Kinematic and 3D Motion Prediction for Gaze Control

Christopher Brown

Department of Computer Science
University of Rochester
Rochester, NY 14627

ABSTRACT

Estimation and prediction of the state of dynamical systems operating in three-dimensions, and of mechanisms acting in space, is one way for multiple-loop, multi-rate, time-delayed control systems to cooperate. Animate vision systems, biological or robotic, employ gaze control systems to acquire, fixate, and stabilize images. Our aim is to build robust gaze control behaviour from cooperating lower-level visual reflexes. Solutions are explored through simulation incorporating ten primitive gaze control capabilities, more or less comparable to subsystems in primate gaze and head control. Versions of several of the subsystems have been implemented on a binocular robot. Smith prediction is the basic paradigm, using kinematic simulation of the agent and optimal filtering to predict world state.

INTRODUCTION

Systems that behave in the world are becoming increasingly sophisticated, raising technical problems of sensing and control and opening new approaches that may make perception easier. One of the goals of artificial animate vision [1, 4, 5] is to exploit the ability to maneuver in 3D to make some vision problems easier. Another goal is to design a systems architecture in which multiple objectives (such as moving and observing) can proceed in parallel. One common premise is that cognitive processes at high abstraction levels rely on a hierarchy of lower-level "skills", "reflexes", "behaviors", and active vision capabilities that autonomously keep the agent out of trouble and perform generally useful vision behaviors. Another premise is that active control over and perception of an agent's own state (proprception) makes many problems in perception, planning, and acting easier.

The work described here is an investigation of a mechanism (based on prediction) for coordinating multiple actions based on perceptual input. The actions take place in the domain of gaze control. The main conclusions are that awareness of the characteristics and actions of other controlling behaviors is needed (as opposed to autonomous, mutually independent behaviors). The secondary conclusion is that 3D modeling and prediction are much more useful than 2D (retinal) modeling.

A gaze control system manages several basic, interacting head, eye, and even body motion capabilities, with the aim of supporting purposeful (or default) activity. One basic activity is the visual acquisition of an object. The action can be reflexive, in response to a stimulus deemed interesting, or under control of a higher level engaged in planning or acting. Another basic ability is to pursue or track an object moving relative to the observer: stabilization of the image on the sensor is necessary for high resolution imaging, and the resulting proprioception (i.e. motor commands that effected the tracking) provides information about object motion. From the point of view of control theory, a gaze control system has two main technical problems: the interaction of component subcontrols and delay. Both biological and robotic systems can easily have delays that are of the same order of magnitude as the timescale of the actions.

Gaze control mechanisms have long been studied in biological systems (there are extensive references in [14, 17, 22]). Much of the work concentrates on how biological systems solve the two basic technical problems mentioned above. This paper investigates predictive mechanisms as a solution for the problems primarily in a robotic context, but occasionally relates the results to some findings and theories from primate gaze control.

One way to cope with delays is to use strictly open loop control. The other approach, more common in engineering applications, is to use predictive and modeling techniques to anticipate the state of the plant, its input, and indeed the world [16], thus coping with both delays and interactions. Smith's principle [24, 25] is the basic tenet that the desired output from a controlled system with delay T is the same as that desired from the delay-free system, only delayed by T. The principle leads to several techniques for controlling delayed systems. Smith's principle may be coupled with signal synthesis adaptive control [3], which predicts object motion to allow more accurate responses. Kinematic and dynamic models for plant prediction can be known a priori or derived from learning and used to replace feedback [15]. The solution described here uses Smith prediction to integrate multiple controls with delays. It uses signal synthesis adaptive control with flexible and general techniques of kinematic simulation to predict the state of the plant and variance-minimizing optimal filtering to predict the state of the world. At least in sim-
ulation, the resulting predictions have three effects. **Delays are overcome, interactions are overcome, and performance is improved.** Predictive techniques seem to form a sound basis for the design of integrated, high-performance sensorimotor systems.

**THE ROCHESTER ROBOT**

Research in artificial animate vision has been made possible by recent technical advances in real-time computer vision and control. The area still includes a wide spectrum of problems from hardware design through software for parallel computation, including frame-rate image analysis hardware. Sometime the sensors are mounted on a moving cart.

Rochester's binocular robot head is mounted on a six-degree-of-freedom arm. The two cameras are on a common tilt platform, and have independent pan axes. The hardware is capable of motions comparable to primate performance (about 1 m/sec head velocity with less than 1 mm. positioning accuracy, and 300 degrees/sec camera rotations with 0.14 degree positioning accuracy). The camera controllers are capable of supporting full-speed gaze-shifts to random directions at a rate of 5/s. The cameras and robot controllers support several types of motion, but the frequency responses of system components under various forms of control have not been measured. The aperture, focal length, and focus of the cameras are not yet controllable. The video output is processed by a Datacube MaxVideo pipeline-parallel image processing system that can do many low and intermediate-level vision operations at 30Hz (video frame rate). The host computer for the system is currently a Sun/3 computer but will soon be a 24-node Butterfly Parallel Processor (BPP). Each node of the BPP is an M68020 processor with M68881 floating point coprocessor and 4MB of memory. There is a fast switch allowing the processors to share memory. The plan is to implement the control algorithms in the BPP, using multiple nodes as necessary for speed, with the Datacube furnishing real-time input. A real-time package for the Psychoe operating system will support the applications. LISP planning programs can communicate with robot applications code over the ethernet. Fig. 1 shows the current hardware organization.

**BASIC CONTROLS**

This paper deals, unless otherwise explicitly noted, with simulated capabilities and occasional psychophysical and physiological data. Early work [10] introduced five control systems, four of them resembling capabilities implemented on the Rochester Robot, and functionally reminiscent of primate capabilities of the same names.

**Saccade:** an eye control to produce rapid gaze shifts, “sampled” at a relatively coarse interval with respect to the discrete simulation time increment, and driven from positional retinal error (distance from retinal origin). At each activation it produced a sequence of maximum-speed pan and tilt velocity commands that was calculated from approximate kinematics to center the image in the retina.

**Smooth pursuit:** a proportional, integral, derivative (PID) eye control driven by the retinal positional error of the object's image, to follow objects in motion relative to the eyes. This and the following controls are "continuous" in that they are computed at every simulation time increment -- we use the quoted "continuous" and "sampled" in the sequel to make this distinction in a discrete (sampled-data) implementation.

**Vergence:** a PID eye control to reduce disparity between left and right images.

**Vestibulo-ocular reflex (VOR):** a proportional-gain eye control to oppose head motion with contrary eye rotation. Head rotation is easily canceled, and this capability also includes a version of "otolith-ocular reflex" that compensates for translational head motions using information about object range [8, 13]. Its input is not sensory, but an "afferent copy" of the head translation and rotation command.

**Head compensation:** a proportional-gain head-control system driven by head-relative eye position that rotates the head in the direction of eye rotation to keep the eyes centered and away from their mechanical stops. This capability may not mirror any named biological one, but it does reflect a widespread primate tendency to move the head, if allowed, during pursuit and eye saccades.

Fig. 2 illustrates the cumulative effect of simply superimposing control capabilities: each operates independently and
their outputs are simply summed at the effectors. Delays are zero, latencies (except for saccades) unity. Fig. 3 shows the dramatic effects of control delay on the system. The smallest delays, applied uniformly or to just one control, destabilize the system seriously.

**ENHANCED CONTROLS AND 3D MODELS**

In later work some improvements were made to the controls. This work is more fully described in [11]. All controls originally operated from retinal coordinates. However, predictions of object position in head or LAB (laboratory) coordinates are needed to predict retinal images. Head rotations, pans, and tilts all induce camera origin translations due to the geometry of the head, and non-retinal representations are more robust (as when the object temporarily is lost). *Versions of pursuit, vergence, and eye saccadic capabilities were added that use 3D spatial information.* There was no capability for estimating the state of objects moving in LAB (relative motion was produced with a static object and observer motion.) *A pipeline of object state descriptions is maintained using variance-minimizing filters to predict object state from observations.* The head compensation reflex was the only head control. *The system was given "head saccade" and "head pursuit" controls.* The eye saccade control algorithm was unsophisticated, and there was no significant head and eye cooperation for quick gaze shifts. *Four fast gaze-shift algorithms involving both head and eyes were investigated.* The two modes of operation (pursuit and saccadic) were simplisticly assumed to correspond to inflexible combinations of lower level capabilities. *All controls can now be activated and deactivated independently.*

The later work is not committed on several issues, such as whether the eye saccadic system should control one or both eyes. Bimocular eye saccades are easy to implement, but there are many other ways that the eyes can cooperate during gaze shifts, depending on technical considerations such as the possibility of visual computation during the shift, how the vergence control is specified, etc. We likewise shall not address issues of orchestrating the smooth transition between saccadic and pursuit (or other) tasks. Considerable work is still needed on the topic of smooth blending of subcontrols, which forms the foundation of motor skills.

**DEALING WITH DELAYS**

The Smith predictor incorporating the object state predictions is shown in Fig. 4. Its derivation appears, for instance, in [10, 16]. The basic idea is to have a zero-delay feedback (path C) based on simulation. The model-delayed simulation data (path B) is compared with the actually-delayed data from the plant (path A); the difference (at D) is zero for perfect simulation, so it provides information about the simulation adequacy and (if slowly varying compared to the control delays) can compensate for inaccurate modeling. A more satisfactory approach (not yet implemented) is parameter adaptive control: compute sensitivity derivatives and use

Figure 2: Increasingly effective delay-free control results from superposition of noninteracting controllers. (a) Tracking only: The left (dominant) eye pans and tilts, inducing tilt in the right eye. The tracker uses a position error signal. The right eye gets no pan signal, and its horizontal error accrues from target motion. The left eye tracks successfully until it hits mechanical stop at tick 14. (b) Add vergence and head compensation: This control is to keep eyes from hitting mechanical stops by turning the head in the same direction as the tracking motion. A less-desirable effect is to amplify the tracking signal, overcompensating and destabilize the tracking. (c) Add VOR, which effectively compensates the head rotation with eye rotations.
the signals from the plant (path A) and the difference between predicted and actual behavior (path D) to adapt the model. This is the dual problem of model-reference adaptive control, which forces the system to conform to the model.

The work described here uses the following interacting controls algorithm [10], which assumes each controller knows its own delay $T$, and the delays of all the other controllers in the set $S$ that share an output with it. Look ahead the maximum delay $M$ of any controller in $S$ and retrieve the predicted robot and control states for that time. Apply the control appropriate for these future states at (possibly future) time $M-T$.

In early work, only the kinematic simulation existed, and so in dynamic vision situations, prediction could only be accurate if the simulated robot moved while the object remained stationary. Fig. 5 shows how the interacting controls algorithm successfully stabilizes performance (compare Fig. fig:nodel4) even when several subcontrols are at work, all with different delays. Prediction fails if knowledge of the world is inaccurate: Fig. 6 illustrates the need for prediction of 3-D object motion. The need is addressed with dynamic models and optimal filter prediction for moving objects.

OBJECT PREDICTION IN 3D

In contrast with the explicit kinematic simulation used to predict the system state, standard optimal (i.e. variance-minimizing) filtering techniques are used to predict the position and velocity of the world object. The simulation has been run with extended Kalman filters, (linear) Kalman filters, and time-invariant filters as predictors. It is standard practice with optimal filtering to use statistical techniques to see if the current dynamic model fits the data, and if not to substitute another model [6, 12]. This "variable dimension" approach is the predictive filtering equivalent of the signal synthesis adaptive control scheme [3, 2], and the block diagrams of the two systems are basically the same. In principle the system can cope with arbitrarily complex object motions as long as they are predictable, either from estimation techniques (see [6] for techniques of tracking multiple maneuvering objects in clutter) or recognized or constructed from a library of motions or motion primitives [21].

Figure 3: The no-delay controller applied with zero delay in all controls except tracking, which has a delay of one tick. Ideally this graph should be a delayed version of Fig. 2(c).

Figure 4: The Smith predictor control. The CONTROL block represents all control systems, and DELAY their independent delays. $K_1$ and $K_2$ are gains to weight the delayed and non-delayed modeled error signals. The $\alpha - \beta$ filters estimate object and image states, and adaptively synthesize the signals for the control system. Dotted boxes show the parameter adaptive control to update model in response to differences with the plant (not yet implemented).

Figure 5: To be compared with Figs. 2(c) and 3. The interacting control algorithm dealing successfully with a mixed set of delays.

Figure 6: With prediction based only on the robot's motions assuming a stationary object, actual object motion leads to mis-prediction and destabilization. Here the object is moving toward the robot. As it gets close its retinal velocity "unpredictably" increases and the controls cannot respond rapidly enough.
For this work, we assume what is called a constant velocity model (although it is good for maneuvering objects as well); starting with some initial value, the object's velocity in LAB evolves through time by process noise of random accelerations, constant during each sampling interval but independent. The cumulative result of the accelerations can in fact change the object's velocity arbitrarily much, so we model a maneuvering object as one with high process noise. For this work we assume position measurements only are available, subject to measurement noise of constant covariance.

Assume the object state (its position and velocity) evolves independently in each of the \((X, Y, Z)\) dimensions. For instance, in the \(Y\) dimension, it evolves according to

\[
y(k+1) = F_y y(k) + v(k),
\]

where

\[
F_y = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}
\]

for sampling interval \(\Delta t\), and \(y = [Y, Y']^T\). The equations for the other two spatial dimensions are similar, and in fact have identical \(F\) matrices. Thus for the complete object state \(x = [X, \dot{X}, Y, \dot{Y}, Z, \dot{Z}]^T\), \(F\) is a \((6 \times 6)\) block-diagonal matrix whose blocks are identical to \(F_y\). The error vector \(v(k)\) obeys

\[
E(v(k) v^T(j)) = Q \delta_{k,j}.
\]

The \(\alpha - \beta\) filter for state prediction has the form

\[
\hat{x}(k+1|k+1) = \hat{x}(k+1|k) + \left( \frac{\alpha}{\beta/\Delta t} \right) [z(k+1) - \hat{z}(k+1|k)],
\]

where \(\hat{x}(k+1|k+1)\) is an updated estimate of \(x\) given \(z(k+1)\), the measurement at time \(k+1\). Here we assume that \(z(k+1)\) consists of the three state components \((X, Y, Z)\) (but not \((\dot{X}, \dot{Y}, \dot{Z})\)). The state estimate is a weighted sum of a state \(\hat{x}(k+1|k)\) predicted from the last estimate to be \(F \hat{x}(k|k)\) and the innovation, or difference between a predicted measurement and the actual measurement. The predicted measurement \(\hat{z}(k+1|k)\) is produced by applying (here a trivial) measurement function to the predicted state.

The \(\alpha - \beta\) filter is a special case of the Kalman filter. For our assumptions, the optimal values of \(\alpha\) and \(\beta\) can be derived (see [6], for example) and depend only on the ratio of the process noise standard deviation and the measurement noise standard deviation. This ratio is called the object's maneuvering index \(\lambda\). The filter is initialized with a velocity given by the difference of the first two position measurements. The sensor and process noise are assumed known, though the latter could be estimated with more initialization effort. The constant covariances assumed by time-invariant filters may suffice for sensors stationary in LAB, but a head-mounted depth sensor with the usual highly eccentric depth error covariance ellipsoid would very likely require the full power of a Kalman filter.

The incorporation of an optimal filter into the control loop was motivated by the necessity of predicting object state for the Smith predictor. The filter, however, also has a noticeable and beneficial effect on pursuit performance (Fig. 7), and indeed it seems that humans may use stochastic prediction [18, 20].

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{figure7.png}
\caption{The angular pointing error of the left eye while pursuing a moving object (initially off center in the visual field), measured to the true object position. Light squares: results with output of the \(\alpha - \beta\) filter.}
\end{figure}

\section*{EYE-HEAD SACCADES}

This section describes four versions of cooperating eye-head motions. A first example of an eye-head saccade is provided by simply running the "sampled" eye saccades simultaneously with the "continuous" VOR and head-compensation reflexes ([10]). The result (Fig. 8(a)) is similar to that which once was claimed (under the "linear summation hypothesis") to hold in primates, viz. that their eye rotational velocity measured in LAB is kept constant (eye saccadic velocity is decreased by head saccadic velocity) [19]. However, in primates, the linear summation hypothesis has been contradicted by findings indicating eye-head gaze shifts outperform eye saccades alone (Fig. 8(b)) [23]. Certainly in a robotic context a speed improvement is attainable, and the following three versions are intended to outperform the first one, and are more fully described in [11].

The \textit{Gaze Feedback} version is inspired by "local feedback" physiological theories, and is based on "continuous" head and eye position feedback controls. A head saccade is simply implemented as a pursuit operation, and cooperating eye-head gaze-shifts are simultaneous head and eye pursuit. For various reasons this version fails to achieve a speed improvement. The \textit{Sampled Optimal} version is a closed-form "sampled" solution for full-speed eye and head movements. It depends on an open-loop calculation depending on known constant head velocities, and so is not robust against variations or disturbances.

The \textit{Simulated Optimal} version uses the familiar pipeline of predicted future states to maintain optimal speed and add the "continuous" ability to deal with arbitrary velocity profiles as well as disturbances. The algorithm first computes a head saccade by determining, for the \(z\) and \(y\) dimensions, the direction to rotate and then driving at maximum velocity, stopping after there is a zero-crossing in predicted object position (say at time \(k\)). The head position at time \(k+1\) may be closer to the object: the closest position is chosen. The sequence of head-saccade commands is inserted into the pipeline and the resulting changes to head and eye position
computed (these values overwrite existing values, as the sac- cade is taken to replace previous control). Then the eye sac- cade is computed exactly like the head saccade, except eye position is computed by adding the head-relative eye veloc- ity to the head velocity. To cope with disturbances, and to implement successive ("catch up") saccades, the simulation is placed in a loop that runs until the saccade is successfully completed, checking if at any time a new calculation must be performed. Fig. 9(a) shows a result from this control scheme. After reaching the object, the eye continues tracking the moving object. When macaque monkeys have their eye velocities electrically disturbed after the target stimulus has vanished, they successfully correct the saccade (Fig. 9(b)). The recal- culation can sometimes be done during the latency period of the original saccade, so the correction does not delay saccade onset [17]. In the simulated system the control delay is un- avoidable, but correction occurs with both eye and eye-head saccades.

**CONCLUSIONS AND CURRENT STATE**

Gaze control is a much-studied complex sensorimotor abil- ity. In gaze control, several simple "reflexes" or "behaviors" cooperate to achieve effects such as gaze shifts, object track- ing, and gaze stabilization. Attempting to implement (in simulation) some simple aspects of gaze control led to several conclusions.

The behaviors acted like a multi-rate, multi-input, multi- output, time-delayed control system. Communication be- tween behaviors and mutual knowledge of individual behavior characteristics was needed to let them cooperate effectively (stably, accurately) in the presence of time delays and control interactions. The communication in this case is through a shared data structure that predicts the future state of the active agent and of the world. The mutual knowledge is knowledge of time delay characteristics. The resulting control is a form of the Smith predictor. The world models are more effective when they are in terms of 3D space, not reti- nal coordinates. Dynamical systems estimation of more or less complexity is an obvious choice for certain sorts of world modeling (e.g. for moving objects), and in particular Kalman and time-invariant (a – p) filters were applied to the prob- lem. Such filters might also be useful to predict the state of the agent, but in this work the knowledge of agent's state was assumed to be accurate enough to allow 3D kinematic simulation.

The simulation program was meant to be a testbed for styles of interaction rather than a quantified tool to predict actual robotic performance. Current work is aimed at quan- tifying the behavior of the Rochester Robot. After that the role of the simulator will be reassessed. For example, with the VAL control system it may be impossible to tell the robot's state during motions accurately enough to stabilize gaze by counteracting head motion with eye motion. Among other things we are investigating a neural net to learn the length of time it takes to move from one point in 3D space to another. Work is also proceeding to speed up the communication be- tween the host and the Puma robot, and to speed up certain low level vision computations on the MaxVideo system. Our next goal is to implement some simple cooperating controls (gaze stabilization, head-eye saccades). Some of the simu-
lator data structures and algorithms may be useful in this effort.

References


