Proceedings of the 1993 IEEE/RSJ International Conference on Intelligent Robots and Systems, July, 26-30 1993, pp. 1173-80

THE USC AUTONOMOUS FLYING VEHICLE : AN EXPERIMENT IN REAL-TIME BEHAVIOR-BASED CONTROL

Andrew H. Fagg [*] ahfagg@mensa.usc.edu

M. Anthony Lewis [‡] lewis@mensa.usc.edu

James F. Montgomery [*] monty@mensa.usc.edu

George A. Bekey [*‡] bekey@pollux.usc.edu

Center for Neural Engineering and
Institute for Robotics and Intelligent Systems
University of Southern California
Los Angeles, CA 90089-0781

Abstract

A control system architecture is described for an autonomous flying vehicle. The vehicle, equipped with fourteen sensors, uses a model helicopter as an airframe. The control system utilizes these sensors to a) remain aloft and in stable flight, b) navigate to a target and c) manipulate a physical object.

The overall approach to the problem is based on a behavioral paradigm. The key contribution of this paper is the demonstration of a situated agent under these severe circumstances; as the craft is airborne, it is in constant risk of crashing. Unlike terrestrial mobile robots, the craft must constantly make sound decisions to maintain its integrity.

1. Introduction

Historically, robots have been used in structured, static environments. Because of this, many solutions that have been developed for robot control have been notoriously fragile when extended to unstructured, dynamic environments.

One technique for dealing with this problem has been the use of behavioral-based or task achieving architectures. The success of this approach undoubtedly has been because many complex tasks can be implemented with a collection of simple, interacting behaviors. Each behavior can be computed in parallel with the other behaviors, and each behavior can extract from the environment just the information that is needed to perform its task.

Brooks has stated [1] that this style of computation can break the Von Neuman bottleneck that throttles computation of information-rich sensory data. He feels that his approach is scalable to the level of higher cognitive functions. Is it unequivocal that this approach is truly scalable? A solution derived from a mathematical analysis seems to be far in the future. Therefore, we must rely on empirical experimentation to demonstrate the limit, if a limit exists, of this approach.

To address this problem, we have begun to experiment with an autonomous flying vehicle based on a behavioral architecture. The robot has been designed to compete in a yearly contest held at Georgia Institute of Technology, sponsored by the Association for Unmanned Vehicle Systems.

1.1 The Aerial Robotics Competition Task

The Aerial Robotics Competition has been designed to promote the development of autonomous flying robots by combined student and industrial teams competing in a very difficult robotic task.

The competition arena is a 60 by 120 foot area (green Astroturf ®), in which two black-colored, six-foot diameter rings are located at either end (Figure 1). A three-foot barrier runs through the center of the arena, separating the two rings. Within one of the rings, six day-glow orange disks are randomly placed.

The task of the robot is to navigate to the source ring, locate and pick up one of the disks, and deposit it into the other ring. This process is then repeated once for each of the remaining disks. The task must be completed within six minutes. Although the craft may land within either of the two rings, it is not allowed to touch the ground or make use of ground effect at any other point in the arena.

The robots must be completely autonomous, in that no human intervention is allowed. However, off-board computation may be employed, with a data-link between the robot and computer. In addition, sensors or navigational systems may be mounted outside the arena to aid the craft in its task.

1.2 Assumptions and Constraints

Our goal is to develop a fully autonomous craft based on a behavioral paradigm. This compelled us to place additional constraints on our solution beyond those of the competition rules.

- *Other than those provided in the competition arena, the craft must not make use of externally engineered cues.* For example, we have rejected the use of beacons as a way of locating the craft within the play-field.

- *All sensing and processing must be performed on-board the craft.* Any data-link would be only uni-directional, from the craft to the operator, as an indicator of system status.

- *A central representation of the craft and task* ***is not*** *necessary to accomplish the task. In fact, such a representation may*

‡ Department of Electrical Engineering

* Department of Computer Science

create information processing bottlenecks.

- *The information processing system would have to make use of vision (the only sensor that provides a rich enough representation of the environment), because of the degree of uncertainty in the results of executed actions.*

1.3 Challenges

This domain provides a set of very interesting challenges, each of which must be met as more autonomous robots are constructed.

- *Good control decisions must be made at all times.* In most domains, including most rolling and walking robots, it is possible to pause the motion of the machine as the computing system attempts to make a decision; it is also possible to make a sequence of bad decisions without endangering the integrity of the robot. This is not the case in this domain. At the very least, approximately correct control decisions must be made in a timely manner, with possible corrections being made a short time later by a higher-level system.

- *The system must be robust to an uncertain environment, as well as to imperfect sensing capabilities.* The sensors used onboard are inexpensive devices that yield rough estimates of various parameters. The uncertainties may have strong biases, and cannot be simply integrated to yield precise heading and position information. What this means practically is that a combination of dead reckoning and landmark piloting must be used. The latter behavior demands a heavy reliance on vision.

- *Not all problems can be anticipated ahead of time.* Therefore, the control system must be at least robust enough to ensure the safety of the craft when unexpected situations occur. An example of this is being unable to lock onto a visual target.

- *Helicopter dynamics are notoriously complex.* By way of example: Forward flight creates a moment in the craft that can cause rolling. Increasing the main rotor pitch or RPM (to climb) can induce a torque about the yaw axis. The elevator cyclic is coupled not only to pitch but to roll by the angular momentum of the rotor blade. In addition, ground effects can create chaotic airflow near the craft. Clearly, helicopter control is a difficult control problem.

2. Previous Approaches

Previous approaches to autonomous control of flying robots have included the use of non-linear controllers [2]. Prior to the work of Heigers *et al,* controllers for helicopter flight were comprised of a collection of linearized controllers around several trim conditions. Their approach addressed the problem of creating a unified control structure for various trim conditions.

Cheng and Lam [3] have discussed the problem of obstacle avoidance in the forward flight of a helicopter. The key issue that they addressed was the selection of a pathway for nap-of-the-earth (NOE) flying. In their approach, the control algorithm for stabilizing and controlling forward flying of the helicopter was synthesized using non-linear control theory. A higher level guidance structure was added to this which used forward looking sensors to detect terrain obstacles and to plan paths around them.

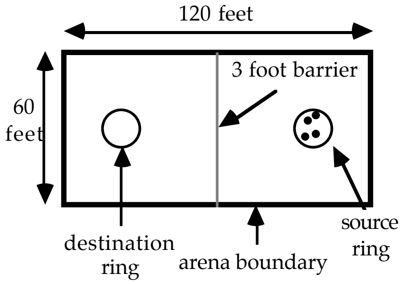

Figure 1 : The competition arena.

At Purdue, Ahmad [4] has reported the development of a neural network controlled helicopter. One key feature has been the rapid, in-flight learning of control parameters. At MIT, Atkeson [5] is pursuing an approach that uses an external vision system to track specially marked helicopters. At the University of Michigan, Borenstein [6] is pursuing an approach that involves a "HoverBot" robot that uses 4 model-helicopter rotor heads. He uses onboard accelerometers for sensing. It has also been reported that Sugeno in Japan has created a voice directed, fuzzy logic controller model helicopter [7]. Apparently he has used several linear, hand designed controllers.

The key distinction between the approach pursued by others and the one presented here is that our goal is to create a unified framework for dealing with the problems of control, navigation and the sequencing of behavior. In addition, we are restricting ourselves to onboard processing and sensing.

3. A Biologically Inspired Approach to Autonomy

Robots designed by control engineers usually depend on what we term the *traditional control paradigm*: they contain a controller which is designed using models of the *plant*, often in a state-space formulation. Robots designed by AI researchers have traditionally been based on planning paradigms, requiring that the system have complex internal representations of an *objective* external world, actions, goals, and events, which are used for reasoning about actions to be taken.

By contrast, living systems have evolved patterns of behavior which require neither explicit models for control nor detailed planning before they can act. Invertebrates and lower vertebrates appear to be capable of a number of simple behaviors; complex behavior arises from an interaction among these simpler behaviors. The evolution of biological systems is probably facilitated by this decomposition, since reactions to different stimuli and the associated simple behaviors can evolve in parallel. At a higher cognitive level, it has been realized that perception is an active process. It involves the active seeking of information, modification of cognitive models and the use of the modified models to seek additional information in the world[8]. Thus at both the lower level of motor interaction with the world and at a higher cognitive interaction, organisms are highly selective in the information that they extract and use from the world. From this perspective, the idea of an objective inner reality makes little sense.

As a result of these observations, a number of investigators have postulated alternative approaches to the design of robot control systems. This view of behavior organization in insects and in vertebrates has led to the development of architectures such as "behavior-based" or "reactive" control, "situated agents", or "subsumption", which have been used successfully by Brooks [9, 10], Maes [11], Payton[12, 13], Arkin ,[14] and others in the design and fabrication of a number of autonomous robots.

It should be noted that behavior based control does not necessarily preclude all forms of planning. For example, Maes [11] points out that action selection can be based on some form of "planning", such as looking ahead or evaluating alternative goals, while preserving the task orientation of behavior based systems and avoiding the need for complete remapping of the world and replanning at every decision point. Payton [12] indicates that plans can be internalized and used to enhance performance, without giving up the ability to react to sensory stimuli. Similarly, Agre and Chapman [15] have suggested that plans can be used as resources in deciding on actions, rather than as rigid formulas for execution. Arkin [14] proposed an architecture which combines world models with behavioral approaches in the generation of actions for robot navigation. In addition, Sutton and Barto [16, 17] have shown how a planning system can instantiate a reactive policy to handle real-time constraints, and how optimal policies may be learned.

The view that simple behaviors are composits of reactions to stimuli has also been termed "reflex control" and used as a basis for the design of robot controllers by Bekey and Tomovic [18]. In their approach, such complex behaviors as human walking or grasping can be decomposed into sequences of simple behaviors triggered by sensory events. Arbib [19] has further developed this approach by representing specific units of sensory or motor behaviors at a level intermediate between overall tasks and neural nets (schemas), and postulating their location in the brain. Arbib's view of the relation of perception and action is also related to Neisser's action perception cycle [8].

While biological systems have evolved their behaviors over millennia, robot behaviors must be postulated from observations of behavior, and evolution can only be simulated. Nevertheless, the fundamental hypotheses which guide our design of autonomous robotic systems are: a) that complex patterns of behavior can be obtained as composites of simpler, individual behaviors, and b) that *central* representations (models) of the system and task are not required to accomplish the goal.

4. The USC Robotics Lab Approach

An overview of the control architecture is shown in Figure 2. At the lowest level, the behaviors implement very tight *reflex* loops. The task of each individual loop is very simple. For example, the roll-reflex tries to maintain a zero roll rate and a zero roll orientation. The combination of these behaviors is responsible for maintaining the overall stability of the craft, including attitude, height, and heading.

The mid-level behaviors assume that the lower level behaviors are behaving with some degree of competence. These behaviors differ from the low-level reflexes in two ways. First of all, they do not directly affect the outputs to the actuators, but instead modulate the low-level behaviors. Secondly, the functions that these behaviors implement vary in time.

The next higher level is responsible for achieving the goal at hand. The goals will be one of the following: a) hovering over a target, b) moving towards a target, or c) searching in place for a target. The *moving towards a target* behavior makes use of an egocentric map to keep track of the estimated position of the target relative to the craft. An egocentric coordinate system (representation of target location with respect to the craft) facilitates the mapping of expectancies from the map onto the retinal plane and vice versa.

This estimated positional information is used to direct the gaze of the camera. Once the image processing system begins to detect the target location to a high degree of certainty, this information may be used to improve the positional estimates stored in the egocentric map.

The highest level of the system, the sequencer, is responsible for generating a set of subgoals to accomplish the entire task. This is done by activating the appropriate set of behaviors, and instantiating the correct set of parameters. When one of the activated behaviors is *move towards a location*, the initial location of the goal (in the egocentric map) is instantiated by looking up the goal's position relative to the craft's current position in the allocentric map (a map of all relevant objects in a global coordinate frame). Likewise, when a positional goal is achieved, the craft's position in the allocentric map is updated.

It may be questioned whether a sequencer should be used or a more internalized plan should be used. In our minds, it is clear that at higher levels of cognitive processing we can purposefully create sequential plans. A more Brooksian approach would be to allow the emergence of such a plan generator through the interaction of several behaviors and the external environment.

4.1 System Components

In this section, the details of the craft's mechanical, sensing, and computing are discussed.

The airframe itself is a Kyosho Concept 60 Helicopter. The dry weight of the craft is about 10 lbs. The main blades have a span of approximately five feet, and the overall length of the craft from the tip of the tail boom to the forward tip of the main rotor blades is about six feet.

The craft is powered by an Enya 80, 2 stroke, nitromethane-powered engine. It carries about 10 oz of fuel, and has a flight duration between 10 and 15 minutes.

The craft has five controllable degrees of freedom. These are: the engine throttle, the tail rotor pitch, the collective pitch, the elevator main rotor cyclic, and the aileron main rotor cyclic.

The craft currently utilizes a total of fourteen sensors.

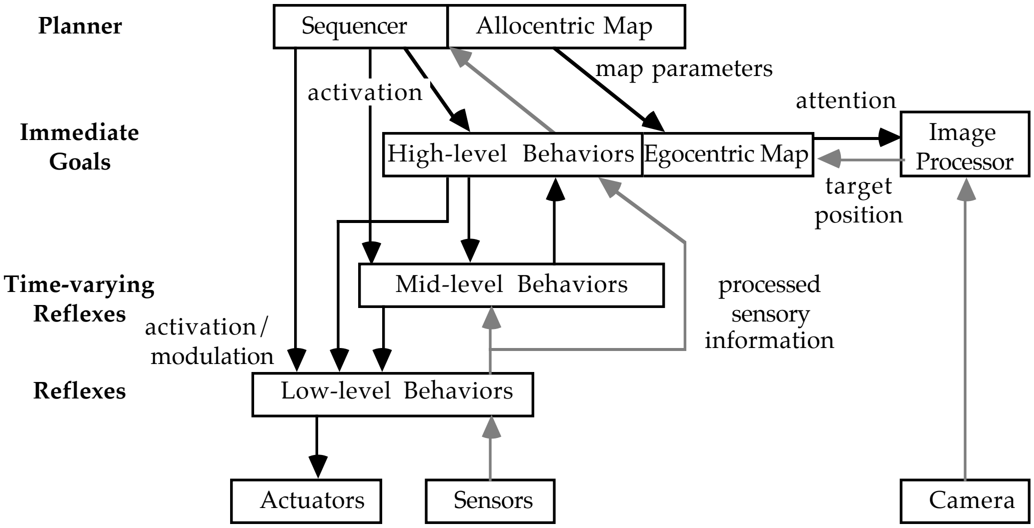

Figure 2 : The system control architecture.

Three downward-looking ultrasonic sensors are used to estimate the height and attitude (roll and pitch) of the craft. Inclinometers provide redundant attitude information. Three rate gyroscopes are mounted on board to give estimates of the rotational rate of the craft and are used to dampen the rotation about each axis. A flux gate compass is used to give heading information. Images are obtained from a Sony XC-999 CCD camera. The camera is fitted with a lens having a visual angle of 96^{o}. A custom-designed RPM sensor has been attached to the main mast to give an estimate of engine RPM. Finally, accelerometers are used to sense lateral movements of the craft.

In addition, several more devices are planned and will be incorporated into the control algorithms in the near future. The orientation of the camera is controlled by a mechanical saccade system, which allows changes in the yaw and pitch of the camera's image plane. Tactile sensors will be placed on the landing gear to detect when the craft has landed.

The on-board processing is performed by two Motorola 68332 microcontrollers. One processor is dedicated to image processing. Using a subsampled image, we are able to perform image processing at about 10 hz. An image processing system based on the Texas Instruments TMS320C30 DSP chip is being developed in order to drastically improve the image processing capabilities of the system.

The subsampling strategy uses a retinal sampling pattern. A total of about 2K pixel samples are made on each image. The sampling of a small amount of data significantly decreases the memory and processor requirements, and hence energy requirements.

4.2 The Control Architecture

This section describes a more detailed view of the control architecture. In Figure 3, most of the individual behaviors are shown, as well as the primary informational links.

Low-level Behaviors. The low-level reflexive system consists of four behaviors:

- *Attitude control* attempts to maintain the craft in a horizontal position. The input is an estimate of the craft's attitude, which is derived from the three downward-looking sonars and the inclinometers. The output of this behavior drives the servos for the elevator main rotor cyclic (pitch) and the aileron main rotor cyclic (roll). In addition, this behavior may be modulated by external inputs, forcing the craft to tilt in one direction.
- The *heading control* behavior maintains a desired directional heading. The input is derived from the compass, and the output controls the state of the tail rotor pitch.
- The *thrust control* behavior attempts to maintain a constant height above the ground. The height estimate is derived from the sonars, and the output drives the engine throttle and collective pitch.
- The *grasp* behavior is responsible for acquiring a disk once the craft has reached the source ring.

Mid-level Behaviors. These behaviors affect the system by modulating the low-level reflexes. As noted earlier, these modules generate time-varying input in response to being activated.

- The *lateral motion* behavior causes the craft to pitch in a specified direction and length of time. The output generates sinusoidal modulatory signal that is input to the *attitude control* behavior.
- The *transition to height* behavior produces a smooth transition from the current height position to a specified one. This behavior is especially critical for landing and taking off.

Immediate Goals. A set of six behaviors are used to obtain a variety of possible subgoals.

- The *egocentric target position* behavior maintains an egocentric estimate of the target position, including the confidence associated with the estimate. This information is updated with information from several sources. The target position is initially instantiated from the allocentric map when the craft is sitting on the ground, and the craft's position is known. This estimate is then updated in flight using motor afferent information from the *lateral motion* behavior, the information from the *vestibular system*, the current craft configuration (attitude, height, heading, and saccade state), and visual information from the *image processor*.

When no visual information is available, the positional estimate depends only on the motor afferent, craft configuration, and vestibular information. This causes a decrease in the positional estimate's confidence, since no direct sensory information is available to confirm that the motor commands are being executed as expected. However, once the visual target has been acquired, the estimated position can be updated with more accurate information, and the estimate's confidence increased.

- The *expected retinal target position* behavior translates the egocentric estimate of the target's position and the current system configuration into an expectation of where in the image the target should be seen in retinal coordinates.

- The *image processor* behavior attempts to locate the visual target in retinal space. The retinal-based estimate of target position is used to direct the module's processing time towards the area that is more likely to contain the visual target. In addition, this behavior commands the configuration of the saccade system, with the goal of keeping the target (or expected location of the target) within the center of the visual field.

- The *move toward target* behavior orients the craft in the direction of the estimated target position and then commands a forward movement. These actions are taken regardless of the source of the positional estimate (dead reckoning or visual information).

- The *hover over target* behavior is activated when the craft has moved over the target and must continue to maintain a stationary lateral position. This module relies on constant visual information to maintain a high-quality positional estimate.

- The *stationary search* behavior is activated when the craft believes that it is hovering over the target location, but is unable to visually locate the target. This behavior commands the craft to slowly turn in circles (yaw), while simultaneously increasing the altitude of the craft.

The Planner. The *sequencer* module is responsible for producing the set of actions that achieve the high-level goal of obtaining a disk and transporting it to the destination circle. For each stage in the plan, the appropriate set of behaviors are activated. The sequencer then waits for the appropriate termination conditions before a new set of behaviors are activated.

For example, moving from the source circle to the destination circle employs the following sequence of behaviors :

1. Lift off and hover above the source circle: *activates transition to height* and *hover over target* behaviors.

2. Fly to destination circle: instantiate the position of the destination circle in the egocentric map, and activate the *move toward target* behavior.

3. If the craft reaches the estimated position of the target, and the target is not yet visually located, then activate the *stationary search* behavior.

4. If the craft is at the target and it has been visually located, then land: activate *hover over target* and *transition to height*.

5. Update position in the allocentric map.

5. Current System Status

Currently, the lower and mid-level behaviors have undergone testing. *Attitude control*, *heading control*, *lateral motion*, *transition to height*, and *thrust control* (hover at altitude) have been demonstrated both on a trainer apparatus and in free flight. In addition, the integrated behavior of the *attitude control*, *heading control*, and *thrust control* behaviors has been demonstrated. The height of flight has ranged from six inches to about three feet.

During the flights, the following observations were made: Firstly, the overall control scheme seemed to be adequate to maintain stability, even when mild winds were present. Attitude control is significantly enhanced by the use of rate gyroscopes on the roll and pitch axes.

Secondly, the use of ultrasound as input to the attitude control behavior is inadequate for flight above all but very level surface. For example if the surface was inclined, a strategy of maintaining perpendicular to the surface would lead to a pronounced side slip. One possible way to approach this is through the use of inclinometers. However, when the craft undergoes acceleration, the apparent *down* direction is the vector sum of the gravitational acceleration and the acceleration of the craft. This gives a false reading of *down*.

It is felt the problem of *down* must be approached by the use of several behaviors using different information sources. This approach could also include the use of visual information to determine a horizon.

The immediate goal is to integrate the vision system into the coordinated behavior of the craft. Vision is critical because it will a) correct for side slip that cannot be detected by other sensors b) allow the acquisition of targets and c) allow the experimentation with the direction of visual processing.

The next goal will be to integrate the various maps (allocentric, egocentric) into the behavior of the craft and to use the maps to direct visual processing.

Finally, the craft will be augmented with a grasping device that will allow the primitive manipulation of objects in the environment.

6. Future Work

It is felt that the use of behavioral based control is not orthogonal to traditional control methods. It might still be possible to analyze the stability of behavioral subcomponents through the use of simplified models of the aircraft dynamics. In this way, an approximate behavior may be developed, and its performance confirmed within some limits.

Added to this will be methods of behavioral adaptation. This might be likened to the development of biological organisms. When one observes a human newborn, one can

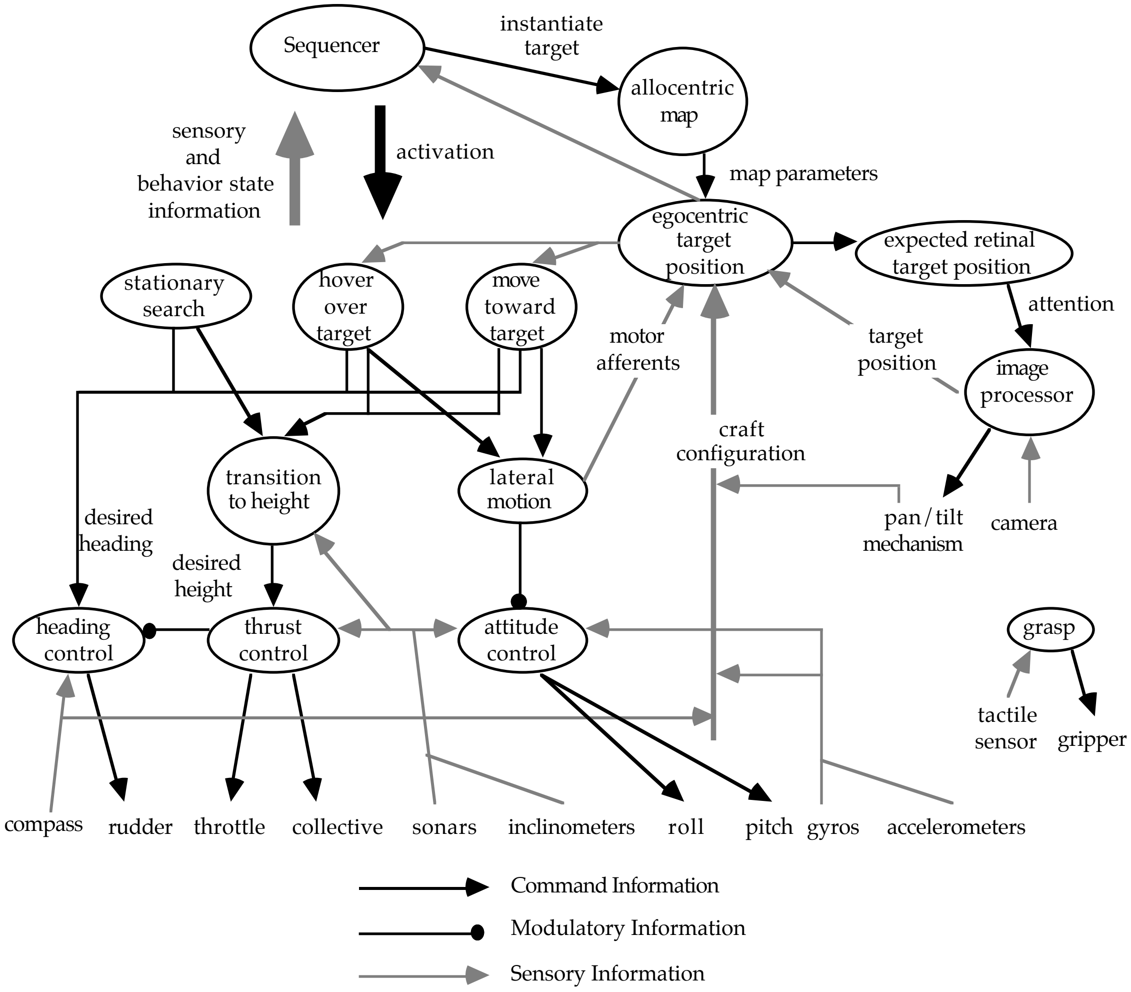

Figure 3: Detailed view of the behavior connectivity.

readily observe the rough outlines of complex behaviors that will be refined through the life of the person. The walking reflex and eye movement elicited by acoustic stimuli are examples. Nature has hardwired a crude architecture that is refined by experience. In a similar way, a crude systems analysis can bring us into the ball-park and adaptive behavioral methods can improve the performance of individual behaviors and the coordination among a set of behaviors.

One example of this can be seen in current work towards adaptive attitude control. In the attitude control behavior described earlier, corrections to the roll and pitch commands are made based upon a linear error control law. Not unexpectedly, when the gains are reasonably stiff, the craft does tend to oscillate slightly while hovering (+/- 5^o). The planned approach to this problem is to *augment* the current system with an additional behavior that learns to compensate for these factors, as shown in Figure 4. Using a reinforcement-learning approach, the goal will be to learn this feed-forward control term *on-board* within a few minutes of flight time.

Through learning, the *Forward Controller Network* attempts to adjust its parameters so as to minimize the future expected error in roll (starting from time t), or more explicitly maximize:

$$E\left\{-\sum_{\tau=t}^{\infty}\lambda^{\tau-t}r(\tau)^2\right\}$$

where λ is a future discount factor on the range [0,1], and $r(t)$ is the roll offset from horizontal at time t.

The network operates by mapping the current state (in this case r and $\dot{r}$) into the parameters of a probability distribution function, from which a control output is randomly selected. At the same time, the *Predictor Network* attempts to learn the above expected value, based on the plant's behavior in response to the control output. When the Predictor P(t) is a perfect one, observe that:

$$\begin{aligned}P(t) &= E\left\{-\sum_{\tau=t}^{\infty}\lambda^{\tau-t}r(\tau)^2\right\}\\ &= E\left\{-r(t)^2-\lambda\sum_{\tau=t+1}^{\infty}\lambda^{\tau-t-1}r(\tau)^2\right\}\\ &= E\left\{-r(t)^2+\lambda P(t+1)\right\}\end{aligned}$$

Once an action is taken at time t, we can observe the actual cost, $r(t)^2$, and the prediction at the resulting state, $P(t+1)$. The expression $R=\lambda P(t+1)-r(t)^2-P(t)$ represents the deviation of the predicted future outcome from the actual future outcome (for one time step). When $R > 0$, the action taken at time t resulted in better than expected performance. Based on this information, the parameters of the probability distribution function can be adjusted so as to improve the future performance of the control network. In addition, R can also be used to adjust the parameters of the prediction function P so as to take into account the changing behavior of the controller.

The key to this approach is that adjustments made for one time step not only take into account the immediate resulting cost, but the whole sequence of future costs. More detail on this type of learning algorithm may be found in [16, 20].

7. Conclusions

The system described here is designed to function within a rather narrow performance envelope. The craft is not meant to move quickly nor recover from large disturbances. A different set of issues were focused upon:

The first issue was the extent to which a behavioral based approach to autonomous helicopter control is feasible. Within the narrow envelope of performance for the task described here, it is felt that this approach is adequate.

The second issue is one of creating a completely autonomous helicopter, with sensing and processing on board. The constraint on payload of the craft translates into a constraint in on-board processing. This necessitates the use of an active vision strategy to selectively probe the environment for information, and thereby reduce the computational requirements. Selective probing then generates the need for a coordination between expectancies from higher centers and actual visual information.

All of these issues combine to create a rather complex control environment which, if successful, should prove as a compelling demonstration of the behavior based approach.

The results presented here have been encouraging. Further experimentation is needed to determine if the system will function well as a whole.

8. Acknowledgments

This work has been sponsored in part by the Gordon Marshall Fund at USC, a grant from the Institute for Robotics and Intelligent Systems at USC (Ram Nevatia and Ken Goldberg), a fellowship from the University of Southern California Graduate School, School of Engineering, Computer Science Department, and from the Hughes Aircraft Company. In addition, we would like to thank those involved with the USC AFV team, especially Alberto Behar, Parag Batavia, Gaurav Sukhatme, Gerry Kim, Lucia Simo and Mathew Lamb.

9. Appendix: The Craft Model

In the discussion of the *egocentric target position* behavior (Figure 3), the actual frame of reference was left ill-defined. In this appendix, we define this frame of reference explicitly and then show how positions in this frame of reference may be mapped to and from the retinocentric coordinate system.

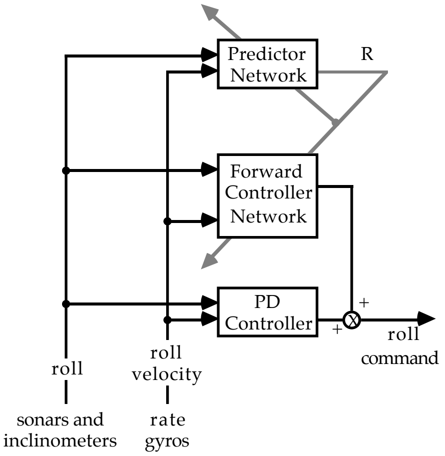

Figure 4: A reinforcement-based learning scheme for acquiring feed-forward control terms.

The frame of reference used by the *egocentric target position* behavior is best described as a geo-ego coordinate system. The orientation of this reference frame is defined by the gravity and magnetic north vectors. In addition, the height of the craft (along Z) is defined relative to the ground. However, the origin of the space is defined relative to the position of the craft (that is for the X and Y dimensions). Thus, this frame of reference has an egocentric component in that the position of the target is measured relative to the current location of the craft within the ground plane.

Figure A1 shows a representation of the craft's configuration as a kinematic chain. Frame 0 in this chain is the geo-ego coordinate frame. Frame 3 is the true egocentric coordinate system, where the transformation from Frame 0 to this frame is defined by the yaw, height, pitch, and roll of the craft (parameters which are derived from the sonars and compass readings).

Frame 5 is the coordinate system of the camera (i.e. retinocentric coordinates). The transformation from Frame 3 to this frame includes the pitch and yaw induced by the saccade system. These and the above parameters are what is referred to as the *craft configuration*.

The final two rotational transformations are one way of representing the position of the object within retinal coordinates. This representation facilitates the transformation between the object location and the position of object in retinocentric coordinates, as defined by the pinhole camera model. In other words, Frame 7 is the coordinate frame rooted at the focal point of the camera, with its X-axis pointed directly at the object. The rotation of the frame about the X-axis is defined by the roll of the craft.

Computation of the Expected Retinal Target Position

Given the estimated position of the object in the geo-ego frame of reference, the system must produce an expectation of the object's location in the visual frame of reference. This enables the vision system to concentrate its computational power towards this region of the image.

In this case, the transformation ${}^{5}_{0}T$ (the homogeneous transform representing the craft configuration) and ${}^{0}P$ (the

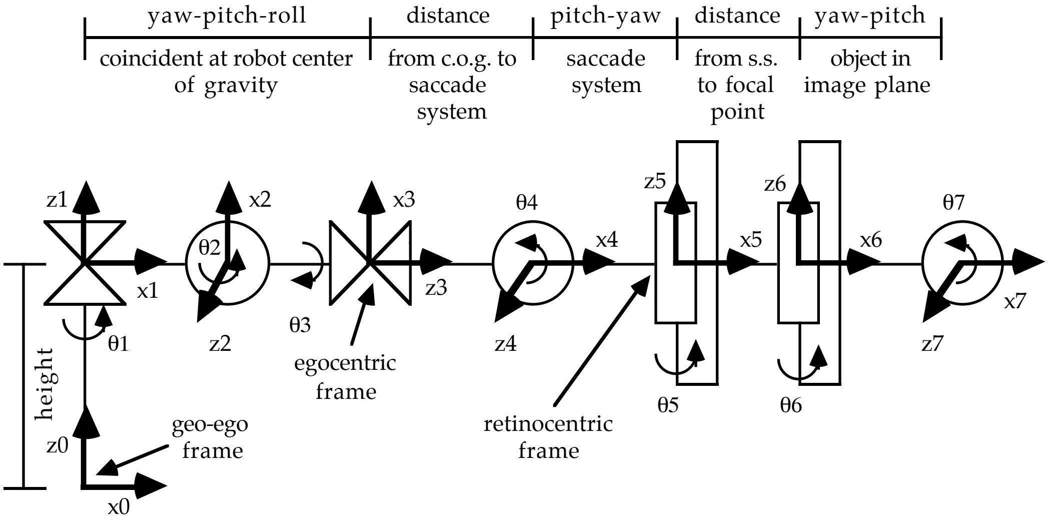

Figure A1: Representation of the craft configuration as a kinematic chain.

position of the target in geo-ego coordinates) are known. The transformation ${}^{7}_{5}T$ is then computed such that Frame 7 is oriented directly at the object. Using the pinhole camera model and a correction for lens distortion, the expected position in the image may then be computed.

Computation of the Geo-ego Position from Visual Information

Once the target object has been visually located, this information is used to compute a more accurate estimate of its geo-ego position. From the position of the object's centroid in retinal coordinates, the transformation ${}^{5}_{7}T$ is computed. Thus, ${}^{0}_{7}T$ is completely defined (using the craft's configuration).

Because Frame 7 is oriented towards the object, its position, ${}^{7}P$ is simply $[\text{depth}, 0, 0, 1]^T$. However, the *depth* is left undefined. This problem may be solved by first transforming this position into the geo-ego reference frame:

$$[x, y, z, 1]^T = {}^{0}_{7}T\,{}^{7}P$$

Solving this equation yields expressions of x, y, and z as a linear function of *depth*. However, since the target object's location is known to be at z=0, *depth* may be computed analytically. From this point, it is a simple substitution to compute the x, y position of the target relative to the craft.

10. References

[1] *Brooks, R.A., Intelligence Without Reason. 1991, MIT.*

[2] *Heiger, M.W., P.K.A. Menon, and O.P. Schrage, Synthesis of a helicopter full-authority controller. Journal of Guidance, Control, and Dynamics, 1992.* ***15****: p. 222-227.*

[3] *Cheng, V.H.L. and T. Lam. Automatic Guidance and Control Laws for Helicopter Obstacle Avoidance. in IEEE International Conference on Robotics and Automation. 1992. Nice, France:*

[4] *Ahmad, S. (1992). personal communication*

[5] *Atkeson, C. (1992). personal communication*

[6] *Borenstein, J. (1992). personal communication*

[7] *Schwartz, D.G. and G.J. Klir, Fuzzy logic flowers in Japan, in IEEE Spectrum. 1992, p. 32-35.*

[8] *Neisser, U., Cognition and Reality: Principles and Implications of Cognitive Psychology. 1976, W. H. Freeman.*

[9] *Brooks, R.A., A robust Layered Control system for a mobile robot. IEEE J. Robotics and Automation, 1986.* ***RA-2****(RA-2): p. 14-23.*

[10] *Brooks, R.A., A robot that walks: Emergent behavior from a carefully evolved network. Neural Computation, 1989.* ***1****(2): p. 253-262.*

[11] *Maes, P., Situated agents can have goals. Robotics and Autonomous Systems, 1990.* ***6****: p. 49-70.*

[12] *Payton, D.W., Internalized plans: a representation for action resources. Robotics and Autonomous Systems, 1990.* ***6****: p. 89-103.*

[13] *Payton, D.W. An architecture for reflexive autonomous vehicle control. in IEEE International Conference on Robotics and Automation. 1986. San Francisco:*

[14] *Arkin, R.C., Integrating behavioral, perceptual, and world knowledge in reactive navigation. Robotics and Autonomous Systems, 1990.* ***6****: p. 105-122.*

[15] *Agre, P.E. and D. Chapman, What are plans for? Robotics and Autonomous Systems, 1990.* ***6****: p. 17-34.*

[16] *Barto, A.G. and S.H. Bradtke, Real-Time Learning and Control using Asynchronous Dynamic Programming. 1991, Department of Computer Science, University of Massachusetts, Amherst.*

[17] *Sutton, R.S. Integrated Architectures for Learning, Planning, and Reacting Based on Approximating Dynamic Programming. in Seventh International Conference on Machine Learning. 1990.*

[18] *Bekey, G.A. and R. Tomovic. Reflex control of robot actions. in IEEE International Conference on Robotics and Automation. 1986. San Francisco:*

[19] *Arbib, M.A., The Metaphorical Brain 2 Neural Networks and Beyond. 1989, New York: Wiley Interscience.*

[20] *Barto, A.G., R.S. Sutton, and C.W. Anderson, Neuronlike Adaptive Elements That Can Solve Difficult Learning Control Problems. IEEE Transactions on Systems, Man, and Cybernetics, 1983.* ***SMC-5****: p. 834-846.*