of this paper and will be integrated in future work.

B. Future Work

The experiments show that as the uncertainty increases the robot tends to chose trajectories that intercept the object early on. Determining the optimal time to start predicting is something that can't be calculated before making a decision. We plan to solve this problem using machine learning, enabling the robot to learn about the best prediction times incrementally, through intercepting the object several times.

It would also be possible to extend the method so that a variety of priors can be combined. Considering the torques in the motors, the prior can be used selectively depending on the application. For example, it may be important to avoid certain areas of the scene that are blocked by objects, or are out of reach for the robot. Trajectories that exert the maximum amount of torque may be needed in certain areas of the scene, in order to pick up an object, or push something out of the way.

REFERENCES

[Allen et al. 1993] P. K. Allen, A. Timcenko, B. Yoshimi, P. Michelman *Automated tracking and grasping of a moving object with a robotic hand-eye system* Robotics & Automation, 1993.

[Beaten and Schutter, 2004] J. Beatten and J. Schutter *Integrated visual servoing and force control: the task frame approach*, Springer Verlag, 2004.

[Capparella et al. 2005] F. Capparella, L. Freda, M. Malagnino, and G. Oriolo *Visual servoing of a Wheeled Mobile Robot for Intercepting a Moving Object* IROS, 2005.

[Chaumette and Hutchinson, 2006] F. Chaumette and S. Hutchinson *Visual servo control. I. Basic approaches* IEEE Robotics & Automation Magazine, 2006.

[Chaumette and Hutchinson, 2007] F. Chaumette and S. Hutchinson *Visual servo control. II. Advanced approaches* IEEE Robotics & Automation Magazine, 2007.

[Corke and Hutchinson, 2001] P. Corke and S. Hutchinson, *A new partitioned approach to image based visual servo control* Robotics & Automation, 2001.

[Guerrero et al. 2008] P. Guerrero, J. Ruiz-del-Solar, and G. Daz *Probabilistic Decision Making in Robot Soccer* Lecture Notes in Computer Science, 2008.

[Hong and Slotine, 1997] W. Hong, and J. J. Slotine *Experiments in Hand-Eye Coordination Using Active Vision*, Experimental Robotics, 1997.

[Lippiello et al. 2007] V. Lippiello, B. Siciliano, and L. Villani *A Position-Based Visual Impedance Control for Robot Manipulators* ICRA, 2007.

[Nelson et al. 1995] B.J. Nelson, J.D. Morrow, P.K. Khosla *Improved force control through visual servoing* ACC, 1995.

[Sahinkaya, 2004] M. N. Sahinkaya, *Inverse dynamic analysis of multi-physics systems.*, Journal of Systems & Control Engineering, 2004.

[Sahinkaya, 2001] M. N. Sahinkaya, *Input shaping for vibration free positioning of flexible systems*, Journal of Systems & Control Engineering, 2001.

[Santos and Ferreira, 2009] C. Santos and M. Ferreira *Timed trajectory generation using dynamical systems: Application to a Puma arm* Journal of Robotics & Autonomous Systems, 2009.

[Santos and Lima, 2009] J. Santos and P. Lima *Multi-Robot Cooperative Object Localization, Decentralized Bayesian Approach* RoboCup 2009.

[Zisserman, 2000] R. Hartley and A. Zisserman *Multiple View Geometry - Second Edition* Cambridge University Press, 2000.