Final Report

NASA Grant No. NAG 2-221

MODELING PILOT INTERACTION WITH AUTOMATED DIGITAL AVIONICS SYSTEMS

Guidance and Control Algorithms for Contour and Nap-of-the-Earth Flight

Ronald A. Hess
Dept. of Mechanical, Aeronautical, and Materials Engineering
University of California
Davis, CA 95616

(NASA-CR-186105) MODELING PILOT INTERACTION WITH AUTOMATED DIGITAL AVIONICS SYSTEMS: GUIDANCE AND CONTROL ALGORITHMS FOR CONTOUR AND NAP-OF-THE-EARTH FLIGHT Final Report, period ending 30 Sep. 1990 (California

N91-13875

Unclas
G3/54 0305302

Summary

The final report for NASA Grant No. NAG 2-221 is a collection of the technical papers written under grant support. These papers cover two different topics (1) Modeling Pilot Interaction with Automated Digital Avionics Systems, and (2) Guidance and Control Algorithms for Contour and Nap-of-the-Earth Flight. The grant title reflects only the first of these topics.

The papers are as follows:

Modeling Pilot Interaction with Automated Digital Avionics Systems:

(1) "Automation Effects in a Multiloop Manual Control System," IEEE Transactions on Systems, Man, and Cybernetics, SMC-16, No. 1, Jan/Feb 1986, pp. 111-121.

(2) "A Qualitative Model of Human Interaction with Complex Dynamic Systems," R. A. Hess, IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-17, No. 1, Jan/Feb., 1987, pp. 31-51.

Guidance and Control Algorithms for Contour and Nap-of-the-Earth Flight:

(3) "Generalized Predictive Control of Dynamic Systems," R. A. Hess, and Y. C. Jung, Proceedings of the 1988 IEEE International Conference on Systems, Man, and Cybernetics, Aug. 8-12, Beijing and Shenyang, China, pp. 844-849.

(4) "An Application of Generalized Predictive Control to Rotorcraft Terrain-Following Flight," R. A. Hess and Y. C. Jung, IEEE Transactions on Systems, Man, and Cybernetics, Vol, SMC-19, No. 5, Sept/Oct 1989, pp. 955-962.

(5) "Self-Tuning Generalized Predictive Control Applied to Terrain-Following Flight," R. A. Hess, and Y. C. Jung, AIAA Paper No. 89-3450, 1989 AIAA Guidance, Navigation and Control Conference, Boston, MA, Aug. 14-16.

(6) "Precise Flight Path Control Using a Predictive Algorithm," Y. C. Jung and R. A. Hess, Journal of Guidance, Control, and Dynamics, to appear.

0018-9472/86/0100-011[illegible] EEE

Automation Effects in a Multiloop Manual Control System

RONALD A. HESS AND B. DAVID McNALLY

Abstract—An experimental and analytical study was undertaken to investigate human interaction with a simple multiloop manual control system in which the human's activity was systematically varied by changing the level of automation. The system simulated was the longitudinal dynamics of a hovering helicopter. The automation-systems-stabilized vehicle responses from attitude to velocity to position and also provided for display automation in the form of a flight director. The control-loop structure resulting from the task definition can be considered a simple stereotype of a hierarchical control system. The experimental study was complemented by an analytical modeling effort which utilized simple crossover models of the human operator. It was shown that such models can be extended to the description of multiloop tasks involving preview and precognitive human operator behavior. The existence of time optimal manual control behavior was established for these tasks and the role which internal models may play in establishing human–machine performance was discussed.

I. Introduction

MANY descriptions of human–machine interaction in the control of dynamic systems exist. Rasmussen [1], for example, partitions human behavior into skill-based, rule-based, and knowledge-based activity as shown in Fig. 1. The structure of this figure is hierarahical and goal-oriented. Existing descriptions of human data processing can be associated with each of the levels shown. For example, heuristic problem-solving models [2] can be associated with the knowledge-based level. So-called production rule models [3] can be associated with the rule-based level. Finally, control theoretic models like the optimal control model [4] or the crossover model [5] can be associated with the skill-based level.

Albus [6] offers a more structured description of a sensory-processing/behavior-generating hierarchy in Fig. 2. This parallel structure offers any number of hierarchical levels, although only four are shown. The H modules decompose input goals C into output subgoals P using feedback F. The M models recall expected sensory data R, which is compared with observed sensory experiences E. The G modules recognize sensory patterns Q and compute feedback errors F.

Rouse [7] offers a very simple yet descriptive model of tasks involving the manual (or automatic) control of dynamic systems as shown in Fig. 3. Here, Ω_i represents a generalized "bandwidth" indicating the relative time scales involved in each of the loops shown. The nesting of feedback loops with $\Omega_1 > \Omega_2 > \Omega_3 > \cdots > \Omega_n$ is a characteristic of nearly all dynamic control systems, no matter how complex. As an example of an aircraft flight control problem, the loops of Fig. 3 could be interpreted as follows: the block denoted Ω_1 represents attitude control with a relatively high bandwidth. Block Ω_2 represents altitude control with a lower bandwidth, while block Ω_3 represents navigation activity with a still lower bandwidth.

There are at least two similarities in all of these models. First, they are hierarchical and second, they are goal-directed or oriented. The way in which a man and computer can interact in the system of Figs. 1–3 can be quite varied. In discussing the system of Fig. 3, for example, Rouse [7] has outlined, classified, and discussed several methods of man–computer interaction. Fig. 3 invites a simple and practical allocation of tasks between human and computer (manual and automatic control) in any task. One can start at the innermost loop and begin automating the feedback activity loop by loop. This means that the human is responsible for fewer loop closures as the automation proceeds and these with lower and lower bandwidths. Conversely, one can start at the outermost loop and begin the automation process. Again, as the automation proceeds, the human is responsible for fewer loop closures, but the bandwidth of the manual control task is, in this case, dominated by the innermost loop. Both of these schemes are consistent with current practice in aircraft flight control automation. For example, the first is exemplified by an automatic landing system while the second is exemplified by the same landing task using a cockpit flight director. Both schemes can result in increased human–machine performance and decreased subjective estimates of "workload". This approach to inner-to-outer-loop automation is quite similar to that adopted by Yoerger [8] in his study of automation effects in the multi-axis control of a simulated transport aircraft.

It is of some interest to analyze the two approaches to automation just described in the context of a multiloop manual control task. To this end, a human-ir-the-loop simulation was conducted in a fixed-base simulator. The task considered was that of the longitudinal control of a hovering helicopter as indicated in Fig. 4. The task required the helicopter to follow a discrete periodic position command. In terms of human participation, the task can

Manuscript received November 25, 1984; revised August 12, 1985. This work was supported by the National Aeronautics and Space Administration under grant NAG 2-221.

The authors are with the Department of Mechanical Engineering, University of California, Davis, CA 95616, USA.

IEEE Log Number 8405916.

0018-9472/86/0100-0111$01.00 ©1986 IEEE

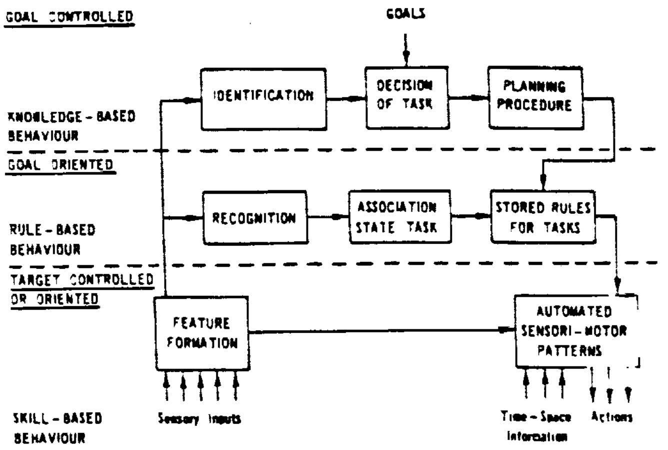


Fig. 1. Three levels of human behavior in the control of dynamic systems, from [1].

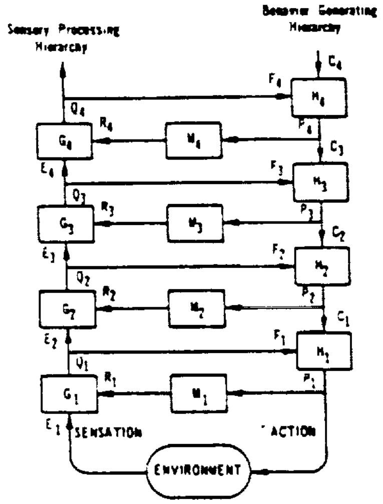


Fig. 2. A sensory-processing behavior-generating hierarchy in the control of dynamic systems, from [2].

be interpreted via Fig. 5, which are annotated forms of Fig. 3.

The selection of appropriate goals ("follow position command") and the ordering of subgoals ("generate appropriate vehicle velocity", etc.) at each level are based upon the task definition and the physical laws governing the vehicle in question. For example, "follow position command" defines the task at hand, and, in a hovering helicopter, pitch attitude is used to control velocity, which in turn is used to control position.

II. Experimental Setup

The vehicle dynamics and display were generated on a Cromemco microprocessor-based simulation system. The display format is shown in Fig. 6. The display itself was a color raster-type measuring 20.3 by 28.0 cm with a nominal eye-to-display distance of 90 cm. Depending upon the type of automation, one of two types of control sticks were used. For the majority of experiments an isometric device was employed. However, for one of the automation levels, an unrestrained finger manipulator was used. The basic vehicle dynamics were very simple and can be given as

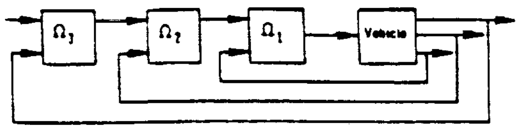


Fig. 3. A simplified model of a dynamic system, from [7].

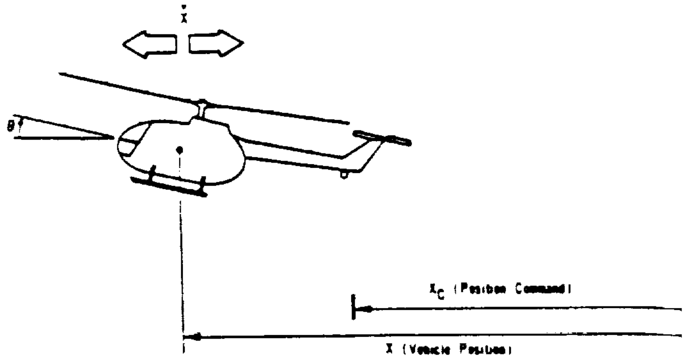


Fig. 4. The simulated longitudinal hover task.

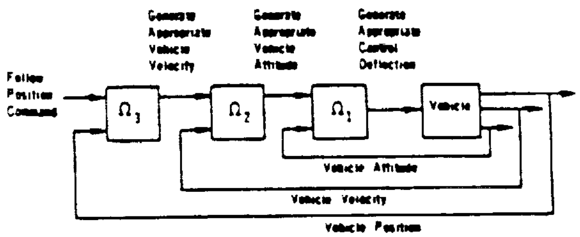

Follow Position Command
Generate Appropriate Vehicle Velocity
Generate Appropriate Vehicle Attitude
Generate Appropriate Control Deflection
Ω_3
Ω_2
Ω_1
Vehicle
Vehicle Attitude
Vehicle Velocity
Vehicle Position

Fig. 5. The structure of Fig. 3 for the task of Fig. 4.

$$\dot{x} = u, \qquad X_u = -0.1/s$$

$$\dot{u} = -g\theta + X_u u, \qquad g = 9.8 \text{ m/s}^2$$

$$\dot{\theta} = K\delta \tag{1}$$

where x represents vehicle position, u vehicle velocity, θ vehicle attitude, and δ control output.

The command signal x_c was chosen as a square wave. Three different fundamental frequencies were chosen for investigation: 0.2, 0.3, and 0.6 rad/s. Only the data associated with the first of these frequencies will be discussed in detail here. This command signal was displayed to the subject in preview fashion as the horizontal translation of the "position command" lines on the display of Fig. 6. The moving command lines in Fig. 6 together with the fixed position reference line represent the discrete position command x_c. The command lines move across the screen from right to left at a constant rate commensurate with the fundamental temporal frequency (0.2, 0.3, or 0.6 rad/s). When a command line is crossing the vertical reference line, it represents the commanded position of 15.24 m (50 ft) from the position reference line. When a command line is not touching the vertical reference line, the commanded position is 0 m and is represented by the position reference line.

The automation levels were chosen as different levels of stability and command augmentation and display augmen-

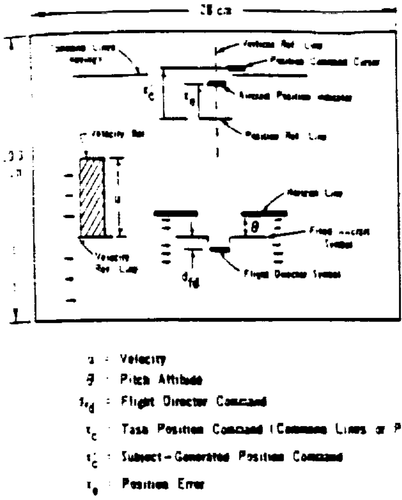


Fig. 6. The display format for the simulated longitudinal hover task.

tation systems as follows. In the case of no automation, the vehicle dynamics were as given in (1), and the human's lowest control level was the control of attitude θ through control input δ. Since the unaugmented dynamics between δ and θ were of the form $\delta = \dot{\theta}/K$, this automation level can, in the parlance of flight control engineers, be called a rate-command attitude-hold system. In the next level of automation, the human's lowest level of control was the control of vehicle velocity through control input θ_c. This means that over some broad but limited frequency range, the vehicle pitch attitude was directly proportional to the human's control input. This is normally referred to as an attitude-command attitude-hold system. In the next level of automation, the human's lowest level of control was the control of vehicle position through control input x_c. Again, over some broad but limited frequency range, the vehicle velocity was directly proportional to the human's control input. This is referred to as a velocity-command position-hold system. In the final automation level associated with the inner-to-outer-loop scheme, the human's lowest level of control was the generation of commanded vehicle position through control input x_c'. This is the highest form of automation possible in this system while still giving the human some control responsibility. This system is referred to as a position-command position-hold system. It was in this system that the unrestrained finger manipulator was used in lieu of the isometric control stick to move the position-command cursor on the display of Fig. 6.

The outer-to-inner-loop automation scheme was mechanized by designing a flight director for this vehicle and task. A flight director is a system in which all the sensed variables used by the human in completing a task are combined into one display element forming a single-loop compensatory tracking task for each control available to the human. Details of the design of the automation systems are given in the Appendix.

Four naive subjects participated in the experiment. Each simulation run lasted approximately 95 s. Each subject saw twelve different configurations presented in the order shown in Table I. This ordering is pseudorandom in that an attempt was made to randomize the order of presentation while not giving the subjects very difficult tasks early in the experiment. Control sensitivities were selected for each automation level by a subject with tracking experience who was not one of the four test subjects. Note that the augmentation and flight director designs were synthesized assuming a 0.2 rad/s command frequency. The 0.3 and 0.6 rad/s command frequencies were included in the experiment to ascertain the performance and subjective opinion decrements associated with higher bandwidth operation. These effects will be summarized briefly in Section IV.

For the reason just given, the flight director configuration was omitted from the experimental matrix at the 0.3 and 0.6 rad/s command frequencies. Task difficulty led to the omission of the rate command system at the 0.6 rad/s command frequency. Root-mean-square (RMS) performance scores were recorded as were pilot opinion ratings of task difficulty quantified on a nonadjectival rating scale [9]. This scale has numerical values from one to ten, with one reflecting very little task difficulty and ten reflecting very great task difficulty. In addition, a "workload" measure consisting of the number of control inputs used in each task by each subject was measured. This metric is similar to that proposed by Weirwille and Connor [10]. As implemented in this study, a single control input (a force for the isometric stick and a displacement for the unrestrained manipulator) was said to occur when a) the control rate changed sign and b) the control amplitude measured from the point where the rate changed sign exceeded a criterion value. The criterion value used here was 75 percent of the RMS value of the control amplitude for the entire run. The subjects were instructed to minimize vehicle position errors while maintaining vehicle pitch attitude rates within "reasonable" levels. To quantify the latter, an audio alarm sounded whenever the pitch rate exceeded 10° s. The percentage of the total run time during which the pitch rate exceeded 10° s was also measured and recorded. Data were taken only after the subjects RMS performance scores stabilized and were repeatable from run to run.

III. Models for Compensatory Human Operator Behavior

To begin an analysis of the task described above, a compensatory control structure was assumed as shown in Fig. 7. It should be emphasized that the compensatory structure is just a starting point. As Figs. 1–3 indicate, human–machine interaction can be a good deal more complex than the servomechanism-like behavior implied by Fig. 7. However, as will be seen, the rather simple multiloop feedback structure of Fig. 7 can shed considerable light on the possible forms of human dynamics at the compensatory level and can be used to generate acceptable automation systems as outlined in the Appendix. In all the cases to be studied, the form of the human compensation was derived by application of the crossover model of the human operator for each loop closure under manual control.

ORIGINAL PAGE IS
OF POOR QUALITY

TABLE I
HOVER TASK AUTOMATION CONFIGURATIONS

Position Command Frequency (rad/s)	Automation System
0.2	velocity command
0.3	attitude command
0.6	position command
0.2	rate command
0.2	flight director
0.3	velocity command
0.2	attitude command
0.3	position command
0.6	velocity command
0.3	rate command
0.2	position command
0.6	attitude command

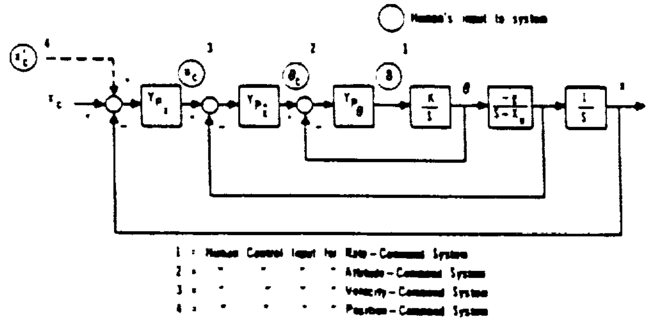

Human's input to system
x_c'
4
3
2
1
u_c
θ_c
δ
x_c
Y_{p_x}
Y_{p_u}
Y_{p_θ}
$\frac{K}{s}$
θ
$\frac{-g}{s-X_u}$
$\frac{1}{s}$
x
1 : Human Control Input for Rate-Command System
2 : " " " " Attitude-Command System
3 : " " " " Velocity-Command System
4 : " " " " Position-Command System

Fig. 7. A compensatory control structure for the hover task.

Rate-Command System: Here no automation is encountered in Fig. 7. The human is responsible for providing all compensation and for sensing appropriate feedback variables for three loop closures: attitude, velocity, and position. A useful rule-of-thumb in multiloop manual control situations is to separate the bandwidths (or individual loop crossover frequencies) of each successive closure by a factor of three [11]. Selecting the crossover frequency of the outer-position loop equal to ω_0, the frequency of the fundamental component of the square-wave position command x_c, yields

$$\begin{aligned} \omega_{c_x} &= \omega_0 \\ \omega_{c_u} &= 3\omega_0 \\ \omega_{c_\theta} &= 9\omega_0. \end{aligned} \tag{2}$$

Now considering the two inner loops to be closed by the human, the outer open-loop transfer function x/u_c can be approximated as

$$\frac{x}{u_c} = \frac{1}{s}\bigg|_{\omega = \omega_{c_x}}. \tag{3}$$

Applying the crossover model to this closure suggests

$$Y_{p_x} = \omega_{c_x}. \tag{4}$$

Moving to the next loop

$$\frac{u}{\theta_c} = \frac{-g}{s - X_u} = \frac{-g}{s}. \tag{5}$$

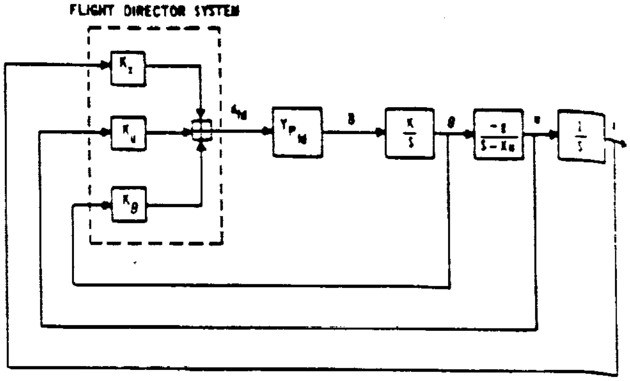


Fig. 8. The flight director design for the hover task.

Again, the crossover model suggests

$$Y_{p_u} = \frac{-\omega_{c_u}}{g}. \tag{6}$$

Finally, looking at the inner loop

$$\frac{\theta}{\delta} = \frac{K}{s} \tag{7}$$

and

$$Y_{p_\theta} = \frac{\omega_{c_\theta}}{K} e^{-\tau_\theta s}, \qquad \tau_\theta = 0.3 \text{ s}. \tag{8}$$

For simplicity, the effective time delay τ_θ of the human operator has been placed in the innermost manual control loop. This delay has been assigned a value of 0.3 s [5].

Attitude Command System: Here, the inner loop closure $\theta \rightarrow \delta$ is handled by the automation and the required human compensation is summarized by

$$\begin{aligned} Y_{p_x} &= \omega_{c_x} \\ Y_{p_u} &= \frac{-\omega_{c_u}}{g} e^{-\tau_u s}; \qquad \tau_u = 0.3 \text{ s} \end{aligned} \tag{9}$$

Velocity Command System: Here, the inner loop closure $\theta \rightarrow \delta$ and $u \rightarrow \theta_c$ are handled by the automation and the required human compensation is

$$Y_{p_x} = \omega_{c_x} e^{-\tau_x s}; \qquad \tau_x = 0.3 \text{ s} \tag{10}$$

Position Command System: The position command system was not considered amenable to description as a compensatory tracking task since the human is providing the command to the system.

Flight Director System: Fig. 8 is a block diagram representation of the flight director system. This task is compensatory in nature and instructions to the subjects emphasized this. As the Appendix indicates, the dynamics of the flight director system can be given as

$$\frac{d_{fd}}{\delta} = \frac{K_{fd}}{s}\bigg|_{\omega = \omega_{c_\theta}} \tag{11}$$

Thus the human's compensation takes the form

$$Y_{p_{fd}} = \frac{\omega_{c_\theta}}{K_{fd}} e^{-\tau_{fd} s}, \qquad \tau_{fd} = 0.3 \text{ s}. \tag{12}$$

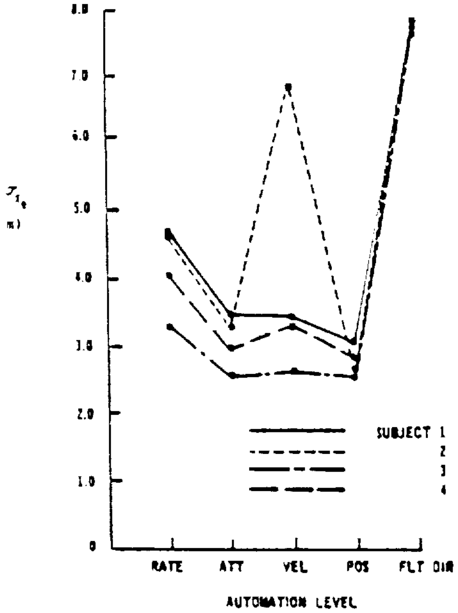


Fig. 9. Outer loop RMS position error performance for different automation levels. Values shown are averages for five runs.

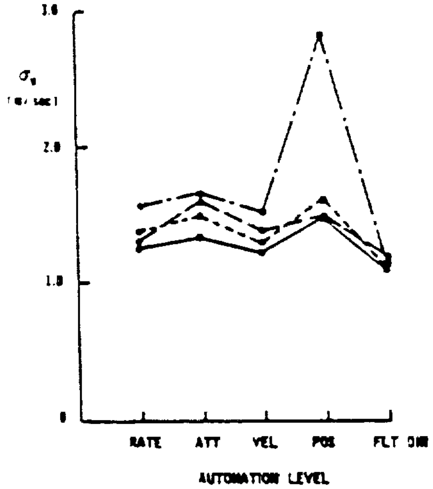


Fig. 10. RMS velocity excursions for different automation levels. Values shown are averages for five runs.

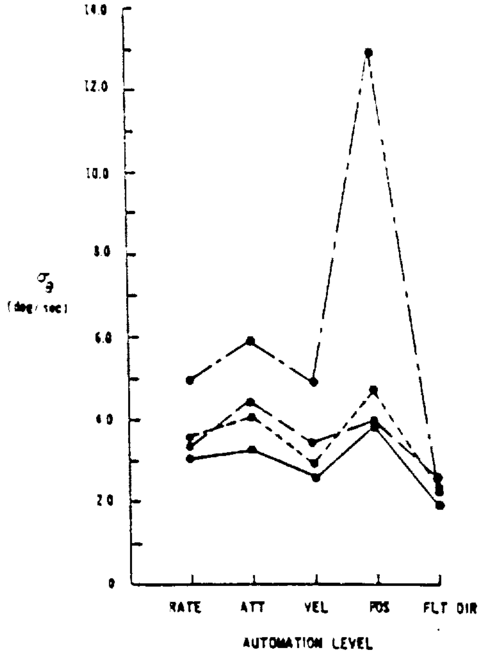


Fig. 11. RMS pitch attitude excursions for different automation levels. Values shown are averages for five runs.

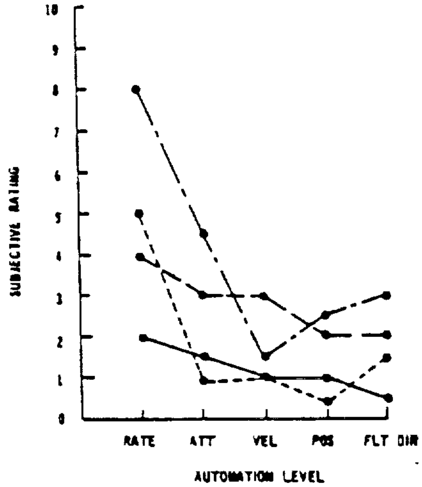


Fig. 12. Subjective task difficulty ratings for different automation levels.

It is important to point out at this juncture that all the automation schemes just discussed were designed using crossover models of the human operator (omitting time delays, of course). Thus, all automated loops should exhibit dynamic characteristics very similar to those in evidence when the same loops are closed manually.

IV. Experiment

Results

Fig. 9–11 shows the RMS position error, velocity, and pitch attitude excursions for the four subjects across the five automation levels for $\omega_0 = 0.2$ rad/s. Table II tabulates these values along with the standard deviations and the subjective difficulty ratings generated by the subjects. Fig. 12 shows the ratings averaged across the subjects. Fig. 13 shows the results of the control input analysis. Table II gives these values and standard deviations for each subject and automation level. Fig. 14 shows representative time histories for the subject with the best position performance (subject 3). These time histories are for two commanded position changes. Finally, Fig. 15 shows a pair of x_c' time histories for the position command system. Fig. 15(a) exhibits the "aggressive" style adopted by subject 3, while Fig. 15(b) demonstrates the "less aggressive" style adopted by the remaining subjects.

Discussion

A review of the results of Figs. 9–15 reveals the following.

1) As Fig. 9 indicates, position performance generally improves with increasing automation from the inner to outer loop, although the performance differences are

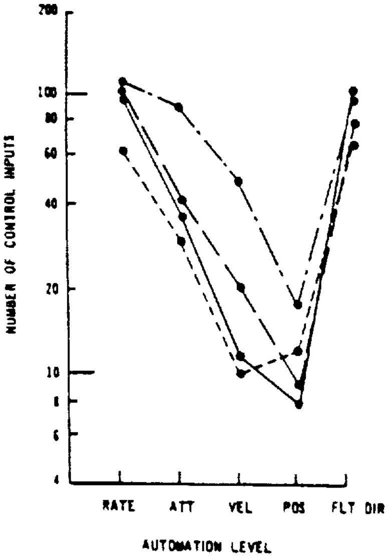


Fig. 13. Number of control inputs for 90 sec run for different automation levels. Values shown are averages for five runs.

surprisingly small. Subject 2's performance with the velocity system is somewhat anomalous. Since the velocity system with $\omega_0 = 0.2$ rad/s was the first system presented to the subjects, and since subject 2 performed better with the rate system than with the velocity system (as opposed to the remaining subjects), insufficient training is indicated. However, subject 2's position performance scores with the velocity system had stabilized. This is corroborated by the small standard deviation of these scores as given in Table II.

The poor position performance of all subjects with the flight director (automation from outer to inner loop) is attributable to the fact that the subjects could not utilize preview information and were forced to track in compensatory fashion. The subjects were instructed to follow the director command and ignore any preview information that might be provided by the translating square wave in the display format of Fig. 6.

2) Subjective opinion data generally indicate decreasing task difficulty with increasing automation whether inner to outer loop (attitude, velocity, and position systems) or outer to inner loop (flight director).

3) Although Fig. 14 exemplifies time histories from subject 3's data, it is generally representative of all the subjects in the experiment. As the figure indicates, preview information was utilized for all the inner-to-outer-loop automation schemes, i.e., no apparent lags are evident between the fundamental components of the command input and the response. The same cannot be said for the flight director results. Here, the position response shows considerable lag (approximately 4–5 s) as compared to the command. This is attributable to the compensatory tracking behavior for this system.

4) Interpreting the control input data of Fig. 13 as a measure of workload [10], suggests that inner to outer-loop automation results in progressive reductions in task difficulty. This is generally corroborated by the subjective ratings of Fig. 12. However, the control input data for the flight director (automating from outer to inner loop) wou indicate a difficulty level comparable to that for the ra system (no automation). Clearly, this is not corroborat by the rating data. The anomaly can be explained by fact that all the subjects tracked quite aggressively with director and, in an attempt to immediately null direc errors, would use rather large control inputs. The subje were aware of the rather sluggish performance of director system (see Fig. 14(e)) and attempted to com pensate for this by abrupt control inputs. This behavior d not seem to have a significant effect upon subjective ra ings, however.

5) The x_c' time histories generated by the subjects we very repetitive and indicate precognitive behavior [5]. The well-rehearsed precognitive control movements are a reflected in the relatively small standard deviations in control input data of Table II for the position command compared to the remaining systems. As mentioned in preceeding, subject 3 was a good deal more aggressive th the remaining subjects in using the position comma system. The impulsive control movements evident in F 15(a) are responsible for the large σ_{u}, and σ_{θ} values hibited by this subject in Figs. 10 and 11. Subject 3 a appeared to ignore the audio alarm on pitch rate. T percentage of the run time during which pitch rate ceeded 10° s was typically over an order of magnitu higher for subject 3 as compared to the remaining subjec

6) For $\omega_0 = 0.3$ rad/s RMS performance scores a time histories showed the same trends as for the 0.2 rad command, e.g., outer-loop position is roughly equival across all levels of automation studied. At $\omega_0 = 0.6$ rad all subjects adopted a control strategy which was qu different than that for the lower frequency commands. all levels of automation studied, this strategy led ou loop position response which was nearly sinusoidal nature but still exhibited approximately the same fun mental frequency as the command with little appar phase lag.

V. Analysis

Analytical models of the human operator were de oped in three stages and implemented in an off-line co puter simulation of the human-in-the-loop tasks just scribed. The models were all based upon very sim crossover representation of the human with some refi ments to handle the effects of preview and precogniti Fig. 7 and (4), (6), (8)–(10), and (12) describe the mod

Level 1 Model: With one exception, a model for pur compensatory behavior (assuming no preview or precog tion) provided poor RMS performance and qualitative ti history matches with experimental results. The one exc tion was the flight director system in which the compen tory model did quite well. This is not surprising since flight director demanded compensatory behavior of subjects.

Level 2 Model: Here the level 1 model was modified allowing the model to be driven by a command identica

TABLE II
EXPERIMENTAL RESULTS—RMS SCORES BASED UPON FIVE RUNS PER SUBJECT[1]

	Subject 1					Subject 2				
Automation Level	σ_{x_e} (m)	σ_u (m/s)	σ_θ (deg)	Control Inputs	Rating	σ_{x_e} (m)	σ_u (m/s)	σ_θ (deg)	Control Inputs	Rating
Rate	4.68	1.27	3.03	96.4	2.0	4.61	1.39	3.57	62.2	5.0
	(0.31)	(0.13)	(0.24)	(19.6)		(0.40)	(0.12)	(0.27)	(8.90)	
Attitude	3.48	1.37	3.25	36.0	1.5	3.34	1.51	4.11	29.4	0.9
	(0.18)	(0.061)	(0.22)	(11.2)		(0.32)	(1.64)	(0.70)	(8.50)	
Velocity	3.43	1.24	2.52	11.4	1.0	6.93	1.30	2.79	10.0	1.0
	(0.18)	(0.070)	(0.30)	(2.50)		(0.28)	(0.12)	(0.39)	(0.80)	
Position	3.06	1.51	3.83	8.0	1.0	2.72	1.62	4.71	12.2	0.3
	(0.25)	(0.024)	(0.16)	(0.70)		(0.19)	(0.015)	(0.11)	(1.90)	
Flt Dir	7.77	1.11	1.93	104.2	0.5	7.85	1.11	2.27	67.6	1.5
	(0.31)	(0.043)	(0.10)	(14.1)		(0.38)	(0.034)	(0.12)	(11.0)	
	Subject 3					Subject 4				
Rate	3.30	1.59	5.00	110.0	8.0	4.04	1.32	3.29	105.2	4.0
	(0.45)	(0.021)	(0.46)	(10.6)		(0.27)	(0.070)	(0.13)	(8.30)	
Attitude	2.56	1.67	5.93	90.2	4.5	3.00	1.62	4.44	41.8	3.0
	(0.10)	(0.024)	(0.29)	(5.70)		(0.11)	(0.11)	(0.63)	(6.40)	
Velocity	2.64	1.55	4.94	48.8	1.5	3.34	1.39	3.28	20.2	3.0
	(0.023)	(0.052)	(0.75)	(9.40)		(0.53)	(0.10)	(0.28)	(7.70)	
Position	2.64	2.85	12.9	17.8	2.5	2.80	1.52	3.90	9.0	2.0
	(0.097)	(0.058)	(0.61)	(1.80)		(0.19)	(0.027)	(0.14)	(1.80)	
Flt Dir	7.79	1.15	2.39	108.4	3.0	7.87	1.21	2.52	80.0	2.0
	(0.54)	(0.021)	(0.052)	(31.0)		(0.34)	(0.043)	(0.17)	(15.7)	

[1] Result in parentheses denotes standard deviation.

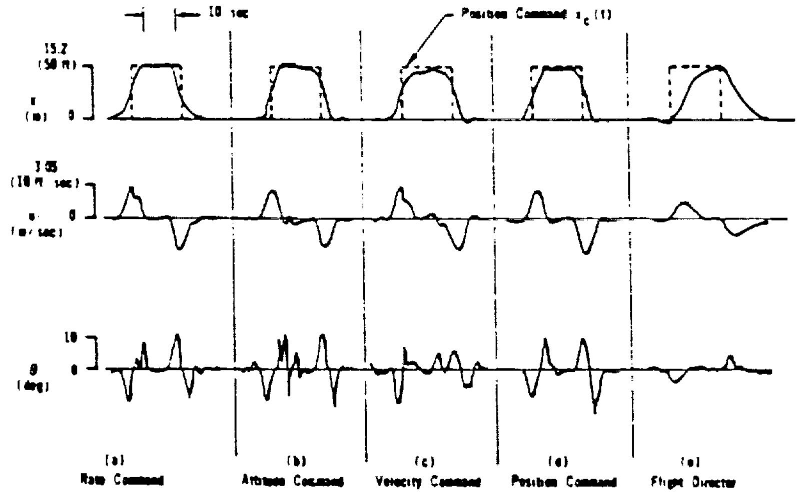


Fig. 14. Typical time responses for subject 3.

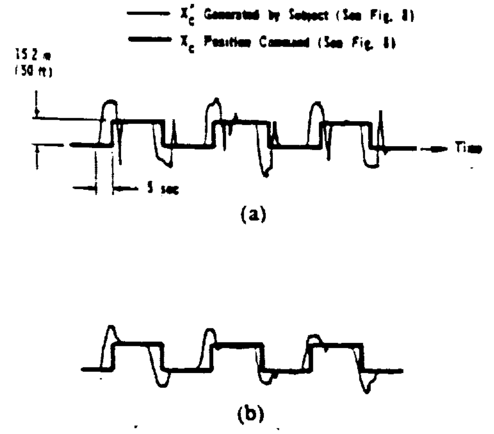


Fig. 15. Subject generated inputs for position command system. (a) the "aggressive" input of subject 3. (b) Less aggressive inputs typical of subjects 1, 2, and 4.

form to the square wave x_c, but advanced by a "preview time constant" of 4 s. The magnitude of this time constant was determined by considering that the final position loop closure in Fig. 7 using the simple Crossover model suggested by (2)–(4) would yield a phase lag of approximately 45° at the command frequency $\omega_0 = 0.2$ rad/s. This, in turn translates into a 3.75 s apparent lag in vehicle position x as compared to commanded position x_c. With this preview time constant, rounded off to 4 s, the RMS performance comparisons improved somewhat. However, as might be expected, the qualitative time history matches were still unsatisfactory.

Level 3 Model: Here the level 2 model was modified by allowing the model to be driven by the command input x'_c which the subjects utilized in the position command system, i.e., the waveforms in Figs. 15(a) or 15(b). Now both

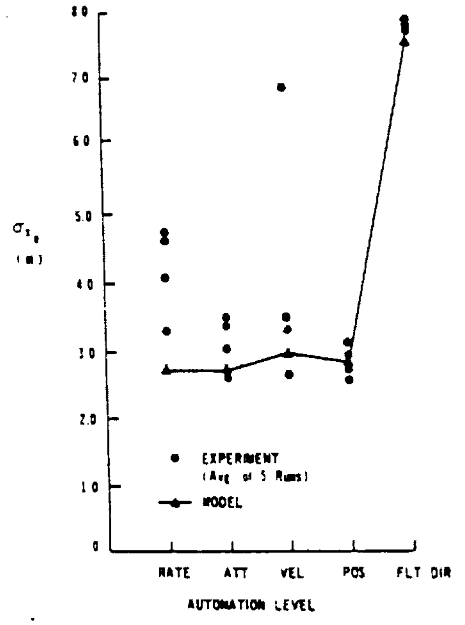


Fig. 16. Model-generated RMS position error performance compared with the data of Fig. 9.

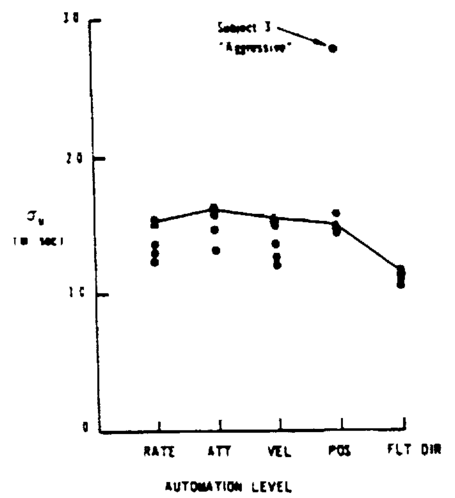


Fig. 17. Model-generated RMS velocity excursions compared with the data of Fig. 10.

RMS and qualitative time history comparisons were quite good. As an example, Figs. 16–18 compare the model RMS performance predictions with the experimental data. The model is being forced with the "less aggressive" input command of Fig. 15(b). As such, it will not match the points in Figs. 17 and 18 attributable to subject 3's "aggressive" tracking behavior. Fig. 19 compares experimental and model-generated time histories for the rate command system for subject 3. In generating the model responses, the Level 3 model was used with the "aggressive" x_c' command of Fig. 15(a) implemented in the model.

Table III summarizes the model parameters used to obtain the acceptable matches with experiment. Table IV shows the model performance values. The type of input command has been included here as a model parameter. While there are eight parameters shown, only the type of input command was derived from the data of this experi-

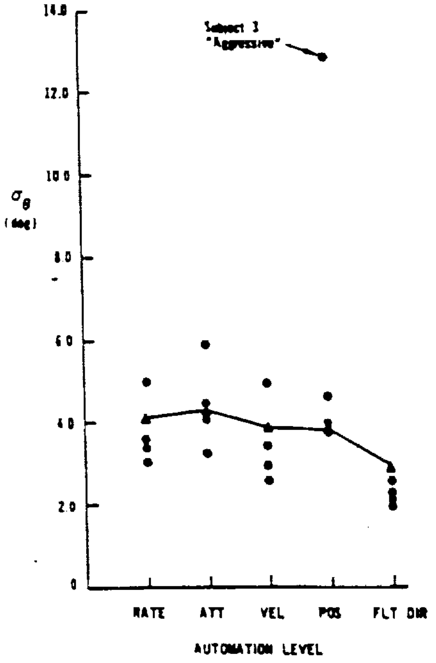


Fig. 18. Model-generated RMS pitch attitude excursions compared with the data of Fig. 11.

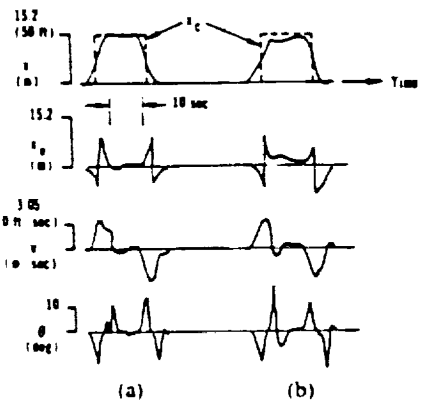


Fig. 19. A comparison of time histories for the rate command system. (a) Experimental. (b) Model generated. Experimental response for subject 3.

TABLE III
Model Parameters

	Automation Level				
	Rate	Attitude	Velocity	Position	Flt Dir
Input Command	x_c'	x_c'	x_c'	x_c'	
τ_θ (s)	0.3	0	0	0	0.3
τ_u (s)	0	0.3	0	0	0
τ_x (s)	0	0	0.3	0	0
ω_{c_θ} (rad/s)	1.8	1.8	1.8	1.8	1.8
ω_{c_u} (rad/s)	0.6	0.6	0.6	0.6	0.6
ω_{c_x} (rad/s)	0.2	0.2	0.2	0.2	0.2
Preview time constant (s)	4.0	4.0	4.0	4.0	4.0

ment. The rest were predicated upon the description of t task and acceptable rules of thumb for application simple crossover models of the human operator.

Fig. 20 compares model-generated control input da with the experimental results. The data range exclud subject 3's data (aggressive inputs) have been noted. T

TABLE IV
MODELING RESULTS

	Automation Level				
	Rate	Attitude	Velocity	Position	Flt Dir
[illegible]	2.71	2.70	2.92	2.77	7.53
[illegible]	1.58	1.65	1.60	1.55	1.21
[illegible] (deg)	4.13	4.33	3.86	3.75	3.06
Control Inputs	33	18	12	6	18

numbers of model-generated inputs are consistently smaller than the experimental values. This is to be expected since no attempt at modeling human operator remnant was included in the model. In addition, since the model is "unaware" of the relatively sluggish performance of the director, it does not control it in aggressive fashion. Hence, the model control input for the director are considerably smaller than experimental values. However, it is worthy of note that the model-generated control inputs reflect the subjective rating results quite well.

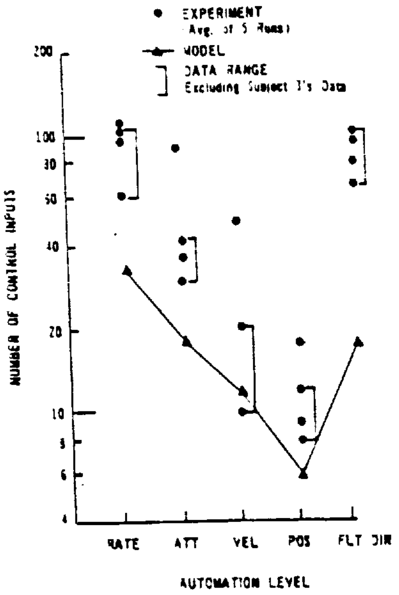


Fig. 20. Model-generated number of control inputs compared with the data of Fig. 13.

Hierarchical-Control Behavior

The analysis just described provides some useful insight into human operator hierarchical-control behavior for the relatively simple manual control task studied. All the experimental data involving inner to outer-loop automation can be adequately explained by the multiloop structure of Fig. 7 using simple crossover models of the human operator—provided that the actual command x_c is replaced by the command x'_c generated by the operator at the highest level of automation (position command system). This suggests that x'_c is being generated by the subjects at the highest hierarchy in Fig. 5 regardless of the level of inner-to-outer-loop automation. Of course, with the flight director essentially no hierarchy is involved, and manual control activity is relegated to the innermost loop and is strictly compensatory in nature.

The fact that all the subjects generated similar, repeatable x'_c time histories which differed from the actual command x_c indicates the existence of some underlying performance criterion. Since the subjects were instructed to fly the vehicle to the commanded position as quickly as possible (with a loose constraint on maximum pitch rate), a time-optimal performance criterion may be in effect. Consider again Fig. 7 and the velocity time histories in Fig. 14(a)–(d). With the θ and u loops closed either automatically, manually or by a combination of the two, the effective vehicle dynamics appear as an integrator (3). McRuer *et al.*, [12] discuss a series of single-loop step command tracking experiments, one of which involved K/s controlled element dynamics. To explain observed operator behavior, [12] analytically solved the time-optimal control problem with the constraint that the control input was limited in magnitude to M, where M may represent either a physical limit on the input magnitude (maximum control input in a single-loop task) or an implicit restraint imposed by the operator for the given situation. Solution of the resulting two point boundary problem in [12] yielded the following "invariance condition" for the time optimality of pulsive control inputs for K/s controlled elements

$$T_c M = A/K_c \tag{13}$$

where

T_c duration of pulsive control input
M average absolute amplitude of control input
A amplitude of step command
K_c controlled element sensitivity.

Now the velocity outputs u shown in Figs. 14(a)–(d) appear to be responses to pulsive velocity commands u_c. In the case of the velocity command system, the u_c inputs, themselves, can be measured. They are pulsive in form and they indicate

$$T_c = 5 \text{ s}$$
$$M = 3 \text{ m/s } (10 \text{ ft/s}).$$

Now (3) indicates $K_c = 1.0$ and solving (13) for A yields $A = 15.24$ m (50 ft). This is, of course, the magnitude of the step command x_c. Thus, the pulsive control inputs u_c, which appear to exist for all the inner-to-outer-loop automation levels, represent time-optimal inputs to the system defined by the dynamics of the lower levels of the control hierarchy. It is interesting to note that the control amplitude M of approximately 3 m/s (10 ft/s) corresponds in magnitude to the second pair of tick marks above and below the velocity reference line for the velocity bar in Fig. 6. The subjects may have been using these marks in generating u_c.

Internal Models

The existence of internal world models and their role in allowing the human to effectively interact with complex dynamic systems has been discussed at some length in the literature [1], [6], and [13]. Such models are explicitly shown in the human operator model of Fig. 2 and are implicit in the models of Figs. 1 and 3. The simple crossover model used here to describe the human elements in Fig. 7, of course, does not contain such an internal model. However, the possible effects which internal model quality might have upon crossover model parameters and man-machine performance can be discussed in qualitative fashion.

It has already been demonstrated that subject 3 exhibited more aggressive control behavior in the position command system. As Fig. 9 indicates, this subject's RMS position errors are lower than those for the remaining subjects for all the inner to outer loop automation levels. Also note in Figs. 10-13 that subject 3 has the largest RMS velocity and pitch attitude values, the largest number of control inputs, and, with one exception, the highest subjective task difficulty ratings.

Using the rate command system as an example, these performance variations can be qualitatively reproduced with the crossover models of (4), (6), and (8) using the aggressive command of Fig. 15(a). For example, increasing the ω_{c_x} and ω_{c_u} values by a factor of 1.25, decreasing the preview time constant from 4.0 to 3.75 s to accomodate the larger value of ω_{c_x}, and decreasing the inner-loop time delay from 0.3 to 0.25 s results in a nine-percent decrease in σ_{x_r}, a 26-percent increase in σ_u, a 90-percent increase in σ_θ, and a 17-percent increase in the number of control inputs as compared to the rate command system model performance given in the first column of Table IV. In terms of experimental values, subject 3's performance with the rate command system showed a 26-percent decrease in σ_{x_r}, a 19-percent increase in σ_u, a 78-percent increase in σ_θ, and a 25-percent increase in the number of control inputs as compared to the averages of the mean RMS figures for the remaining three subjects with the rate command system. Although the model parameter adjustments were ad hoc in nature, a fair qualitative comparison exists between model and experimental results. In addition, this favorable qualitative comparison could only be obtained by model parameter variations consonant with increased human operator gains and decreased time delay.

The question arises as to whether the ability of the operator to adopt these higher gains and smaller time delay and improve outer-loop performance is related to higher quality internal models. There is some evidence in the literature that suggests that this may be the case. Levison [14] utilized the optimal control model (OCM) of the human operator to determine the effects of training on model parameters in a single-loop tracking task with K/s-like controlled element dynamics. The model parameters were adjusted via a "quasi-Newton" identification procedure to provide a best match to both RMS tracking scores and frequency-domain human-operator-describing functions. Fig. 21, taken from [14] indicates training effects on experimental and model describing functions. Note that in "late training" the pilot-describing function amplitude is similar in form to that for "early training," except that a substantial increase in gain is evident. This would translate into an increase in crossover frequency. The pilot-describing function in late training exhibits considerably less phase lag at higher frequencies than in early training. Given the similarity between the amplitude curves, this would translate into a significantly smaller time delay for the late training results as opposed to those for early training. The similarity between Levison's results and those reported here is evident. In addition, as Levison points out, these phenomena may well be attributed to differences in the quality of internal models developed by the subjects as training progresses.

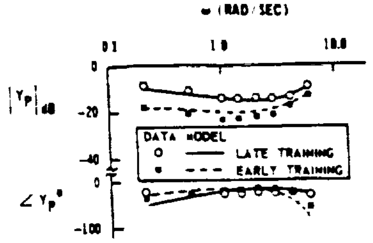


Fig. 21. OCM and experimental human operator describing function data from [14] showing effects of training.

It is difficult to say whether the hypothesized differences in internal model quality reported here can be attributed to training or to an innate ability of the subject in question to develop internal models of higher quality than those of the remaining subjects. Seeking answers to such questions should be a pertinent objective of future research.

VI. Summary and Conclusion

An experimental and analytical study has been undertaken to investigate human interaction with a simple multiloop manual control system, in which the human's activity was systematically varied by changing the level of automation. The control-loop structure resulting from the task definition can be considered as a simple stereotype of a hierarchical control system. The automation philosophy was predicated on a straightforward allocation of tasks between human and machine suggested by existing models of human-machine interaction. The task definition of controlling the longitudinal motion of a hovering helicopter involved a position command that was deliberately selected to be periodic to encourage higher levels of skill development on the part of the subjects, (e.g., precognitive behavior). Finally, very simple representations of human operator dynamics based upon the well-known crossover model were utilized in the analytical effort. The primary conclusions of this study are as follows:

1) In the inner-to-outer-loop automation scheme, defined by the rate, attitude, velocity and position command systems, subjects were able to utilize preview information from the display and to generate signals at a high level of the control hierarchy. The signals represent time optimal inputs to the system defined by the dynamics of the lower levels of the control hierarchy.

2) Simple crossover models of the human operator could provide acceptable qualitative and quantitative matches to experimental data for all automation levels. In the case of the inner-to-outer-loop schemes, the model was forced by a position command significantly different than a square-wave task command. This position command was that used by the subjects in the highest level of inner-to-outer-loop automation (position command system) and can be thought of as a precognitive input existing at the highest level of the human control hierarchy.

3) The number of control inputs generated by the subjects over a run correlated reasonably well with a subjective estimate of task difficulty as the automation level was varied. For reasons discussed in the paper, model-generated control input data correlated better with the subjective ratings than experimental data.

4) The inputs used by one of the subjects in the position command system was considerably more aggressive than that of the remaining subjects. Model results suggested that this behavior could be attributed to this subject having developed a more accurate internal model of the vehicle and task.

Appendix
Automation Systems

Vehicle Equations of Motion:

$$\dot{x} = u$$

$$\dot{u} = -g\theta + X_u u$$

$$\dot{\theta} = K\delta$$

K = control stick sensitivity.

Rate-Command, Attitude-Hold:

δ = subject's control inpout.

Attitude-Command, Attitude-Hold:

$$\delta = \omega_{c_\theta}(\theta_c - \theta)$$

θ_c = subject's control input.

Velocity-Command, Position-Hold:

$$\delta = \omega_{c_\theta}\left[-\frac{\omega_{c_u}}{g}(u_c - u) - \theta\right]$$

u_c = subject's control input.

Position-Command, Position-Hold:

$$\delta = \omega_{c_\theta}\left\{-\frac{\omega_{c_u}}{g}\left[\omega_{c_x}(x'_c - x) - u\right] - \theta\right\}$$

x'_c = subject's control input.

Flight Director:

$$d_{fd} = K_d\left[-\omega_{c_\theta}\frac{\omega_{c_u}}{g}\omega_{c_x}(x_c - x) - \omega_{c_\theta}\frac{\omega_{c_u}}{g}u - \omega_{c_\theta}\theta\right]$$

δ = subject's control input

d_{fd} = flight director command

K_d = display sensitivity.

The flight director design yielded the desirable K/s characteristics [15] in a broad frequency range around expected crossover, i.e.

$$\frac{d_{fd}}{\delta} \approx \frac{K_{fd}}{s}\bigg|_{\omega = \omega_{c_\theta}}.$$

References

[1] J. Rasumssen, "Skills, rules, and knowledge; signals, signs, and symbols, and other distinctions in human performance models," *IEEE Transactions Syst. Man, Cybern.*, vol. SMC-13, no. 3, pp. 257–266, May 1983.

[2] A. Newell and H. A. Simon, *Human Problem Solving*. Englewood Cliffs, NJ: Prenuce-Hall.

[3] S. K. Card, T. P. Moran, and A. Newell, *The Psychology of Human-Computer Interaction*. Lawrence Erlbaum.

[4] D. L. Kleinman, S. Baron, and W. H. Levison, "An optimal control model of human response," *Automatica*, vol. 6, no. 3, 1970, pp. 357–369.

[5] D. T. McRuer, and E. Krendel, "Mathematical models of human pilot behavior," AGARDograph no. 188, 1974.

[6] J. Albus, *Brains, Behavior and Robotics*. Nashua, NH: Byte Books, 1981.

[7] W. B. Rouse, "Human–computer interaction in the control of dynamic systems," *Computing Surveys*, vol. 13, no. 1, pp. 71–99, Mar. 1981.

[8] D. R. Yoerger, "Man–machine performance for a simulated aircraft with multi-level automatic control system," M.S. thesis, Dept. of Mechanical Engineering, M.I.T, May 1979.

[9] R. A. Hess, "Nonadjectuval rating scales in human response experiments," *Human Factors*, vol. 15, no. 3, 1973, pp. 275–280.

[10] W. W. Weirewille and S. A. Connor, "The sensitivity of twenty measures of pilot mental workload in a simulated ILS task," in *Proc. Eighteenth Ann. Conf. Manual Control*, 1982, pp. 150–162.

[11] R. L. Stapleford S. J. Craig, and J. A. Tennant, "Measurement of pilot describing functions in single-controller multiloop tasks," National Aeronautics and Space Administration, NASA CR-1238, 1969.

[12] D. T. McRuer, *et al.*, "New approaches to human-pilot/vehicle dynamic analysis," Air Force Flight Dynamics Laboratory, AFFDL-TR-67-150, 1968.

[13] W. Veldhuyzen and H. G. Stassen, "The internal model—what does it mean in human control," in *Monitoring Behavior and Supervisory Control*, Sheridan and Johannsen, Eds., New York: Plenum, 1976, pp. 157–171.

[14] W. H. Levison, "A quasi-Newton procedure for identifying pilot-related parameters of the optimal control model," in *Proc. Seventeenth Ann. Conf. Manual Control*, 1981, pp. 315–328.

[15] R. H. Klein and W. F. Clement, Application of manual control display theory to the development of flight director systems for STOL Aircraft," Air Force Flight Dynamics Laboratory AFFDL-TR-72-152, 1973.

A Qualitative Model of Human Interaction with Complex Dynamic Systems

RONALD A. HESS

Abstract—A qualitative model describing human interaction with complex dynamic systems is developed. The model is hierarchical in nature and consists of three parts: a behavior generator, an internal model, and a sensory information processor. The behavior generator is responsible for action decomposition, turning higher level goals or missions into physical action at the human–machine interface. The internal model is an internal representation of the environment which the human is assumed to possess and is divided into four submodel categories. The sensory information processor is responsible for sensory composition. All three parts of the model act in consort to allow anticipatory behavior on the part of the human in goal-directed interaction with dynamic systems. Human workload and error are interpreted in this framework, and the familiar example of an automobile commute is used to illustrate the nature of the activity in the three model elements. Finally, with the qualitative model as a guide, verbal protocols from a manned simulation study of a helicopter instrument landing task are analyzed with particular emphasis on the effect of automation on human–machine performance.

I. Introduction

In SPEAKING of human–machine interaction, it is commonplace now to find the human as "controller" being supplanted by the human as "manager." Research aimed at developing mathematical models of human–machine interaction has been increasingly directed toward modeling the higher supervisory activities, e.g., [1]. While quantitative models are of definite use in this area [2], the importance of qualitative representations cannot be ignored [3], [4]. Thus, as pointed out by Rasmussen [4], rather than a single integrated quantitative model of human behavior, an overall qualitative model may be more desirable. This model can then serve as a framework in which to incorporate a number of more detailed and preferably quantitative models.

The purpose of the research to be described is to develop such a qualitative model. As will be seen, the model is based upon an hypothesized internal representation of the environment called the "internal model" (IM) which serves as an active link between a "behavior generator" (BG) and a "sensory information processor" (SIP). The human's well-documented preference for certain dynamic systems will be discussed, as will human error, both in reference to the internal world model. An example of an automobile commute is used to illustrate the activity in these model elements. Finally, a verbal protocol experiment involving a pilot-in-the-loop simulation with an advanced digital avionics system will provide an opportunity to interpret pilot behavior in terms of the model which has been developed.

II. A Model of Human Interaction with Complex Dynamic Systems

As used here, a dynamic system will refer to one whose state can change in time without human intervention [5]. This definition excludes such systems as text editors, etc., whose output time dependency depends exclusively on human input. A dynamic system is said to be complex to the extent that the human can observe it in nonequivalent ways, in different levels of abstraction, all of which are pertinent to the system operation [6], [7]. The ability of the output of a dynamic system to evolve without explicit human input inevitably forces the human controller or supervisor to "keep ahead" of the system in successfully completing any realistic task [7]. The requirement for keeping ahead of the system leads to anticipatory, as opposed to purely reactive, behavior [6]. Anticipatory behavior, in turn implies the ability of the human to predict future system output on the basis of present system state and present and future input. This all leads somewhat naturally to the topic of internal models and to the model which is the subject of this research. It is interesting to note that nearly three decades ago, Kelley [8] made a strong case for the importance of anticipatory control in man–machine systems. Indeed, he forcefully argued that it is the *future* state of a dynamic system, not the past or present state, that is the prime concern of a human controller. Perhaps the success of feedback models of the human controller in explaining many human–machine dynamic phenomena [9], has discouraged active research on the topic of anticipatory behavior. However, the comparative complexity of the systems now evolving which are to be under human control and supervision is likely to change this picture [10].

Fig. 1 is a diagram of the primary elements of the model for human interaction with complex dynamic systems. The model consists of a behavior generator, an internal model, and a sensory information processor. All three elements are hierarchical in nature and the internal model serves as

Manuscript received February 4, 1986; revised July 12, 1986. This work was supported by the National Aeronautics and Space Administration under Grant NAG 2-221.

The author is with the Division of Aeronautical Science and Engineering, Department of Mechanical Engineering, University of California, Davis, CA 95616, USA.

IEEE Log Number 8610539.

0018-9472/87/0000-0033$01.00 ©1987 IEEE

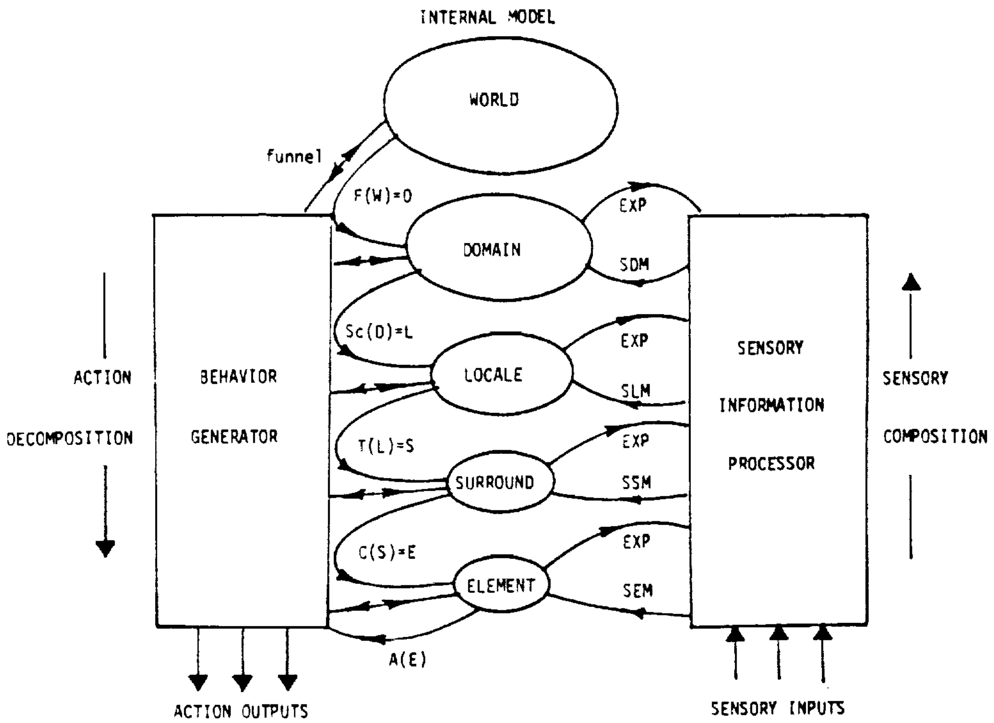


Fig. 1. Model of human interaction with complex dynamic system.

a link between the two elements responsible for sensing (the SIP) and action (the BG).

A. The Internal Model

The internal model is a volatile internal spatial/temporal representation of the environment which the human is assumed to possess and use when interacting with complex dynamic systems. The idea of a human possessing an internal or mental model is certainly not a new one, e.g., [1], however, the interest in using such a construct to explain human behavior seems to be growing, e.g., [4], [6], [12]–[18]. As shown in Fig. 1, the IM is equivalent to a world model (the terms may be used interchangeably) which has been divided into four submodel categories. The nature of the submodels changes from a broad representation of the environment to a narrow one in moving from the domain to the element categories.

Fig. 1 indicates activity occurring between the IM and the BG and SIP. On the left, relations like $F(W) = D$ are indicating transformations in which the submodels at higher levels in the hierarchical structure are being transformed into submodels at lower levels through interaction with the BG. (Symbols are defined in the Nomenclature at the end of the paper.) Indeed, such transformations constitute the principle activity of the BG and will be discussed further herein. These transformations are assumed to occur at discrete instants of time but with increasing frequency as one moves down the hierarchy. The latter frequency characteristics are typical of any hierarchical control system [19].

As shown in Fig. 2, the IM can be described in more concrete fashion as a problem space of large dimension through which a trajectory passes with implicit time dependence representing the dynamic relationship between the many variables which define the human's internal representation of the environment at various levels of detail or abstraction. The trajectory represents past and present states of the world model. Now the world space is transformed into a dimensionally smaller subspace called the domain space via a transformation $F(W) = D$. The domain space will also contain a trajectory. The domain space and trajectory define the domain model which is viewed with a time scale T_D, as shown in Fig. 2. This scale represents a smaller scale than that of the world model, i.e., a unit length of the domain trajectory involves less elapsed time than a unit length of the world trajectory. The domain space is transformed into yet a smaller subspace called the locale space via a transformation $Sc(D) = L$. The locale space will contain a trajectory and the locale space and trajectory define the locale model. The locale model is viewed with a smaller time scale still: T_L. As Fig. 2 indicates, the transformations continue, with the last transformation $A(E)$ denoting an action output of the human. The nature of these transformations will be discussed in the next section.

At this point the question may arise as to the number of categories of models which have been discussed, i.e., four. Why not ten, or two? To answer this, one must recognize, as Rasmussen has [20], that the model decomposition and change of abstraction implied by the model categories just discussed is the principle means by which a human copes with complexity. Four categories were felt to be a *minimum* number to describe and stratify human interaction with a complex system adequately. Indeed, in any given situation, many more categories may exist. This is allowed

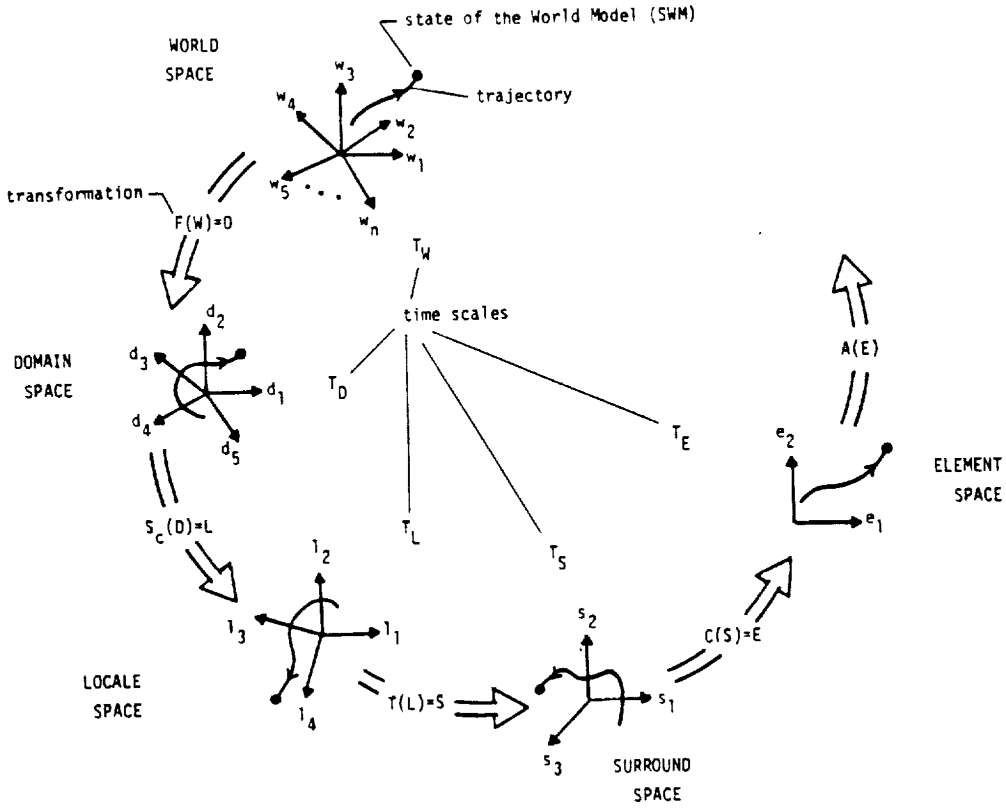


Fig. 2. Internal model generated by transformations of world model space.

in the present framework, of course, through the existence of subdomains, sublocales, etc.

B. The Behavior Generator

The hierarchical nature of the BG is evident in Fig. 3. Here, a detailed breakdown of the activity hypothesized to occur in the BG is provided. The major elements are 1) the planner/fault manager, consisting of a framer and scripter, 2) a tasker, 3) an executor/monitor, and 4) an actuator. The latter three elements constitute a metacontroller, a term coined by Sheridan in a manual control context [21]. Actually, a higher level exists in the BG, that of a metaplanner. This implies an activity concerned with making plans about plans [22], [23]. While a very important issue in artificial intelligence research, the activity of the metaplanner will not be discussed here. Rather, we assume that a product of the metaplanning activity, the mission subphase, constitutes the top level in the behavior generating hierarchy. Fig. 4 shows that the behavior characteristics of the BG can be interpreted in terms of the knowledge-, rule-, and skill-based behavior discussed by Rasmussen [4], and the recognition/classification, planning, and execution/monitoring levels for a human problem solver offered by Rouse [24].

Planning activity, particularly that of human pilots, is receiving increased attention in the literature, e.g., [25], [26]. The planning which is hypothesized to take place in the BG shown in Fig. 3 is more akin to that of a "skeletal planner," wherein a plan is selected which already contains basic steps. In the context of the BG of Fig. 3, the skeletal plan is instantiated by frames and scripts which interact with the domain and locale models. The frame was introduced by Minsky as a basis for understanding complex human behavior like natural-language dialogues [27]. Scripts are framelike structures developed by Shank and Abelson for representing sequences of events [28]. The use of frames and scripts by pilots as means of avoiding more abstract planning has been suggested by Johannsen and Rouse [26], and it is this interpretation which is exploited here. The action of selecting and monitoring scripts in the scripter can be referred to as "time-driven" planning while the action of changing scripts because of an unanticipated situation (a change in the world model) can be referred to as "event-driven" planning [26].

The first element in the framer is a mission subphase or fault detector. This indicates the point in the behavior generating hierarchy where the existence of a mission subphase or a system fault has been made part of the knowledge base. While the mission subphase is assumed to be generated in the metaplanner, the possibility of faulty system operation is assumed to arise from activity in the executor/monitor in the metacontroller. The first elements in the scripter, tasker, and executor involve event sequence or subtask detection and play a role similar to that of the first element in the framer. Now one can see that it is the frame, script, task, control, and action selection which defines the successive world model transformations performed by the BG and shown in Fig. 2, and which repre-

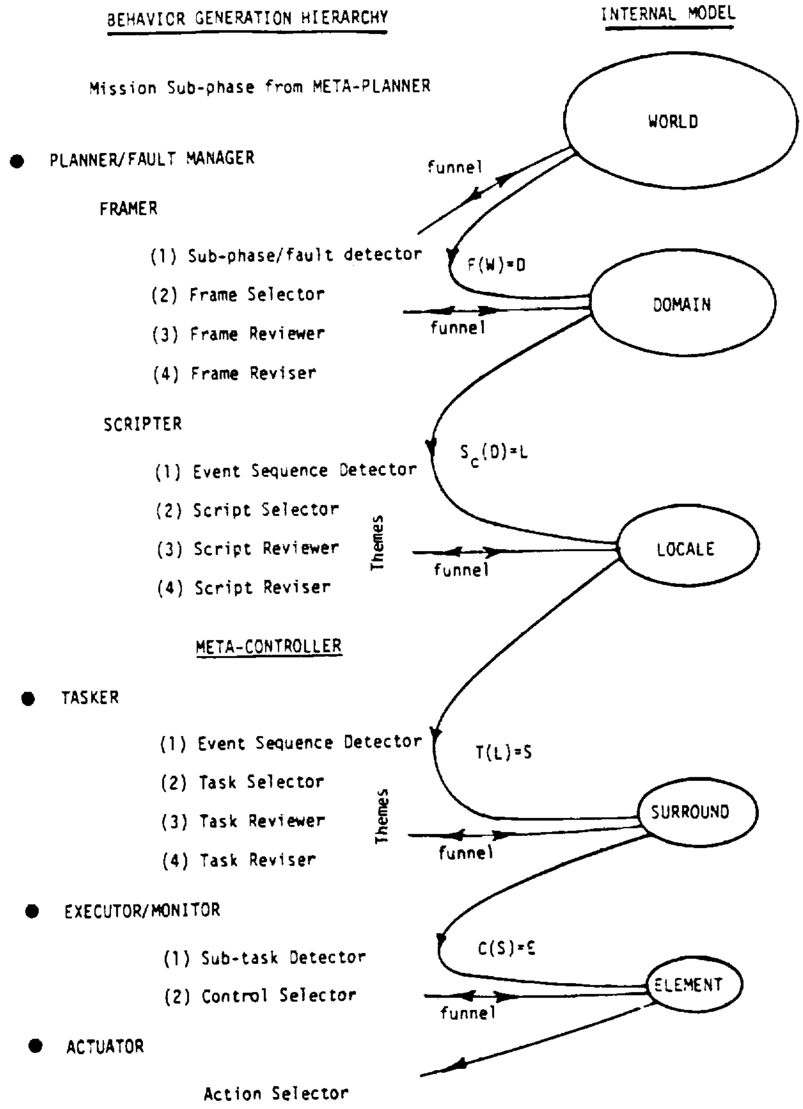


Fig. 3. Behavior generator hierarchy.

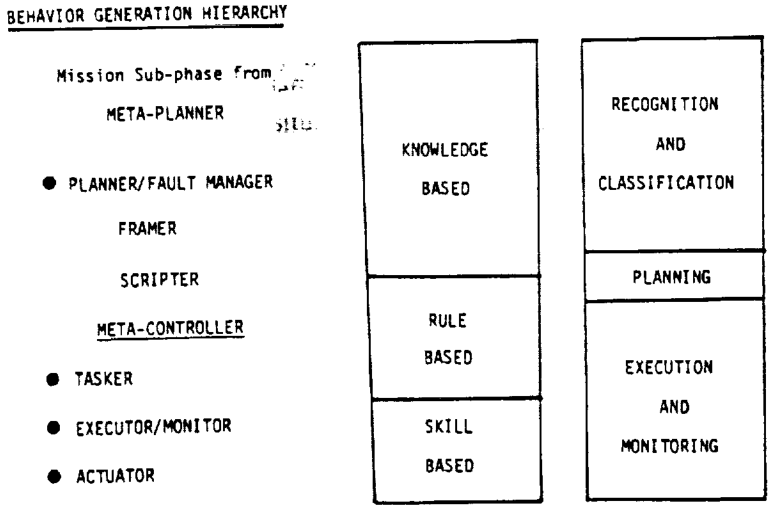


Fig. 4. Interpreting BG activity in terms of taxonomies of [4] and [24].

sent human knowledge-, rule-, and skill-based activity in interacting with complex dynamic systems.

Frame, script, task, control, and action selections occupy operationally similar places in the behavior-generating hierarchy. Note that control selection can involve continuous, discrete, verbal, or manual interaction with the dynamic system. In the parlance which describes continuous manual tracking tasks, the control selection will result in interaction which can be categorized as precognitive, pursuit, or compensatory [29]. The particular type of interaction which results depends upon the kind of variables in the element space which are created by the control transformation $C(S)$. If variables classified as system error signals are created, then the activity can be classified as compensatory. In the other extreme, the control transformation may bypass the element space entirely and produce an action output directly, i.e., $C(S) = A$. This transformation would describe precognitive behavior. Note that definite performance advantages accrue in this latter case as the SIP inputs necessary to define the error variables in the element space are obviated, as are the transformations from the element space to action output. However, certain types of errors may also result [30].

The activities involved in the second step in each division of the BG could be involved with maximizing, say, a subjective expected utility of candidate frames, scripts, tasks, controls, or actions. Sheridan has proposed such a maximization scheme in a model for supervisory control which also uses internal models [31], [32]. However, it is more likely that the human "satifices" rather than maximizes [25], and this is the idea we will adopt here.

In terms of the informal mathematical structure introduced so far, the script, task, control, and action activities are hypothesized to evolve as follows. The world space, mission subphase, and the present state-of-the-world model (SWM) define what will be called a "trajectory funnel" in the world model n space as shown in Fig. 5. Future time is the implicit variable in this funnel. A funnel shape has been deliberately chosen to emphasize that predictions of future world trajectories becomes increasingly imprecise as future time increases. The SWM obtained from an SIP input anchors the narrow tip of the funnel in the WM space and represents knowledge about the world at a particular instant [33]. The human then selects a frame which, in previous encounters with similar WM funnels, has eventually led to a world trajectory within the funnel. Once such a satisficing frame has been selected, the transformation $F(W) = D$ can be completed and the domain space is created. Now, the domain space, the frame, and the present state-of-the-domain model (SDM) defines a trajectory funnel in the domain space. Again, an input from the SIP giving an SDM anchors the narrow end of the funnel in the domain space. With the domain funnel established, the human then selects a script which, in previous encounters with similar DM funnels, has eventually led to a domain trajectory within the funnel. Once the script has been chosen, the transformation $\mathrm{Sc}(D) = L$ can be completed, and the locale space is created. The script, locale space, and present state-of-the-locale model (SLM) define a trajectory funnel in the locale space. Again, an input from the SIP anchors the end of the funnel.

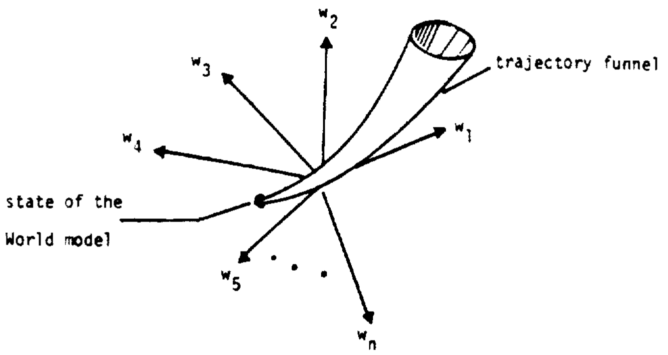


Fig. 5. Trajectory tunnel in world model space.

The process just described continues down to the action level. Note that the number of transformations per unit time will increase as one goes down the model hierarchy. In addition, many different model spaces within any one category can be defined, e.g., many surround spaces. The role which training and experience play in this process is obvious: both determine the human's ability to select rapidly appropriate frames, scripts, tasks, controls, and actions. This may explain how humans develop "expertise" through concrete training and experience which allows them to interact with complex dynamic systems at all levels of the behavior-generating hierarchy in the fluid manner characteristic of anticipatory behavior. This attempt to describe human decisionmaking can also obviously be approached from the standpoint of fuzzy set theory [34]. Indeed, such models have been derived for human fault diagnosis tasks [35] and for more skill-based tasks, such as automobile driving [36].

As the names imply, the frame and script reviewers are activities in which the human reviews or rehearses a frame or script before it is actually used in a transformation. This is an important step since faulty transformations at higher levels of the behavior-generating hierarchy can have serious consequences at lower levels, as will be seen. The script reviser allows changes or deletions in the selected script, possibly because of conflicts with themes. A theme has been defined as something that gives rise to a goal in a given situation [18]. It can be thought of as a general behavior rule which is always in existence, as opposed to specific event sequences called out in scripts. As such, themes can be represented by "forbidden" regions in the locale and surround spaces. The task reviewer plays a role similar to the script reviewer. Finally, the task reviser allows changes in the task chosen by the task selector. As in the case with script revision, task revision may be the result of theme conflict.

The actuator in the executor/monitor part of the metacontroller is responsible for physical action, i.e., the human output. It is the means by which the human imparts his will to the machine. The actuator forms the lowest level in a hierarchy which decomposes goals into physical action.

In terms of the hierarchy, the monitor exists on the same level as the executor. No delineation of the minitor will be undertaken here, however.

A final point should be made here. The trajectory funnels which the human is hypothesized to create in the IM spaces can be viewed as a generalization of Rasmussen's concept of *symbols*, *signs*, and *signals* [4]. At the top of the hierarchy, the funnels can be likened to symbols, at the bottom, to signals, with signs occurring in between. The symbols project relatively far into future space as compared with signals. All three are intimately related to human behavior. The funnels differ from symbols, signs, and signals as discussed in [4] in that the latter are more closely allied to present time sensory information, whereas the former are related herein to future time IM characteristic which allow anticipatory behavior.

C. The Sensory Information Processor

No detailed breakdown of the hypothesized activity of the sensory information processor will be attempted here. As with the BG, activity between the IM and the SIP is discrete and takes place at different frequencies. While the hierarchy of the BG is responsible for action *decomposition* as one moves from top to bottom, the SIP hierarchy is responsible for sensory *composition* as one moves from the bottom up. Each level of the SIP processes the data from lower levels and, with the help of the IM, extracts features and recognizes patterns. Again, the IM trajectory funnel created by frames and scripts in the BG plays an important role here by allowing anticipatory SIP behavior. This is indicated in Fig. 1 by "EXP" indicating expectations being provided the SIP. Information relevant to the IM at appropriate levels in the dual hierarchy of Fig. 1 is provided via state-of-the-IM updates. In terms of Fig. 3, IM updates refer to locating the particular point in the model space corresponding to present time. The partially processed sensory data that remain are then passed to the next higher level in the SIP hierarchy. Albus [16] describes the activity of a structure similar to that of Fig. 1 in equivalent terms.

D. Integrated Human Activity

The trajectory funnels at all levels of the human model hierarchy represent implicit commands to lower levels, i.e., they are in essence saying, "do what is necessary at *lower* levels to cause the state of the world to move along the axis of trajectory funnel existing at *this* level." At the action selector of the metacontroller, the human's manipulative output attempts to bring this about. Note that the lower limits of our model, where the action output $A(E)$ occurs, future time does not exist. No trajectory funnel is created beyond the element space, since no space, as such, is assumed to exist. Therefore, physical human output as a continuous function of time is created by allowing the action to define a new point in the output space. Compared to higher levels in the hierarchy, very frequent inputs from the SIP update the state-of-the-element model (SEM) at the action level and cause macroscopic continuity in human action output. Of course, the SIP has limits as far as the frequency of operation goes. For example, for the visual system, this limit would be the cycle time of the "visual processor" with a value on the order of 100 ms [37]. Soon in this process, but less frequently than SEM updates themselves, the trajectory in the element space moves far enough into the wider (less certain) part of the element trajectory funnel that the BG decides that a new element funnel needs to be generated (or perhaps an entirely new element space). Failure to generate new funnels/spaces with appropriate frequency constitutes a particular type of human error which will be discussed in more detail in Section V. Possible criteria for generating new funnels/spaces will not be explored here but may well depend upon minimizing the errors just mentioned. The new funnel/space is defined by the *existing* task, the *existing* surround space, but a *new* update of the SSM. The process continues, and of course the trajectory in the *surround* space soon moves far enough into the wider part of the surround funnel that the BG takes action and, with help from the SIP, defines a new surround funnel. One can see how the process propagates up the hierarchy and continues until the mission subphase is completed.

E. Human Performance Models

Existing quantitative human performance models can be interpreted in terms of the qualitative model of Figs. 1 and 3. For example, successful as they have been in modeling human operator behavior, feedback control models such as the crossover and optimal control models [9] describe activity only at the level of the control selector in the hierarchy which has been described. Extensions of the optimal control model which treat human monitoring behavior [39] and dynamic decisionmaking [40] move further up the hierarchy but only to the level of the task selector. Even then, the model decisionmaking predictions which have been experimentally verified have involved competing tasks which are very similar in nature. Baron *et al.* have developed a procedure-oriented crew model (PROCRU) which is an analytic/computer model of the activities of the crew of a representative transport aircraft in a nominal category I instrument landing system (ILS) approach [41]. This model is quite complex and employs a procedure selection scheme which is related to the subjective expected utility approach mentioned in Section II-B. PROCRU can be considered to describe human activities encompassing all the metacontroller activity as can the human operator simulator [42], which incorporates very detailed micromodels of human manipulative activity.

Other human performance models have been developed which represent applications of artificial intelligence concepts to manual control problems. Doring and Knauper [43] have, for example, developed a model which utilizes a production system [44] for describing pilot behavior in an ILS landing approach. Anzai also uses a production system to model the steering behavior of the helmsman of a large ship [45]. This research is of interest since it parallels similar modeling efforts which have their origins in manual control theory [41], [13]. The situation-action pairs constituting the production rules in the former models essentially describe the activity of the tasker in the metacontroller of Figs. 1 and 3.

The preceding discussion is an attempt to show that the qualitative model being presented here can provide a framework in which a number of quantitative models can be interpreted. In briefly reviewing the capabilities of some existing quantitative models of human performance in light of the proposed BG, it would appear that model capabilities are limited to the activity of the metacontroller of Fig. 3. This is not intended as a criticism of these models, but rather as an affirmation of the difficulty of the modeling task at hand, i.e., human interaction with complex dynamic systems. Indeed, as pointed out by McDermott [46], the modeling complications which arise when one allows the system state to evolve continuously without human intervention are not often addressed by AI researchers.

III. Human Workload

An impressive amount of research in human-machine interaction has been devoted to the subject of human workload [47]. In terms of the model proposed here, the following workload hypothesis is offered: in any task involving human interaction with a dynamic system, any exogeneous constraints in accessing the internal model above the level of the metacontroller in the behavior generator give rise to human concern for workload.

This hypothesis is based upon the simple tenet that successful mission subphase completion, which is the goal of the BG, will demand anticipatory behavior, which in turn requires effective operation of the higher levels of the BG hierarchy as outlined in Section II-B.

Since the problem of aircraft navigation and control will be discussed in Section VII, the following hypothesis regarding handling qualities is relevant: handling qualities are perceived in a manner inversely proportional to the utilization of element models of the internal model, i.e., the greater the utilization of the element models, the poorer the perceived handling qualities. As such, poor handling qualities are an exogenous constraint in accessing, updating, and utilizing higher levels of the world model, and thus contribute to workload.

Utilizing element models means 1) defining trajectory funnels in the element space, 2) updating the SEM with inputs from the SIP, and 3) creating action output points. This really amounts to element space "processing demands." This can be generalized to processing demands for any part of the IM, as shown in Fig. 6. In Rasmussen's terminology [4], the shaded lines in Fig. 6 would represent instantaneous signal and/or sign processing demands in the IM categories noted. The thickness of the lines passing through the IM categories is intended to portray graphically the amount of the processing demands being required at this instant; e.g., the instantaneous demand being required for the element model is denoted D_E.

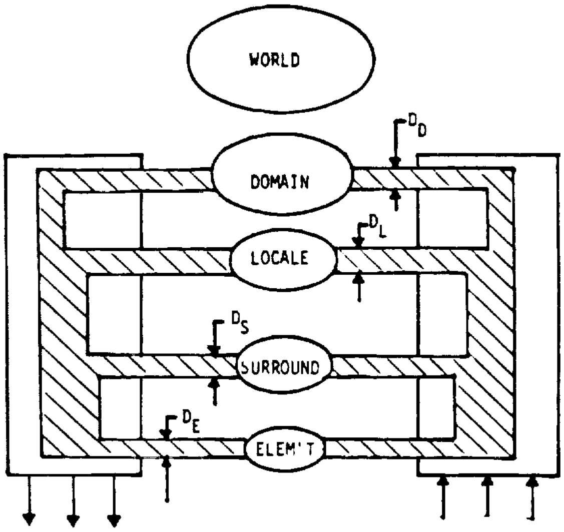


Fig. 6. Internal model processing demands.

This discussion can be formalized by hypothesizing that the human possesses a processing demand capacity D_C, where $D_E + D_S + D_L + D_D \leqslant D_C$ at all times and D_C is a function of human motivation and training. This idea is similar to one proposed by Senders [48]. One can then postulate that

$$\text{workload } \alpha \frac{[D_L + D_D]}{D_C - [D_E + D_S]}. \tag{1}$$

Likewise, one can postulate

$$\text{handling qualities } \alpha \frac{D_C - D_E}{D_C}. \tag{2}$$

Note that "high" workload and "poor" handling qualities are reflected by the right-hand side of (1) approaching unity and the right-hand side of (2) approaching zero, respectively. The metaphor of a time-shared computer has considerable merit here. The processing demands are, of course, time varying, and the shaded lines of Fig. 6 can be thought of as widening and narrowing throughout the interaction in question. For specific tasks making up the mission subphase, average processing demands could be considered reflecting "average" workload. Note that even monitoring activity will be an exogenous constraint since the monitor interacts with the element submodel category of the world model in Fig. 3.

Equations (1) and (2) suggest that instantaneous workload and handling qualities can be changed in a variety of ways. For example, assuming that the terms within brackets in the numerator and denominator of (1) remain constant, workload can be reduced by an increase in D_C possibly brought about by increased training and/or motivation. Given a constant D_C and numerator value in (1), workload would be increased by an increase in D_E, brought about by, say, a stability augmentation system failure in a flight control system. This would also bring about handling qualities deterioration as evidenced by (2).

IV. Automation-Induced Complexity

Automation refers to allocating to machines (which includes computers) the responsibility for tasks which formerly were the responsibility of humans. In discussing aircraft piloting tasks, Wiener and Curry [49] list three driving factors in cockpit automation which are also valid in explaining the appearance of automation in other human–machine systems: technology, safety, and economics. Part and parcel of the technological advances which allow automation to provide increased safety and improved economics is increased system complexity. In discussing process control automation, Crossman [50] points out that the introduction of automatic control of process variables reduces the amount of routine work to be done by the operator but considerably complicates the decisions he must make. In addition, automated system which support human interaction with dynamic systems typically automate specific functions and consequently possess a good deal of flexibility at the task level. Thus in terms of the model of Fig. 3, internal models within the metacontroller can become numerous with concomitant increases in training requirements (to hone the ability to make the numerous transformations implied by $T(L) = S$ and $C(S) = E$ efficiently) and increases in the frequency of inputs from the SIP (to update the SLM's and SEM's). The problem can become particularly acute in automated cockpits. Consider Table I, taken from [51], which lists typical modes in a modern automatic flight control system (AFCS). Quoting from [51]:

> "The Autopilot and the Flight Director display are wholly available for selection by the pilot. Two separate selections must be made, the first determines the parameter to control the aircraft pitch..., the second to control roll.... In addition to the selection of the immediate control parameter it is possible to select a value to be acquired and maintained in the future. The pilot will normally select 'Altitude Hold' once he has achieved his cruising level. Depressing the appropriate push-switch will cause this mode to be displayed on an indicator. Thereafter, the AFCS carries out all the movements necessary to maintain constant altitude. In the event that the pilot wishes later to alter his altitude, the sequence of actions might be:
>
> 1. Rotate knob to select desired height in Acquire display, 2. Select 'Vertical Speed' as the pitch mode, 3. Rotate knob to select required rate of climb or descent."

TABLE I
Typical Modes Available in an Automatic Fight Control System (from [51])

Hold Facility		Acquire Facility	
Pitch	Roll	Pitch	Roll
attitude	attitude	altitude	heading
air speed	heading	glideslope	inertial nav.
mach number	wings level		localizer
vertical speed	inertial navigation		
altitude	localizer		
glideslope			

The AFCS and tasks just outlined are representative of AFCS operation, even for advanced systems [52]. The AFCS and its operating procedures constitute surround and element submodels for the transport pilot. By this is meant that specific subspaces and trajectories/funnels at the surround and element levels of the model of Fig. 1 need to be created which deal soley with a pilot-centered description of the AFCS and the evolution of its state over time. One sees that Crossman's statements about process control automation also apply to modern aircraft cockpits. It may be apparent at this juncture that automation carries with it the seeds for disaster. The central thesis of the model being discussed herein is that anticipatory human behavior is essential for successful human interaction with complex dynamic systems. Further, it is hypothesized that anticipatory behavior of a quality consistent with acceptable human/machine performance demands hierarchical behavior generating and sensory processing structures which employ an IM. While automation can and does relieve workload caused by processing demands at the actuator level, it does so at the price of an increase in the number of element and surround models. At certain times in a mission subphase completion, this could lead to *increased* workload attributable to a sharp increment in the demands at the tasker level. Curry [53] has noted that airline pilots using automated flight control systems often complain of such increased workload. However, more frequently, automation-induced complexity leads to the commission of serious human errors which are the subject of the next section.

V. Human Error

Considerable effort has been expended by psychologists and engineers in the study of human error, e.g., [30], [54]–[57]. For the purposes of this discussion, we shall define human error as an inconsistency with a predetermined behavioral pattern used in establishing system requirements, specifications, and the resulting system design [30]. Of particular interest is the production of "grievous" human error, which can be defined as a human error which involves exceeding safe operating tolerances [30]. Various human error taxonomies have been proposed in the past. A traditional classification is fourfold [58]: 1) failure to perform a required activity, 2) incorrect performance of a required activity, 3) performance of a required activity out of sequence, and 4) performance of a nonrequired activity. Norman [55] offers a simple but useful classification of human error as either mistakes or slips, where a mistake implies an incorrect intention and a slip implies a correct intention but incorrect execution. Singleton [59] discusses a dichotomy of errors often used by system analysts: formal and substantive, where the former refers to an error where rules have been broken and the latter to an error involving nonintended performance. Rouse [57] outlines three key elements of human error: 1) misunderstandings, 2) incompatibilities, and 3) catalysts

Misunderstandings mean inadequate, inaccurate, or misleading information either from an individual's own knowledge base or from the system involved. Incompatibilities refer to fundamental mismatches between task characteristics and human abilities and limitations. Finally, catalysts refer to conditions which themselves do not cause errors but which provide an environment in which errors are more likely to occur.

The purpose of this section is to view human error within the framework provided by the model of Figs. 1 and 3. Now, it is almost axiomatic from the discussions of Sections II–IV that human error is synonymous with faulty operation of the BG, IM, and SIP triune of Figs. 1 and 3. The sources of such faulty operation can be summarized in surprisingly few categories, referred to here as cardinal errors. They are

1) incorrect definition of a trajectory funnel (or action output point) given the space, the mission subphase, frame, script, etc., and the SWM, includes incorrect recognition (or ignoring) of "forbidden" regions in model space corresponding to themes;
2) transformation of a model space using an incorrect or inappropriate frame, script, task, etc.;
3) incorrect transformation of a model space given a trajectory funnel;
4) anchoring new trajectory funnels or generating trajectory with incorrect input from SIP;
5) failing to define new trajectory funnels at an appropriate rate.

These cardinal errors are ordered in terms of their impact on the operation of the hierarchy of Fig. 1. In addition, however, one must remember that the severity of a human error is also dependent upon the level of the hierarchy in which it occurs. The higher levels affect behavior at all lower levels, and the frequencies at which activity occurs at higher levels is lower than at lower levels. This means it takes longer to correct errors propagated at higher as opposed to lower levels of the hierarchy.

Clearly, the cardinal errors define faulty operation of the hierarchy of Fig. 1 which, in turn, is an immediate precursor to "erroneous" human performance. Given a specific situation, one can postulate an error which clearly fits into one of the taxonomies outlined in the introductory paragraph of this section and trace it to the commission of one of the cardinal errors just enumerated at a particular point in the hierarchy of Fig. 1. As an example of the latter, consider the case of a pilot deliberately descending below the minimum descent altitude in an instrument landing system landing approach. This fits our original definition of an error and can be classified as 1) incorrect performance of a required activity (initiating a go-around), 2) a mistake, 3) a formal error, and 4) a misunderstanding, i.e., the pilot has not been sufficiently trained as to the dire consequences which often accompany such an action. Now this error can be traced to the first cardinal error, ignoring a forbidden region of the IM space corresponding to a theme in a locale space. Here the theme would (*should*) be, "Do not descend below published minimum descent altitude on instruments!"

Finally, note that human errors can be a source of workload [60]. This can be appreciated by considering a case where the first cardinal error has been committed at a high level (domain or locale) in the hierarchy of Fig. 3. Assuming that the error is detected (through the activity of the monitor), a considerable surge in processing demands D_D and D_L can ensue with concomitant increases in workload (see (1)).

VI. An Example

We will now consider an example of human–machine interaction which illustrates some of the concepts which have been discussed thus far. The example will involve the familiar activity of automobile driving. A similar example was used by Johannsen and Rouse [25] to describe the variety of human activities which occur in realistic human–machine interaction. As will be seen, this example is really more of a "gedanken" experiment using the model proposed in the preceding sections.

Fig. 7 is a sketch of a map showing the nominal automobile commuting route to be discussed. The nominal route from A to J is some 15 km in length, with the stretch B–I occurring on a major six-lane highway, referred to here as highway BI. The commute is assumed to take place in a typically crowded urban setting and is patterned after a drive the author took daily from NASA Ames Research Center to Stanford University in a recent summer. Let us interpret the hypothesized trip in terms of the model of Fig. 3. Let us assume that the driver is seated in his car in a parking lot at point A, with his seat belt buckled and the ignition key in his hand.

The mission subphase emanating from the metaplanner in the BG could be succintly summarized as "transport self from point A to point J in own car." The world space, mission subphase, and the present SWM define the trajectory funnel in the world space. Given the time scale and number of variables involved, of course, this trajectory is, itself, unsuitable for generating action output. The SWM anchors the funnel in the world space. This point might be a description of the environment one could perceive while sitting in the car in the parking lot, commensurate with the scale of the world model, i.e., very large, encompassing little more in detail than night or day and very approximate car location. The driver now can select a frame which, through past daily commutes, has led to a world model trajectory within the funnel. A concise verbal description of this frame might be simply "short commute." However, the frame itself is a more complex entity than just a two-word phrase. It serves to map the world model space into an appropriate dimensionally smaller time-scaled domain space. The domain space can be thought of as consisting of all the variables necessary to describe the environment along the nominal and alternate routes shown in Fig. 7 with a detail commensurate with the time scale T_D.

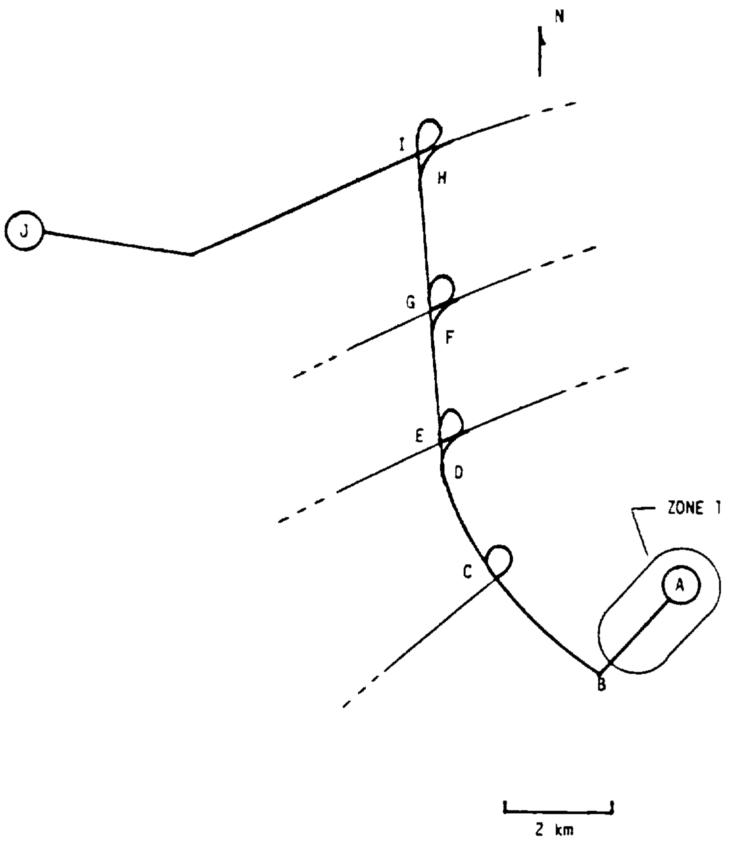


Fig. 7. Map of routes pertinent to automobile commute.

As a symbolic representation, we can refer to the map of Fig. 7 as the domain space. Now the short commute frame, the domain space, and the present SDM define a trajectory funnel in the domain space. Once again, this funnel is, itself, unsuitable for generating action output. The SDM anchors the funnel in the world space. This point might describe the parking lot environment on a scale commensurate with the map in Fig. 7, i.e., still broad in scope but now delineating things like the present direction the car is facing, etc. The driver can now select a script, which in past daily commutes has led to a domain model trajectory within the funnel. A concise verbal description of this script might be, "drive to destination J via highway BI." It is interesting to note that the short commute frame has led to a different initial script than would have been in evidence had, say, a long trip frame been selected. The latter may well have led to a domain funnel which yielded a script summarized as, "drive to service station and have car checked." Pursuing this a bit further, let us suppose a long trip *was* the mission subphase, and the transformation from world space to domain space was correct, as was the funnel definition. Suppose, however, that instead of the "drive to service station" script, the driver selected "drive to destination L along highway BM." Somewhere on the way to L, the driver might find the car radiator boiling over for lack of coolant (which would not have happened on the short commute due to the length of the trip). This could strand the driver in the middle of a desert if highway BM traversed once. This is a serious consequence brought out by the commission of a cardinal error high in the hierarchy of Fig. 3. The reader will see that it was cardinal error 2.

Getting back to the commuting example, the selected script transforms the domain space into an appropriate dimensionally smaller time-scaled locale space. The locale space can be thought of as consisting of all the variables necessary to describe the environment along the first portion of the nominal route of Fig. 7, with a detail commensurate with the time scale T_L. The script, locale space, and present SLM define a trajectory funnel within the locale space still unsuitable for directly generating action output. The funnel is anchored by the SLM. This point now represents the parking lot environment delineating things like the location of the car relative to a parking lot exit on a street with traffic allowed in a favorable direction. As a symbolic representation, we can think of the locale space as "zone 1" of Fig. 7, remembering again that the locale space is more complex than a two-dimensional map.

The driver can now select a task, which in past daily commutes has led to a locale model trajectory within the funnel. A concise description of this task might be, "leave parking lot in direction appropriate for getting on Highway BM." This task now transforms the locale space into an appropriate dimensionally smaller time-scaled surround space which can be thought of as consisting of all the variables necessary to describe the environment within the car and its immediate vicinity with a detail commensurate

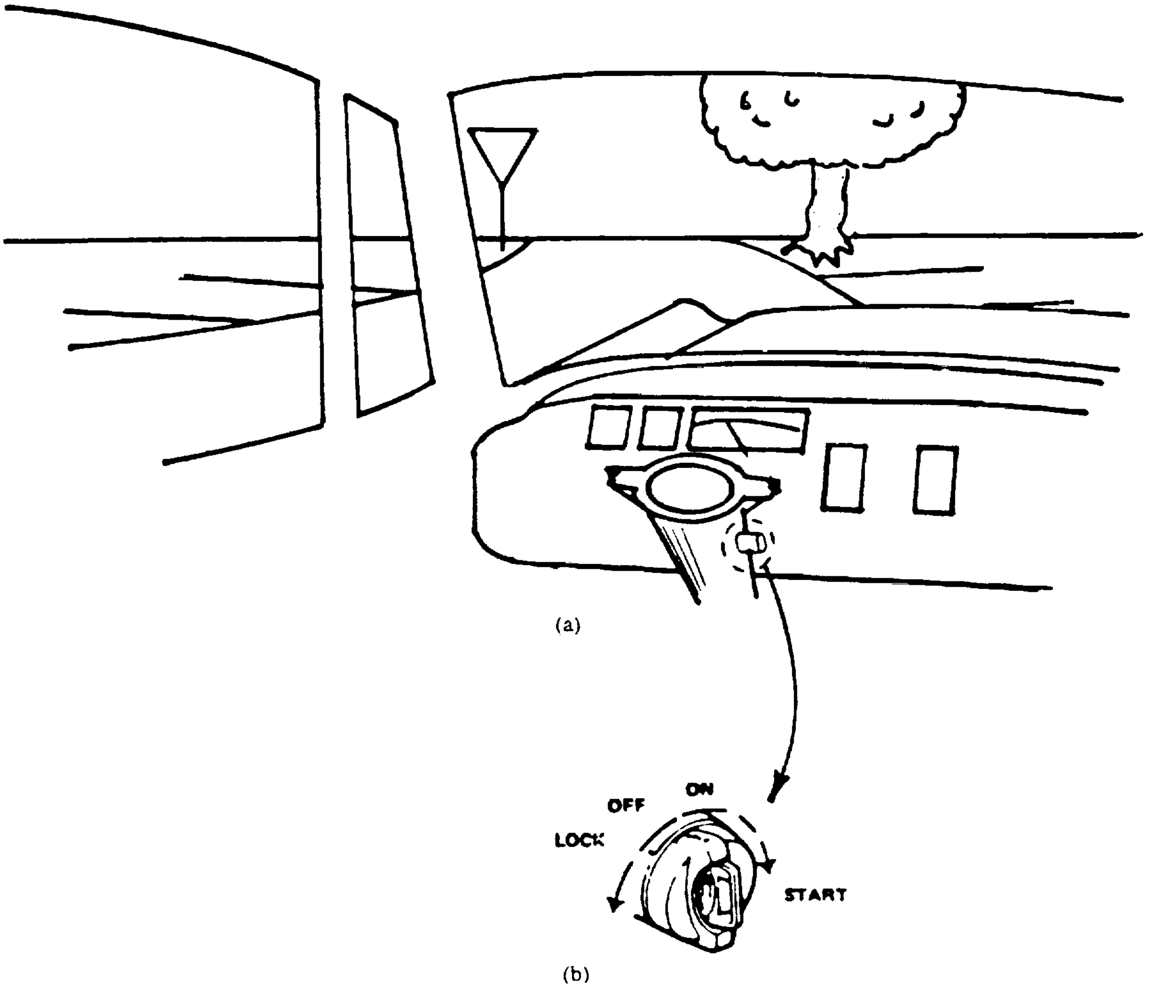


Fig. 8. Representations of initial stage of automobile commute. (a) Surround space. (b) Element space.

with the time scale T_S. The task, surround space, and present state-of-the-surround model (SSM) define a trajectory funnel within the surround space not yet suitable for directly generating action output. The funnel is anchored by the SSM. This point represents the present environment from the contiguous parking lot (delineating things like the location of other cars which have a "time to contact" less than the time between new transformations or new funnel definitions in the locale space) to the state of the car instruments like the fuel gauge, headlight switch, etc. As a symbolic representation, we can think of the surround space as shown in Fig. 8(a).

The driver now selects a control which, in past daily commutes, has led to a surround model trajectory within the funnel. The control might be succintly summarized as "start car." This control transforms the surround space into an appropriate dimensionally smaller time-scaled element space. The element space can be thought of as consisting of all the variables necessary to describe the characteristics of the ignition switch with a detail commensurate with the time scale T_E. The control, element space, and present SEM define a funnel in the element space. The funnel *is* suitable for directly generating action. The funnel is anchored by the SEM. This point represents the present state and location of the ignition switch and can be represented symbolically by Fig. 8(b). The driver now initiates an action output: the ignition key is inserted into the switch and rotated. If the car is the driver's own, the action of inserting and turning the key is probably precognitive in nature, i.e., a direct transformation from surround to action output may be possible. If, on the other hand, an unfamiliar rental car is being driven, the action may well be compensatory in which variables in the element space represent the relative linear and angular orientation of the key and switch. This implies many more transformations of the element space into action outputs with associated SEM updates from the SIP than would be the case with the precognitive action. In terms of Fig. 6, this implies more momentary workload with the rental as opposed to the driver's own car.

The process of action decomposition and sensory composition just described has finally led to an action output. The process continues, with the driver and car starting along the selected route. As mentioned in Section II-D, of course, the process just described is not repeated for every action output, rather many element transformations and ensuing action outputs are instigated by a single task transformation, etc. As the reader is well aware, in any commute such as the one being described, lane changing or lane merging often occurs. In terms of the model of Fig. 1, the successful and safe completion of such maneuvers requires extensive use of surround models, particularly as regards the motion of other vehicles. This utilization is synonymous with an increase in D_S in (1) and constitutes a workload increment for the driver.

An interesting example of event-versus time-driven planning can be imagined in this gedanken experiment by the driver coming upon an unexpected traffic jam, say just after exit F in Fig. 7. It is assumed that this traffic holdup was not "predicted" by the funnel in the domain space, i.e., that the driver's first knowledge of its existence was in an SIP update of the locale model. Now it is almost a certainty that a theme exists in the scripter of every automobile commuter which can be summarized as "avoid traffic jam." Indeed, no American urban area is without traffic advisory reports broadcast over commercial radio stations, whose sole purpose is to warn commuters of such problems. In terms of the model discussed here, such reports are processed by the SIP as an SLM and used appropriately by the BG. Encountering such a problem in unexpected fashion here necessitates event-driven planning on the part of the driver, i.e., the script reviser is called into play to avoid theme conflict. In this case, the driver may select a revised script which can be summarized as "drive to destination J along alternate route beginning at interchange G." An interesting byproduct of this decision would be created if the alternate route is not as familiar to the driver as the primary route. In terms of the model of Fig. 1, the trajectory funnels generated in the locale space for the alternate route would be a good deal "wider" than those generated in the locale space for the primary route. This increased width represents the increased uncertainty the driver possesses regarding the future path of the trajectory in the locale space. This means that it will take less time for the actual trajectory to move into the part of the funnel where the BG takes action and defines a new trajectory funnel with help from the SIP. This, in turn means that, relative to the commute along the primary route, the commute along the alternate will involve increased processing demands D_L and, according to (1), increased workload for the driver. For this reason, it is not unreasonable to assume that another theme exists in the scripter which can be summarized as, "don't deviate from known route." This theme obviously conflicts with "avoid traffic jam" in this case, and the resolution of this conflict may be based upon hierarchies within the theme structures themselves.

The example just discussed may have belabored the obvious in some instances. However, it was deliberately chosen to pave the way for a discussion of a simulation experiment to be described in Section VII, dealing with human interaction with a much more complex dynamic system than an automobile, i.e., a helicopter with a sophisticated flight control and navigation system.

VII. An Experiment

A. Introduction

The experiment to be described involves a fixed-based manned simulation of a UH-1H helicopter in a single-pilot instrument landing approach. The simulation facility, including the model of the vehicle dynamics is discussed in [61]. In addition to the standard instruments such as air speed, altitude, instantaneous vertical speed, and attitude indicators (electromechanical in nature), the cockpit contained two CRT displays. The first was a stroke-written horizontal situation display (HSD) which presented detailed navigation information in a moving map type of format, shown in simplified form in Fig. 9. The second CRT, called the control display unit (CDU) was a multifunction device which allowed the pilot to update, monitor, or select navigational waypoints which define the linear course segments along which the vehicle flew. The general cockpit layout is shown in Fig. 10.

The scenario under study was an instrument radio navigation (RNAV) approach to Salinas Municipal Airport, in Salinas, CA. Navigation aids, such as VHF Omnirange with colocated distance measuring equipment (VOR/DME) was simulated. The simulated helicopter was equipped with the rotorcraft digital advanced avionics system (RODAAS) described in detail in [52]. Like the automated flight control system alluded to in Section IV, RODAAS offers considerable flexibility in terms of automation level. In the present experiment, three automation levels were exercised. 1) The first is "automatic" in which the autopilot was coupled to the RODAAS navigation system. Here the pilot's input to the system consisted in selecting air speed, altitude, and ground tracks (courses) for the helicopter to fly. 2) The second automation level is "flight director" in which the flight director giving three control commands (longitudinal and lateral cyclic and collective) to the pilot was coupled to the RODAAS navigation system. Here the pilot's input to the system consisted of those just outlined for the automatic system and the control stick inputs commanded by the flight director. 3) The third automation level is "manual" which is similar to the flight director mode except for the fact that the pilot had to integrate the pertinent displayed information (air speed, altitude, course deviation, attitude) for control stick inputs rather than relying on the flight director commands to provide these. In cases 2) and 3), no artificial stabilizaton was provided. These general automation schemes follow a pattern often used in human-machine studies involving aircraft flight control, e.g., [62], [63]. The Salinas airport was chosen since it provided a challenging and obviously realistic scenario identical to that used in simulation *and* flight test with a predecessor of the RODAAS system for fixed-wing aircraft [64].

Three pilots were used in the experiment. The first (pilot A) was a NASA test pilot who was very familiar with

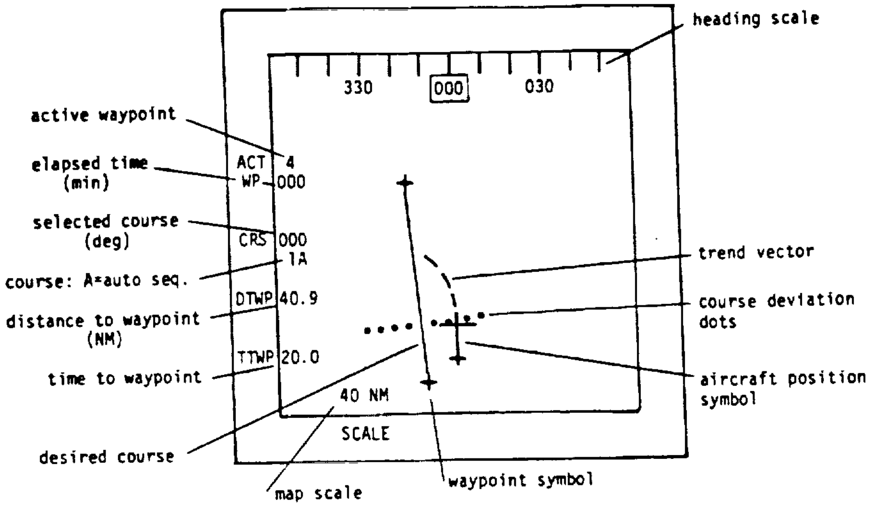


Fig. 9. RODAAS horizontal situation display.

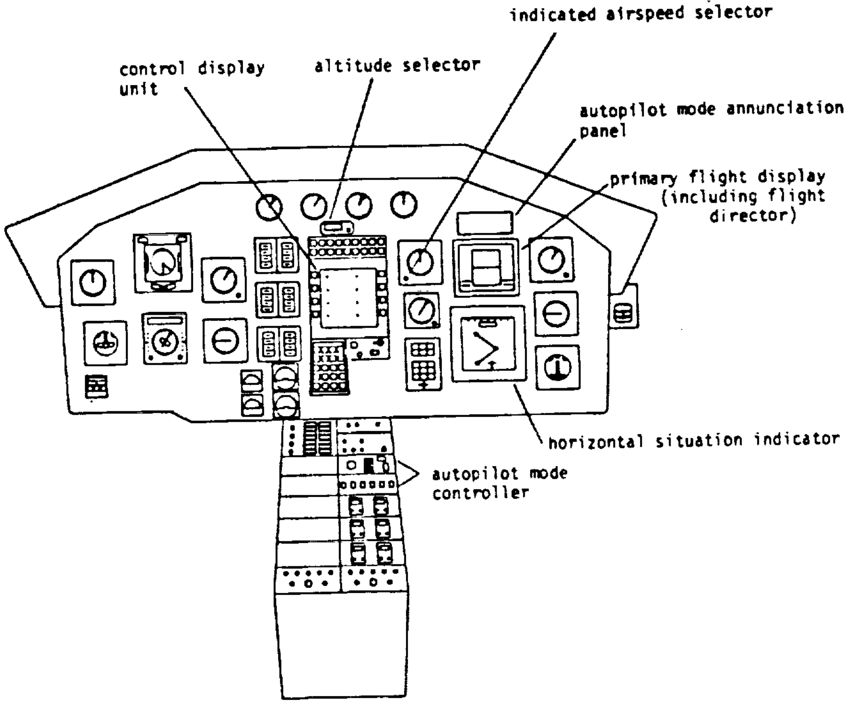

indicated airspeed selector
control display unit
altitude selector
autopilot mode annunciation panel
primary flight display (including flight director)
horizontal situation indicator
autopilot mode controller

Fig. 10. Simulator cockpit layout.

RODAAS. The second (pilot B) was equally familiar with RODAAS operation and was an instrument-rated, but fixed-wing, private pilot. The third (pilot C) was an instrument-rated fixed- and rotary-wing private pilot, who was initially unfamiliar with RODAAS. The varied amount of experience exhibited by the subjects both in terms of fixed- versus rotary-wing experience and in terms of familiarity with RODAAS operation was felt to be useful in the experimental design.

Fig. 11 is a simplified approach plate for the Salinas airport. The names "YAHOO," "JUNTA," etc., represent so-called "intersections" in the area and are used to locate points in the RNAV approach where minimum altitudes are changed. The shaded circles represent "waypoints" entered into the navigation computer of RODAAS prior to the experiment. The nominal scenario for this simulation went as follows. The flight was begun with the helicopter heading north at waypoint 3, stabilized at an altitude of 2000 ft and an air speed of 70 kn. The vehicle was then turned and flown toward waypoint 4, climbing to 5500 ft. At that point, the vehicle was turned (automatically, under direction of the flight director, or manually) toward YAHOO intersection, and a descent was begun, guided by the minimum altitudes denoted on the lower part of the approach plate. After descending and decelerating to an air speed commensurate with a landing, the subjects ini-

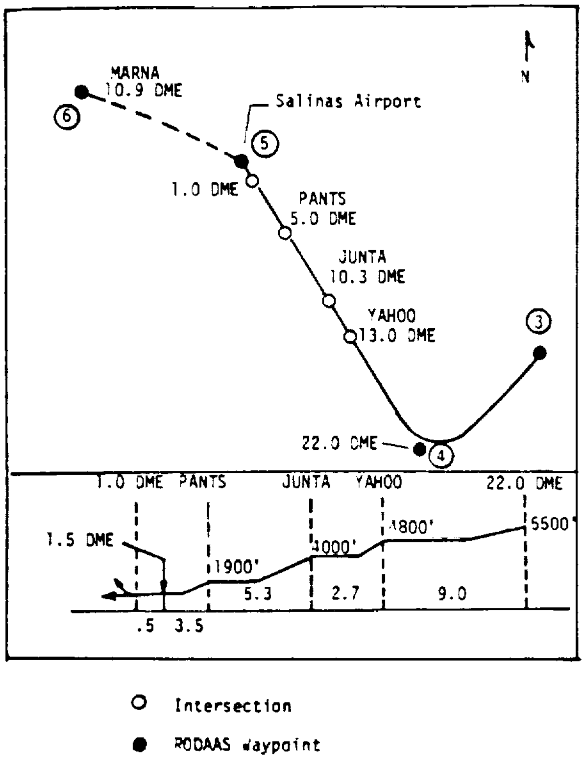


Fig. 11. Simplified approach plate for Salinas, CA, municipal airport.

tiated a "missed approach" procedure at 1.0 DME. The subjects were fully aware throughout the simulation that no actual landing would be made. This transformed what would normally be a mission involving event-driven planning into one involving time-driven planning only. For this airport, the missed approach procedure entailed a straight-ahead climb to 700 ft, then a climbing left turn to 2000 ft via a 275° radial from the airport VOR to MARNA intersection at 10.9 DME. Up to this point, all the necessary waypoints including proper inbound and outbound courses had been set up in the navigation computer before the experiment. In addition, a RODAAS capability called "automatic course sequencing" was used in all the experiments. As the helicopter flew over a waypoint, the outbound course was displayed. This was identical in direction to the inbound course to the next waypoint. The pilot did have to "activate" the next waypoint as s/he flew along, however, or the vehicle would fly along the old outbound course until the guidance signal faded. On the way to MARNA intersection the pilot had to perform in-flight planning to set up a course back to an intersection close to an alternate airfield, which was Moffett Field Naval Air Station.

The approximate duration of each simulated flight was 30 min. Since the subjects were free to select the air speed at which they wished to fly, the actual elapsed time varied somewhat for each simulation run. For example, lower air speeds were sometimes selected for the manual as compared to the automatic run. The only data recorded in the simulation were those obtained from verbal protocol. The subjects were instructed to "think aloud" throughout the simulation and their comments were recorded on tape. Of the three pilots selected, only pilot C needed significant training time on the simulator. Although pilot B was not a helicopter pilot he had extensive time on the simulator itself. Pilot C was allowed to train until she felt comfortable with the simulated vehicle and with those aspects of RODAAS operation pertinent to the landing approach scenario at hand.

B. Protocol Analysis

The data from the verbal protocol were analyzed in top-down fashion and interpreted in terms of the qualitative model discussed in previous sections. In transcribing the protocol, new paragraphs were begun whenever more than 3 s elapsed between the end of one comment and the beginning of another. The time at which the leading comment in each paragraph began was also recorded, measured from initiation of the simulation run. In the protocol excerpts which are to follow, ··· indicates that phrases or sentences have been deleted from the paragraph in question and --- indicates that intervening paragraphs have been deleted. Nine protocols were recorded and transcribed (one for each pilot and configuration). Based upon these protocols, the following conclusions can be drawn:

1) Internal models can be postulated which are quite similar in nature to those discussed in the automobile commute example. For example, Fig. 11 itself is a symbolic representation of the domain space, with the nominal aircraft flight path indicating, symbolically, the trajectory of the SDM. Locale spaces can be symbolically represented by areas around the waypoints/intersections in Fig. 11. In the instrument approach with no outside visual cues available, the union of the surround spaces can be represented symbolically by the cockpit itself, as shown in Fig. 10, with element spaces similarly represented by the various manipulators, switches, etc., pertinent to the operation of the vehicle and flight control and navigation systems.

2) The mission subphase in the simulation can be described simply as, "approach to land, then fly to missed approach intersection." A concise verbal description of the frame is, "instrument approach in instrument meteorological conditions." Next, a script can be described as, "RNAV approach to Salinas airport." The verbal protocols indicated that the tasks employed in this simulation could be summarized quite simply, i.e., "ascend/descend at constant vertical velocity to a desired altitude while maintaining air speed or while accelerating/decelerating to a desired air speed," and "turn at constant rate." The protocols clearly indicated that these were discrete *tasks* as the pilots would often indicate when they were initiating, say a deceleration or altitude change, and what the desired final air speed or altitude would be. For example, the following are excerpts from the protocol of pilot C in the manual configuration between JUNTA and PANTS intersections:

18:09 OK, I'm coming up on my altitude. I've still got 2.5 miles to go. That's good. Start to stabilize, here. I'm at 80 knots. And once I get stabilized on my altitude, I plan to start slowing down to 70.

18:54 OK, I'm going to go ahead and start trying to trim up for an air speed of about 70. Pretty well stabilized on my altitude, now.

19:40 OK, about 68 knots right now, and a little bit high.

19:53 OK, we're all stabilized on 5 DME, so I'm going to make the step down to 420 minimum descent altitude, and I'm going to try and hold 70 the whole way down.

Now while the tasks admit to simple verbal descriptions, it is the nature of the tasks as *transformations* (as in Fig. 2) which mark the real delineation of the manual, flight director, and automatic systems and which constitute the automation-induced complexity outlined in Section IV. The controls (again using the term as a transformation as in Fig. 2) can be exemplified verbally by phrases like "change the reading on the altitude select display," which would accompany the initiation of an altitude change in the automatic mode. Finally, action outputs are described by the movement of a manipulator such as the rotation of the dial which controls the rate at which the digital display of desired altitude changes on the altitude select display in the automatic mode.

3) The workloads associated with each of the three configurations were represented by a dichotomy separating the manual and the flight director/automatic systems. The subjects clearly indicated that the workload associated with the manual system bordered on the unacceptable, especially if air-to-ground communications, atmospheric turbulence, etc., were included in the simulation (which they were not). This result is certainly in keeping with the workload hypothesis offered in Section III, primarily in terms of marginal vehicle-handling qualities which determine the nature and frequency of element model utilization. For example, the following are excerpts from the protocol of pilot A in the manual configuration.

Between waypoint 4 and YAHOO intersection:

09:21 Very high workload, trying to keep this thing balanced on all three axes, lateral, speed, maintaining the vertical.

Just prior to initiating a go-around at the minimum decision altitude:

19:09 OK, my impression is that this would be a lot more difficult if we had turbulence in here.

Climbing out from the missed approach and heading for waypoint 6:

22:36 And because this thing is like balancing on top of a bowling ball, we'll get it all trimmed up before we try to do anything · · · .

Doing in-flight planning on the way to waypoint 6:

24:59 · · · OK, now this is, the workload is just getting ridiculous here trying to maintain some semblance of attitude and air speed control while punching buttons · · ·

The flight director system did reduce the subjects subjective impression of workload somewhat but not nearly as much a use of the automatic system. However, the pilots preferred the flight director over the manual system. Consider the following comments of pilot B, with the flight director.

Between waypoint 4 and YAHOO intersection:

12:03 OK, I'm doing a better job at tracking in all four axes right now, but it, I feel it's a pretty high workload in doing it. I don't have any more time available because the flight director is here, I'm just not going as far off course, (but) the flight director is taking up my time · · · .

14:25 It, the flight director, it's definitely hard to fly. It does help me in that I don't have to worry about my course. I know that by flying the flight director, it will keep me on course · · · .

In post-simulation comments pilot A remarked, "The flight director really helped. I think it particularly helped during the flight planning phase of it there · · · . But I felt more comfortable, because the flight director was giving me some indication how far things were off without having to scan the whole panel."

Now in terms of (1), the flight director as compared to the manual configuration seemed to result in a decrease in D_S, the processing demands of the surround model, but no change, or perhaps a slight increase in D_E, the processing demands of the element model. Decreases in D_S would arise from the decreased instrument scanning required to enable SSM updates. A slight increase in D_E is attributable to the fact that the flight director requires compensatory behavior on the part of the pilot [65], with a requirement for very frequent element transformations. Depending upon the amount of increase in D_E, the sum $D_E + D_S$ for the flight director as compared to the manual configuration would probably decrease, resulting in a decrease in workload for the subjects (see (1)).

4) Surprisingly few errors were committed in the simulated approaches for which verbal protocol data were taken. They will be examined here in order of importance to safe mission completion.

a) For a portion of a fight director approach, pilot B was using an altitude profile which was essentially one intersection behind his actual position. It is interesting to trace the development of this error through the protocol starting from the right turn at waypoint 4:

05:25 OK, we're getting closer, here. There, it commands the turn to the outbound course from waypoint 4. 4500 feet, 1000 feet to up to the altitude.

The problem begins here, with the pilot not having reached the 5500-ft altitude by waypoint 4. Rather than beginning to descend to 4800 ft which is the minimum at YAHOO, the pilot continues his climb to the 5500 ft which really is not pertinent after waypoint 4, although the action itself, is not "unsafe."

07:13 OK, still off course to the left. But the flight director is taking me back to it. We've leveled out pretty well at 5500 feet. Waypoint 5 is available. I'm going to make that active.

08:43 OK, a breather spell here to check the minimum descent altitude at 420 feet, and it's annunciated, which means it knows that we're annunciating it. Set at 1.5 nautical miles before the final approach fix. 16 DME, we've got a little ways to go yet before YAHOO at 13. 4800 feet at YAHOO, 4000 at JUNTA, the outer marker, down to 1900 at PANTS, and then down to the minimum descent altitude.

Here the pilot has correctly stated the desired minimum altitude at YAHOO as 4800 ft, but still has the vehicle stabilized at 5500 just 3 nmi from that intersection.

12:03 OK, I'm doing a better job at tracking in all four axes right now, but I feel it's a pretty high workload in doing it ···. OK, I'm at 4800 feet, haven't dropped, there's altitude hold.

12:34 OK, 10.3 is coming up. And at 10.3, we drop down to 4000 feet. OK, 10.3, flight path angle hold, and I missed that by a little bit. So again, I'm working pretty hard. OK, we're descending now, 4000 feet. Ooops! Holy Cow! I just, I screwed up! I'm one waypoint behind! [Note: The subject said "waypoint," but he meant "intersection."] I should have been decending down to 1900 feet. So, I erred on the safe side. But I'm one waypoint (intersection) behind. I should have been down to 4000 feet on my last waypoint [intersection] ··· .

Over 7 min elapsed from the error onset until the pilot noticed it. As he commented, it was an error on the safe side; however, it could have been serious had it gone much further and had he attempted to make the descent to the minimum decision altitude in too abrupt a fashion. In terms of the cardinal errors outlined in Section V, this error could be classified as error 1, incorrect definition of a trajectory funnel, in this case in the locale space. This error was particularly interesting in that it was not discovered in early updates of the SLM, i.e., the pilot was looking but not seeing.

b) For a brief portion of a manual approach, pilot A committed on error similar to the one just discussed. We pick up the protocol between waypoint 4 and YAHOO.

07:56 We need to get our waypoints changed. [To] tell me when it's time to come down. 19 miles, 6 miles to go before we start down.

08:12 Lost 7 knots air speed there, down to 93. Recovering back to 100.

08:27 (unintelligible) ft low on altitude.

08:37 18 DME.

09:21 Very high workload, trying to keep this thing balanced on all three axes, lateral, speed, maintaining the vertical.

09:42 OK, coming up on 16 miles, and we'll descend to 4000 feet.

09:52 OK, passing 16, altitude down to 4.

10:02 Got 2.7 miles to do it in, do it till 10.3.

10:24 Oh-oh! [I] screwed up! We're supposed to hold that till 13 DME! Supposed to be 4800 until 13 DME. Recovering back to 4800.

Note that in the transmission beginning at 7:56, the pilot has correctly stated that he is at 19 DME, and YAHOO is 6 mi away at 13 DME where a new descent is to be initiated. However, at 09:42, the YAHOO intersection has been incorrectly stated as being at 16 DME, and this initiates the error. Only 32 s elapse, however, before this error was recognized and corrected. As was the case with pilot B, this error can be classified as the first of the cardinal errors. Although somewhat speculative, the difference in the times between "commission" and "recognition" of errors for pilots A and B may well be attributed to the greater processing demand capacity D_C of the test pilot as opposed to the civilian fixed-wing pilot. Referring to (1) and (2), the larger hypothesized D_C would permit increases in D_E associated with monitoring (error detection) activity to occur under tolerable workload levels. Remember that pilot A's error was with the completely manual configuration, while pilot B's error was with the flight director.

c) At the very end of the run with the automatic configuration, pilot C was unable to perform a navigation operation with RODAAS called a "lateral direct to." This operation enables the pilot to define a direct course to a third waypoint while flying a course between two previously defined waypoints. In terms of the scenario of Fig. 11, this "lateral direct to" was to occur while flying between waypoints 5 and 6.

The pilot was to define a waypoint 7 at an intersection near Moffett field, then fly directly to that waypoint rather than flying to waypoint 6 first, then to 7. The pilot's difficulty centered upon her inability to erase an error message from the control display unit. Before any further navigation commands could be entered, this message had to be erased. The standard way of handling such problems was to depress a switch labeled "message acknowledged." Due to a design quirk in RODAAS, however, the particular error message at hand could not be eliminated in this fashion but required depressing a switch labeled "clear" instead. The pilot was unaware of the latter, and this seemingly innocuous bit of ignorance frustrated any attempt to fly to the desired waypoint. In terms of the error taxonomy of Section V, this error would fall into category 2, transformation of a model space using an incorrect or inappropriate frame, script, task, etc. In this case an incorrect control transformation was involved in mapping a surround space into an element space. This error instantiates a statement made in Section IV regarding the increase in training requirements which accompany the successful use of automated systems, i.e., "to have the ability to efficiently make the numerous transformations implied by $T(L) = S$ and $C(S) = E$."

5) Several examples of script/task reviewing were revealed in analyzing the protocols. For example, pilot B in the automatic configuration between PANTS and 1.0 DME in the approach:

> 18:41 And once again, I have lots of time to think about what's going to happen next, which is what I'm planning in my mind. I've got, I know the descent, the minimum decision altitude, of 420. I know that I'm going to climb straight ahead to 700 feet and then make a left climbing turn if I don't see the ground. I know that waypoint 6 is my missed approach waypoint, and that I can just fly straight to that, after punching the "go-around," I can fly "nav-couple" and it will fly me outbound from 5 straight to 6. I would then call up the controller and tell him that I was on a missed approach and that I'm requesting a flight back to my alternate. All this is stuff that I'm thinking about with the time I have available ··· previously I had no time to do that.

The "previously" the pilot was referring to was the approach with the manual configuration.

6) Finally, two themes could be discerned in the experiment at the level of the task selector. These could be summarized verbally as, "don't fly below the approach plate altitude minimums" and "maintain vehicle airspeed well above 50 knots for all but the terminal portion of the approach. Even then don't decelerate to less than 50." The first theme is based upon FAA regulations with obvious issues of flight safety involved. The second theme is essentially a workload/handling qualities trade-off. At higher air speeds, the unaugmented helicopter handling qualities improve (a decrease in D_E in (1) and (2)) because of the increased aerodynamic damping. However, at higher air speeds, things simply happen faster, particularly in the last part of the approach, (leading to an increase in D_S and D_L in (1) and (2)).

The experiment just discussed was chosen as a realistic example of human interaction with a complex dynamic system in which the controlled nature of the experiment allowed verbal protocol to be employed to shed light upon the activities of the humans involved. It was seen that the activities of the pilots were amenable to qualitative description quite similar to that used in describing human activity in the familiar but far less demanding automobile commute.

VIII. Concluding Remarks

The qualitative model of human interaction with complex dynamic systems described and exemplified in the previous sections represents a framework within which to study a variety of issues concerned with human–machine systems. Among these are workload, handling qualities, automation effects, human error, and finally, more quantitative representations of human activity, itself. The model's *raison d'etre* is to provide a qualitative explanation of human anticipatory, as opposed to reactive, behavior in interacting with complex dynamic systems. This former viewpoint would appear to be essential in developing veridical quantitative representations of the human's higher supervisory activities. Finally, the framing, scripting, tasking, controlling, and action outputting which have been hypothesized to be fundamental modes of describing human behavior also suggest a means for describing the function of automated systems. This description would be compatible with the concerns of the human who is operating or supervising the system at hand.

IX. Nomenclature

$A(E)$	Transformation of the element space to action output.
BG	Behavior generator.
$C(S)$	Transformation of the surround space by a control.
D	Domain space.
D_C	Processing demand capacity of human.
D_D	Processing demands associated with domain model.
D_E	Processing demands associated with element model.
DME	Distance measuring equipment.
E	Element space.
EXP	Expectation.
$F(W)$	Transformation of the world space by a frame.
IM	Internal model.
L	Locale space.
RODAAS	Rotorcraft digital advanced avionics system.
S	Surround space.
$\mathrm{Sc}(D)$	Transformation of the domain space by a script.
SDM	State-of-the-domain model.
SEM	State-of-the-element model.
SIP	Sensory information processor.
SLM	State-of-the-locale model.
SSM	State-of-the-surround model.
SWM	State-of-the-world model.
T_D	Time scale associated with the domain space.
T_E	Time scale associated with the element space.
T_L	Time scale associated with the locale space.

$T(L)$	Transformation of the locale space by a task.
T_S	Time scale associated with the surround space.
T_W	Time scale associated with the world space.
WM	World model.

References

[1] T. B. Sheridan and G. Johannsen, Eds., *Monitoring Behavior and Supervisory Control.* New York: Plenum, 1976.

[2] R. W. Pew and S. Baron, "Perspectives on human performance modelling," *Automatica*, vol. 19, pp. 663–676, 1983.

[3] T. B. Sheridan and R. T. Hennessy, Eds., *Research on Modeling of Supervisory Control Behavior—Report of a Workshop.* Washington, DC: National Academy Press, 1984.

[4] J. Rasmussen, "Skills, rules, and knowledge: Signals, signs, and symbols, and other distinctions in human performance models," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-13, pp. 257–266, May/June 1983.

[5] W. B. Rouse, "Human–computer interaction in the control of dynamic systems," *Computing Surveys*, vol. 13, pp. 71–99, Mar. 1981.

[6] R. Rosen, *Anticipatory Systems, Philosophical, Mathematical and Methodological Foundations.* New York: Pergamon, 1985.

[7] J. Rasmussen, "On the structure of knowledge—A morphology of mental models in a man–machine system context," Riso Nat. Lab., Roskilde, Denmark, Riso-M-2192, 1979.

[8] C. Kelley, *Manual and Automatic Control.* New York: Wiley, 1968.

[9] R. A. Hess, "Feedback control models," in *Handbook of Human Factors/Ergonomics*, G. Salvendy, Ed. New York: Wiley, in press.

[10] K. J. Stein, "DARPA stressing development of pilot's associate system," *Aviation Week Space Technol.*, p. 69, Apr. 22, 1985.

[11] K. Craik, *The Nature of Explanation.* Cambridge: Cambridge Univ. Press, 1943.

[12] R. D. Smallwood, "Internal models and the human instrument monitor," *IEEE Trans. Human Factors Electron.*, vol. HFE-8, pp. 181–187, 1967.

[13] W. Veldhuyzen and H. G. Stassen, "The internal model concept: An application to modeling human control of large ships," *Human Factors*, vol. 19, pp. 367–380, 1977.

[14] R. J. Jagcinski and R. A. Miller, "Describing the human operator's model of a dynamic system," *Human Factors*, vol. 20, pp. 425–434, 1978.

[15] P. N. Johnson-Laird, "Mental models in cognitive science," *Cognitive Sci.*, no. 4, pp. 71–115, 1980.

[16] J. S. Albus, *Brains Behavior and Robotics.* Peterborough, NH: BYTE Books, 1981.

[17] D. Gentner and A. L. Stevens, Ed., *Mental Models.* Hillsdale, NJ: Erlbaum, 1983.

[18] W. B. Rouse and N. M. Morris, "On looking into the black box, prospects and limits in the search for mental models," Georgia Inst. Technol., Center for Man–Machine Syst. Res., Rep. 85-2, May 1985.

[19] M. G. Singh, *Dynamical Hierarchical Control.* Amsterdam, The Netherlands: North Holland, 1977.

[20] J. Rasmussen, "Coping with complexity," in *Proc. 1st European Annu. Conf. Human Decision Making and Manual Control*, 1981, pp. 69–91.

[21] T. B. Sheridan, "The human operator in control instrumentation," in *Progress in Control Engineering*, R. H. Macmillan *et al.*, Eds. New York: Academic, 1962, pp. 141–187.

[22] B. Hayes-Roth and F. Hayes-Roth, "A cognitive model of planning," *Cognitive Sci.*, vol. 3, pp. 275–310, 1979.

[23] R. Wilensky, "Meta-planning: Representing and using knowledge about planning in problem solving and natural language understanding," *Cognitive Sci.*, vol. 5, pp. 197–233, 1981.

[24] W. B. Rouse, "Models of human problem solving, detection, diagnosis, and compensation for system failures," in *Proc. IFAC Conf. Analysis, Design and Evaluation of Man–Machine Systems*, 1982, pp. 167–184.

[25] G. Johannsen and W. B. Rouse, "Mathematical concepts for modeling human behavior in complex man–machine systems," *Human Factors*, vol. 21, pp. 733–747, 1979.

[26] ——, "Studies of planning behavior of aircraft pilots in normal, abnormal, and emergency situations," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-13, pp. 267–278, May/June 1983.

[27] M. Minsky, "A framework for representing knowledge," in *The Psychology of Computer Vision*, P. H. Winston, Ed. New York: McGraw-Hill, 1975.

[28] R. C. Schank and R. P. Abelson, *Scripts, Plans, Goals, and Understanding.* Hillsdale, NJ: Erlbaum, 1977.

[29] E. S. Krendel and D. T. McRuer, "A servomechanism approach to skill development," *J. Franklin Inst.*, vol. 269, pp. 24–42, Jan. 1960.

[30] D. T. McRuer, W. F. Clement, and R. W. Allen, "A theory of human error," in *Proc. Annu. Conf. Manual Control*, 1981, pp. 583–594.

[31] T. B. Sheridan, "Toward a general model of supervisory control," in *Monitoring Behavior and Supervisory Control*, T. B. Sheridan and G. Johannsen, Eds. New York: Plenum, 1976, pp. 271–281.

[32] T. B. Sheridan and W. R. Ferrell, *Man–Machine Systems: Information, Control and Decision Models of Human Performance.* Cambridge, MA: MIT Press, 1974.

[33] G. G. Hendrix, "Modeling actions and continuous processes," *Artificial Intell.*, vol. 4, pp. 145–180, 1973.

[34] L. A. Zadeh, K. S. Fu, K. Tanaka, and M. Shimura, Eds., *Fuzzy Sets and Their Applications to Cognitive and Decision Processes.* New York: Academic, 1975.

[35] R. M. Hunt and W. B. Rouse, "A fuzzy rule-based model of human problem solving," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-14, pp. 112–120, Jan./Feb. 1984.

[36] U. Kramer and G. Rohr, "A fuzzy model for driver behavior: Computer simulation and experimental results," in *Proc. IFAC Conf. Analysis, Design and Evaluation of Man–Machine Systems*, 1982, pp. 31–35.

[37] S. K. Card, T. P. Morgan, and A. Newell, *The Psychology of Human-Computer Interaction.* Hillsdale, NJ: Erlbaum, 1983.

[38] R. A. Hess, "Pursuit tracking and higher levels of skill development in the human pilot," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-11, pp. 262–273, Apr. 1981.

[39] P. H. Wewerinke, "A model of the human observer and decision maker," in *Proc. 17th Annu. Conf. Manual Control*, 1981, pp. 557–570.

[40] K. R. Pattipati, D. L. Kleinman, and A. R. Ephrath, "A dynamic decision model of human task selection performance," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-13, pp. 145–166, Mar./Apr. 1983.

[41] S. Baron, R. Muralidharan, R. Lancraft, and G. Zacharias, "PROCRU: A model for analyzing crew procedures in approach to landing," Bolt Beranek and Newman, Inc., Rep. 4374, 1980.

[42] R. J. Wherry, "The human operator simulator—HOS," in *Monitoring Behavior and Supervisory Control*, T. B. Sheridan and G. Johannsen, Eds. New York: Plenum, 1976, pp. 283–293.

[43] B. Doring and A. Knauper, "A simulation study with a combined network and production systems model of pilot behavior on an ILS-approach," *Automatica*, vol. 19, pp. 741–747, 1983.

[44] A. Newell and H. A. Simon, *Human Problem Solving.* Englewood Cliffs, NJ: Prentice-Hall, 1972.

[45] Y. Anzai, "Cognitive control of real-time event-driven systems," *Cognitive Sci.*, vol. 8, pp. 221–254, 1984.

[46] D. McDermott, "A temporal logic for reasoning about processes and plans," *Cognitive Sci.*, vol. 6, pp. 101–155, 1982.

[47] N. Moray, Editor, *Mental Workload: Theory and Measurement.* New York: Plenum, 1979.

[48] J. W. Senders, "Axiomatic models of workload," in *Mental Worklaod: Theory and Measurement*, N. Moray, Ed. New York: Plenum, pp. 263–267, 1979.

[49] E. L. Weiner and R. E. Curry, "Flight-deck automation: Promises and problems," *Ergonomics*, vol. 23, pp. 995–1011, 1980.

[50] E. R. F. W. Crossman, "Automation and skill," in *The Human Operator in Process Control*, E. Edwards and F. P. Lees, Eds. London: Taylor and Francis, 1974, pp. 1–24.

[51] E. Edwards, "Automation in civil transport aircraft," *Appl. Ergonomics*, vol. 8., p. 98, 1977.

[52] E. M. Peterson, J. Bailey, and T. J. McManus, "Rotorcraft digital advanced avionics system (RODAAS) functional description," NASA CR-166611, 1983.

[53] R. E. Curry, "What pilots like (and don't like) about the new cockpit technology," in *Proc. 20th Annu. Conf. Manual Control*, 1984, pp. 199–215.

[54] W. T. Singleton, "Theoretical approaches to human error," *Ergonomics*, vol. 16, pp. 727–737, 1973.

[55] D. A. Norman, "Categorization of action slips," *Psychol. Rev.*, vol. 88, pp. 1-15, Jan. 1981.

[56] W. B. Rouse and S. H. Rouse, "Analysis and classification of human error," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-13, pp. 539-549, July/Aug. 1983.

[57] W. B. Rouse, "Optical allocation of system development resources to reduce and/or tolerate human error," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-15, pp. 620-30, Sept./Oct. 1985.

[58] D. Meister and G. F. Rabideau, *Human Factors Evaluation in System Development.* New York: Wiley, 1965.

[59] W. T. Singleton, "Techniques for determining the causes of error," *Appl. Ergonomics*, vol. 3, pp. 126-131, 1972.

[60] S. G. Hart and M. R. Bortolussi, "Pilot errors as a source of workload," *Human Factors*, vol. 26, pp. 545-556, Oct. 1984.

[61] W. F. Nedell, G. H. Hardy, T. M. Lichtenstein, and G. Leong, "Development, and validation of the crew system integration research station," NASA Ames Res. Center, NASA TM 86841, 1985.

[62] D. R. Yoerger, "Man-machine performance for a simulated aircraft with multilevel automatic control system," M.S. thesis, Dep. Mech. Eng., Mass. Inst. Technol., Cambridge, May 1979.

[63] R. A. Hess and B. D. McNally, "Automation effects in manual control systems," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-16, pp. 111-121, Jan./Feb. 1986.

[64] G. P. Callas, D. G. Denery, G. H. Hardy, and B. F. Nedell, "Flight evaluation results from the general-aviation advanced avionics system program," NASA Ames Res. Center, NASA TM 84397, 1983.

[65] R. A. Hess, "Aircraft control-display analysis and design using the optimal control model of the human pilot," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-11, pp. 465-480, July 1981.

Ronald A. Hess was born in Norwalk, OH, on March 12, 1942. He received the B.S., M.S., and Ph.D. degrees in aerospace engineering from the University of Cincinnati, Cincinnati, OH, in 1965, 1967, and 1970, respectively.

After completing his doctoral work he joined the faculty of the Department of Aeronautics at the Naval Postgraduate School in Monterey, CA. In 1976, he joined the staff in the Flight Systems Research Division at NASA Ames Research Center. At NASA he conducted research in the areas of aircraft handling qualities, control/display analysis and design, and manual control theory. He currently is a Professor in the Department of Mechanical Engineering at the University of California, Davis. His current research interests lie in the areas of automatic and manual control of aircraft.

Dr. Hess is a member of AIAA and Sigma Xi, and is an Associate Editor of this TRANSACTIONS and the *Journal of Aircraft*.

corrected copy

PRECEDING PAGE BLANK NOT FILMED

GENERALIZED PREDICTIVE CONTROL OF DYNAMIC SYSTEMS

R. A. Hess and Y. C. Jung
Department of Mechanical Engineering
University of California
Davis, California 95616

Abstract

Generalized Predictive Control (GPC) describes an algorithm for the stable, adaptive control of dynamic systems. In the algorithm, a control input is generated which minimizes a quadratic cost function consisting of a weighted sum of errors between desired and predicted future system outputs and future predicted control increments. The predictions are obtained from an internal model of the plant dynamics. The GPC approach is similar in concept to preview control, which has been discussed in the manual control literature. The GPC algorithm is applied to a simplified rotorcraft terrain following/terrain avoidance problem and its performance is compared to that of a conventional compensatory automatic system in terms of flight path performance, control activity and control law implementation. The potential of the GPC algorithm to serve as a paradigm for the human operator is briefly discussed.

Introduction

Background

In many manual control tasks, the ability of the human operator to "look ahead" or "preview" is a vital strategy in achieving acceptable man/machine performance. Models of human preview control have often employed an "internal model" of the plant dynamics with which the human is presumed to generate predictions of future plant output given current plant state and present and future control inputs e.g., [1,2]. Over the past decade, a technique for the design of automatic controllers, called variously, Model Predictive Heuristic Control, Model Algorithmic Control, or Output Predictive Control, has been introduced which approximates the activity of the human preview controller [3-5].

More recently, Clarke and Zhang [6], and Clarke, et al., [7] have introduced Generalized Predictive Control (GPC) and related it to the earlier approaches of Refs. 3-5 and state-space Linear Quadratic (LQ) designs. It is the GPC approach which is the subject of the research reported herein. Details of the GPC algorithm can be found in Ref. 7, however a brief review of the salient features of the approach will be undertaken in the following sections.

The GPC Algorithm

The plant is modeled in discrete fashion using the so-called Controlled Auto-Regressive Integrated Moving Average model [7]:

$$A(q^{-1})y(k) = B(q^{-1})u(k-1) + \xi(k)/\Delta \qquad k = 0, 1, 2, \text{etc.} \tag{1}$$

where $A(q^{-1})$ and $B(q^{-1})$ are polynomials in the delay operator q^{-1}, $y(k)$ and $u(k)$ are output and control variables, respectively, and $\xi(k)$ is an uncorrelated random sequence. Δ represents the differencing operator $1 - q^{-1}$. The actual sampling interval is T, so that, at each sampling instant, the independent variable is kT. Now a prediction of the plant output, given measured output up to time kT and control input $u(k + i)$ for $i \geq -1$, is

[illegible] these parameters is

$$\hat{y}(k + j|k) = G_j\Delta u(k + j - 1) + F_jy(k) \tag{2}$$

where

j = the number of time steps ahead being predicted

$G_j(q^{-1})$ = E_jB and where E_j results from a recursive solution of the Diophantine relation

$$1 = E_j(q^{-1})A\Delta + q^{-j}F_j(q^{-1}) \tag{3}$$

Here, E_j and F_j are polynomials uniquely defined, given $A(q^{-1})$ and the integer j.

Now a predictive control law can be defined as that which minimizes the cost function given by

$$J(N_1,N_2) = E\left[\sum_{j=N_1}^{N_2}[\hat{y}(k + j) - w(k + j)]^2 + \sum_{j=1}^{N_2}\lambda(j)[\Delta u(k + j - 1]^2\right] \tag{4}$$

where

N_1 = the minimum costing horizon

N_2 = the maximum costing horizon

$w(k)$ = the desired value of the output y at the k^{th} sampling instant

$\lambda(j)$ = a control weighting sequence

Equation (4) is concerned only with a subset of future time defined N_2T secs into the future and is dependent upon data up to time kT. Note how the control is implemented: The optimal control at the first sampling instant is applied and the minimization of J is repeated at the next sample. Also note that the cost on the control is over all future control inputs which effect the outputs included in J. This control law can be classified as Open-Loop-Feedback-Optimal with an autoregressive disturbance process [7]. The authors feel that this control philosophy is similar to that which the human operator employs when controlling plants for which desired future output can be defined. Examples are automobile driving or aircraft flight path control in near-earth flight.

Significant reductions in the order of the matrices involved in computing the optimal control can be made by requiring that, after an interval NU < N_2, projected control increments are assumed to be zero, i.e.,

$$\Delta u(k + j - 1) = 0 \qquad j > NU \tag{5}$$

where NU is called the "control horizon". This is equivalent to placing infinite weights on control changes after some future time. In addition to computational simplifications, introduction of the control horizon also allows the stable control of non-minimum phase plants [7].

With the introduction of the control horizon, the prediction equations become

$$\underline{\hat{y}} = \underline{G}_1\underline{\tilde{u}} + \underline{f} \tag{6}$$

where

$$\underline{\hat{y}} = [\hat{y}(k + 1),\hat{y}(k + 2),....\hat{y}(k + N)]^T$$

$$\underline{\tilde{u}} = [\Delta u(k),\Delta u(k + 1),....\Delta u(k + N - 1)]^T$$

$$\underline{f} = [f(k + 1),f(k + 2),....f(k + N)]^T$$

$$N = \text{output horizon} = N_2 \text{ here.}$$

$$\underline{G}_1 = \begin{bmatrix} g_0 & 0 & \cdots & 0 \\ g_1 & g_0 & & \\ \cdot & \cdot\cdot & \cdots & 0 \\ \vdots & \cdot & & g_0 \\ \vdots & \cdot & & \vdots \\ g_{N-1} & g_{N-2} & \cdots & g_{N-NU} \end{bmatrix} \tag{7}$$

with f(k + j) being that component of y(k + j) composed of signals which are known at time kT [7], and the g_i are elements of the polynomial $G_i(q^{-1})$, itself obtained from the recursive Diophantine relation (3). The corresponding control law is given by

$$\underline{\tilde{u}} = (\underline{G}_1^T\underline{G}_1 + \lambda\underline{I})^{-1}\underline{G}_1^T(\underline{w} - \underline{f})$$

where (8)

$$\underline{w} = [w(k + 1),w(k + 2),....w(k + N)]^T$$

The matrix involved in the inversion above is of dimension NU x NU. Equation (8) and the oertinent relations preceding it define the GPC algorithm. Although not considered here, the GPC algorithm can be made adaptive by the inclusion of a "standard" recursive least-square parameter estimator [8]. Some theoretical stability results are presented in Ref. 7 by relating GPC to state-space LQ control laws. The reader is referred to this reference for details.

A number of parameters are obviously available as design variables in applications of the GPC algorithm. They are: The minimum and maximum costing horizons, N_1 and N_2, the control horizon NU, and the control weighting sequence λ(k). The role played by these parameters is best demonstrated by means of the flight control example of the following section.

The on-line computational requirements of the GPC algorithm for cases in which no adaptation is occurring are very minimal since all major computations including the matrix inversion of Eq. (8) can be performed off-line. Thus, on-line computations are limited to the matrix multiplications shown in Eq. (8), with $N = N_2$.

Applications to Flight Control

Introduction

Terrain-following/terrain-avoidance (TF/TA) flight offers a significant challenge to the designers of automatic flight control systems. The response requirements of these systems imply relatively high bandwidth outer loop command following characteristics which are difficult to obtain using classical design techniques. The ability of the human pilot to succesfully complete such tasks has led to the investigation of pertinent preview control models for near-earth flight [9]. The similarity between the philosophy of these models and that of the GPC approach led to a consideration of the latter algorithm as a candidate for automatic flight path control in the TF/TA task. Indeed, Reid, et al., [5], have applied an Output Predictive algorithm to a terrain following flight control task. Conceptually at least, this algorithm is a special case of GPC as it considers the control input to be held constant over some number of sampling intervals, then provides a least-square control solution which minimizes a cost function similar to Eq. (4), but with no weighting on control inputs, a minimum output horizon of zero, and a control horizon matching the maximum output horizon. The necessity of holding the control input constant over a number of sampling intervals arose in ensuring output stability.

In conducting some preliminary evaluations of the Output Predictive algorithm for the height control task to be considered here, performance was, in general, unsatisfactory. The necessity of holding the control input constant for multiples of the sampling interval coupled with the lack of control weighting in the cost function led to unrealistic control inputs, i.e., control signals which resembled relay-like functions alternating between large positive and negative amplitudes in all applications. For this reason, the Output Predictive algorithm was eschewed in favor of the GPC system to be described.

Simplified Rotorcraft Vertical Dynamics

Figures 1-3 show the three "plants" which were utilized in this study. They all involve a simplified rotorcraft "bare-airframe" vertical velocity to collective input transfer function given by

$$\frac{\dot{h}}{\delta}(s) = \frac{-(s - 20)}{(s + 1)(s + 20)} \doteq \frac{e^{-0.1s}}{(s + 1)} \tag{9}$$

The introduction of the first-order Pade' approximation to the time delay offers an interesting challenge to the control algorithm since it involves non-minimum phase dynamics. Figure 1 represents a "bare-airframe" in which the control input for the GPC algorithm will be collective control. Figure 2 represents the bare-airframe with a vertical velocity control loop closed about it. Here, the control input for the GPC algorithm will be commanded vertical velocity $\dot{h}_c$. The effective plant for this case will be

$$\frac{h}{\dot{h}_c}(s) = \frac{-4(s + 0.5)(s - 20)}{s(s^3 + 17s^2 + 98s + 40)} \tag{10}$$

$$= \frac{-4(s + 0.5)(s - 20)}{s(s + 0.44)[s^2 + 2(0.87)(9.52)s + 9.52^2]}$$

Finally, Fig. 3 represents the bare-airframe with velocity and height control loops closed. Here the control input for the GPC algorithm will be commanded height. The effective plant for this case will be

$$\frac{h}{h_c}(s) = \frac{-4(s + 0.5)(s - 20)}{s^4 + 17s^3 + 94s^2 + 118s + 40} \qquad (11)$$

$$= \frac{-4(s + 0.5)(s - 20)}{(s + 1)(s + 0.579)[s^2 + 2(0.93)(8.31)s + 8.31^2]}$$

The rationale for selecting the dynamic systems of Figs. 1-3 was that they represented the range of possible levels of GPC utilization in a typical flight control application from inner-loop control actuator commands in Fig. 1 to outer-loop height guidance commands in Fig. 3.

Terrain Following/Terrain Avoidance

The commanded vertical flight path for this application was actually a time history similar to that utilized in Ref. 5, represented as a sum of sinusoids

$$h_c = 20[\sin(.05(2\pi t)) + \sin(.06(2\pi t)) + \sin(.08(2\pi t))] \text{ ft} \qquad (12)$$

Equation (12) can be thought of a representing a commanded flight path which would be provided by an on-board computer in a TF/TA task. In implementing the GPC algorithm, the "desired" output or vehicle path was an exponential curve which continuously defined a smooth "capture" trajectory from the vehicle's present position to the command of Eq. (12). This capture trajectory was given by

$$h_c'(j + k) = h_c(k + j) - \exp(-\tau_e j)[h_c(k + j) - h(k)] \qquad (13)$$

$$j = 1, 2, \ldots, N_2$$

Although the time constant τ_e could serve as another design variable in the GPC algorithm, it was maintained at 0.5 secs for this study. Thus, the time to 50% and 95% amplitudes for the trajectory of Eq. (13) was 0.14 and 0.6 secs, respectively. These values were deemed acceptable for this vehicle and task.

Figure 4 shows the performance of the system of Fig. 3 without GPC and with the command trajectory of Eq. (12) serving as the system input. This serves as a benchmark system for GPC performance comparisons as it prepresents the performance of a "classical" multi-loop control design with fairly high loop bandwidths. Note the height errors exceed 20 ft in some instances. This classical design has been discretized with a 0.1 sec sampling interval so that it is comparable to the GPC implementation.

Figures 5-7 show the performance of the GPC systems. The command and actual vehicle trajectories (dashed and solid lines, respectively) are indistinguishable in these figures because of the excellent tracking performance. This performance is indicated by the small height errors, where, with the exception of the initial and final transients, they are less tha 1 ft in magnitude. The transients are due to the abrupt initiation and termination of the height command at the zero crossings of the sum of sinusoids at the begininning and end of the simulation. The figure parts labeled "GPC Input" represent the "control" as provided by the GPC algorithm (u in Eq. (1)), and this input varies from the systems of Figs. 1 through 3. The excellent performance of the GPC algorithm is evident in all the systems with performance deteriorating slightly as one moves from the system of Fig. 1 to that of Fig. 3. The GPC parameters for all the applications were

N_1	=	minimum output horizon	=	1 (0.1 secs)
N_2	=	maximum output horizon	=	50 (5 secs)
NU	=	control horizon	=	20 (2 secs)
λ	=	control weighting sequence	=	0.2

These values were obtained by a trial and error procedure.

It was of interest to investigate the robustness of the GPC algorithm as regards the quality of the "internal model" which was used in the Diophantine relation (3). To this end, a brief investigation was conducted on the system of Fig. 1 in which the dynamics of Eqs. (9) were not changed in the digital simulations, but the "internal model" of these dynamics in the GPC algorithm were given by

$$\frac{\dot{h}}{\delta}(s) = \frac{1}{(s + 1)} \qquad (14)$$

i.e., the Pade' approximated time delay was omitted. For all practical purposes, the results were identical to those of Fig. 5 with no change in the GPC design parameters necessary. This is an encouraging result, as it indicates that inaccurate modeling of system delays or higher frequency system dynamics, will not have a detrimental effect upon GPC performance.

Manual Control Applications

Although not pursued in this study, the application of the GPC algorithm to the description of manual control tasks in which desired future output can be defined apprears promising. Tasks which immediately come to mind are automobile driving and aircraft near-earth flight. The inclusion of weightings on control rate in the cost function of Eq. (4) as is typically done in the Optimal Control Model of the human operator [10], can be accomplished by suitable modification of the GPC algorithm [7]. The basic format of the GPC approach, with its output and control horizons, its internal model, and its output (as opposed to state) feedback structure make it a worthy candidate for future research in the manual control area.

Conclusions

Based upon the analyses performed to date, the following conclusions can be drawn:

1.) The GPC algorithm offers tracking performance superior to classical multi-loop control system designs. In the simplified TF/TA task studied here, an order of magnitude reduction in absolute height errors was achieved.

2.) The GPC algorithm can be introduced with equal ease and success at a number of different points in a control hierarchy. In the examples studied here, GPC produced optimal control policies where "control" was defined from inner-loop actuator commands to outer-loop guidance commands.

3.) The on-line computational requirements for the non adaptive GPC applications are minimal.

4.) A limited examination of the effects of inaccuracies in the GPC internal model upon GPC performance indicates that errors in the estimation of plant time delay or higher frequency dynamics have minimal effect upon performance.

Acknowledgment

This work was supported by a grant from the Aircraft Guidance and Navigation Branch of NASA Ames Research Center.

References

[1] Sheridan, T. B., "Three Models of Preview Control," IEEE Transactions on Human Factors in Electronics, Vol. HFE-7, No. 2, June 1966, pp. 91-102.

[2] Kelley, C. E., Manual and Automatic Control, Chap. 15, Wiley, New York, 1968.

[3] Richalet, J., Rault, A., Testud, J. L., Papon, J., "Model Predictive Heuristic Control: Applications to Industrial Processes," Automatica, Vol. 14, No. 5, Sept 1978, pp. 413-428.

[4] Rouhani, R., and Mehra, R. K., "Model Algorithmic Control (MAC): Basic Theoretical Properties," Automatica, Vol. 18, No. 4, July 1982, pp. 401-414.

[5] Reid, J. G., Chaffin, D. E., and Silverthorn, J. T., "Output Predictive Algorithmic Control: Precision Tracking with Applications to Terrain Following," Journal of Guidance and Control, Vol. 4, No. 5, Sept.-Oct. 1981, pp. 502-509.

[6] Clarke, D. W., and Zhang, L., "Long-Range Predictive Control Using Weighting-Sequence Models," IEE Proceedings, Vol. 134, Pt. D., No. 3, May 1987, pp. 187-195.

[7] Clarke, D. W., Mohtadi, C., and Tuffs, P. S., "Generalized Predictive Control, Parts I and II," Automatica, Vol. 23, No. 2, March 1987, pp. 137-160.

[8] Gelb, A., Applied Optimal Estimation, MIT Press, Cambridge, 1974.

[9] Hess, R. A., and Chan, K. K., "A Preview Control Pilot Model for Near-Earth Maneuvering Helicopter Flight," to appear in Journal of Guidance, Control and Dynamics, Vol. 11, No. 1, Jan.-Feb. 1988.

[10] Hess, R. A., "Feedback Control Models," in Handbook of Human Factors, Wiley, New York, G. Salvendy, Ed., 1987, Chap. 9.5.

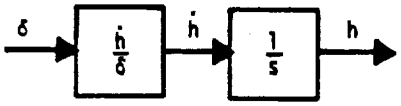
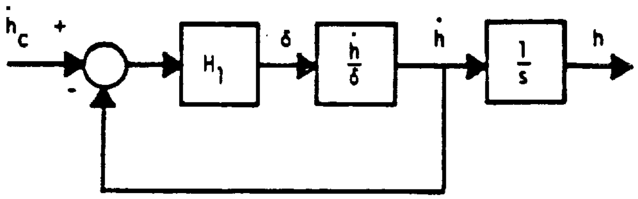
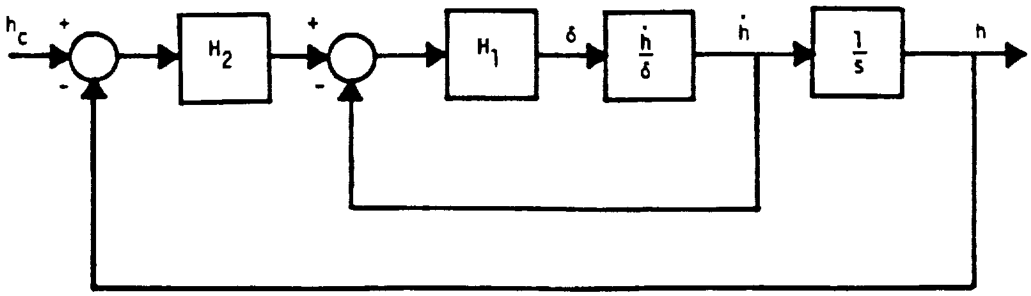


Fig. 1 Bare-airframe



$$H_1 = \frac{4(s + 0.5)}{s}$$

Fig. 2 Bare-airframe with vertical velocity control loop



$$H_1 = \frac{4(s + 0.5)}{s} \qquad H_2 = 1.0$$

Fig. 3 Bare-airframe with vertical velocity and height control loops

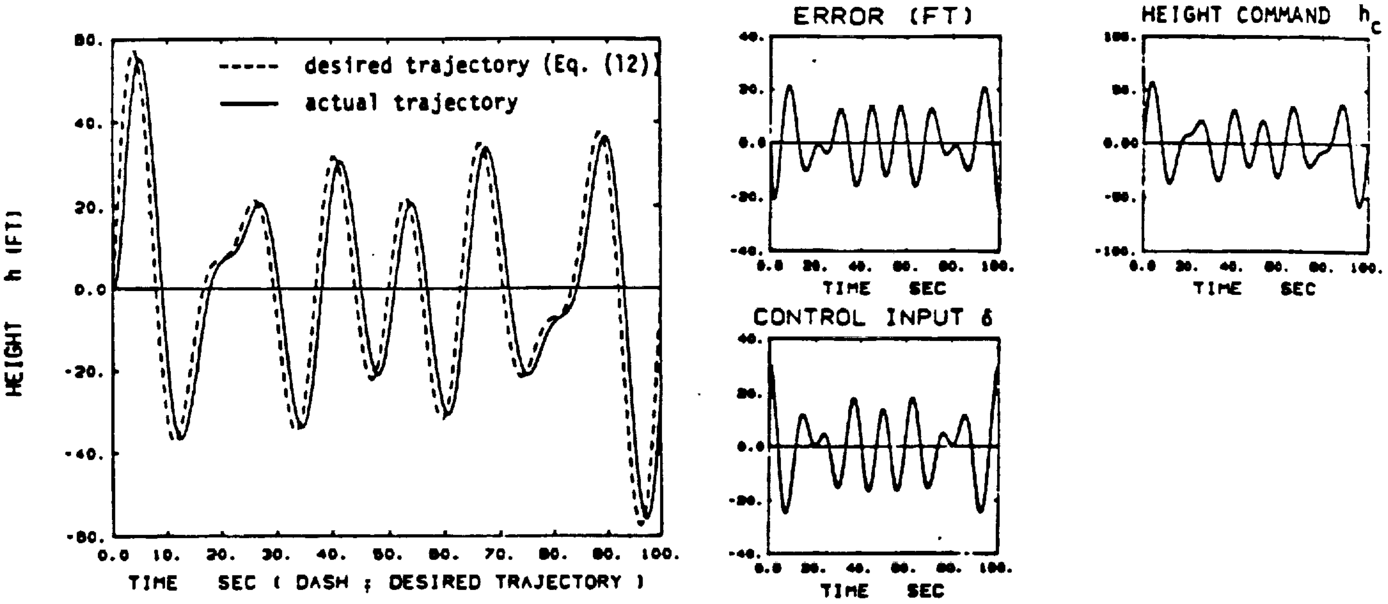


Fig. 4 Terrain following performance of system of Fig. 3

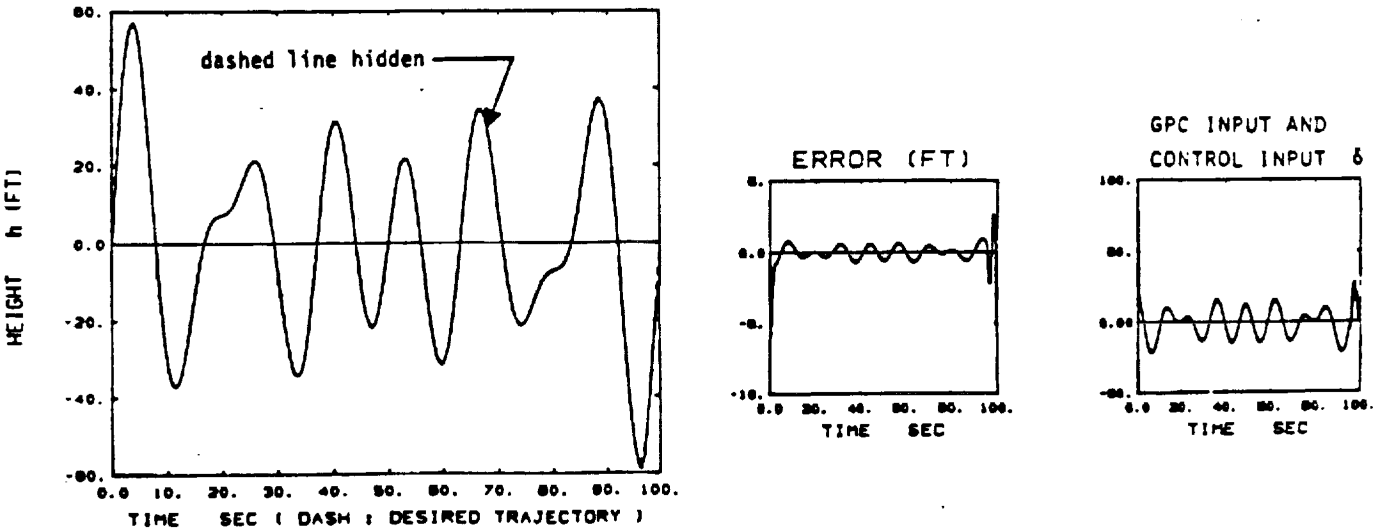


Fig. 5 Terrain following performance of system of Fig. 1
with GPC

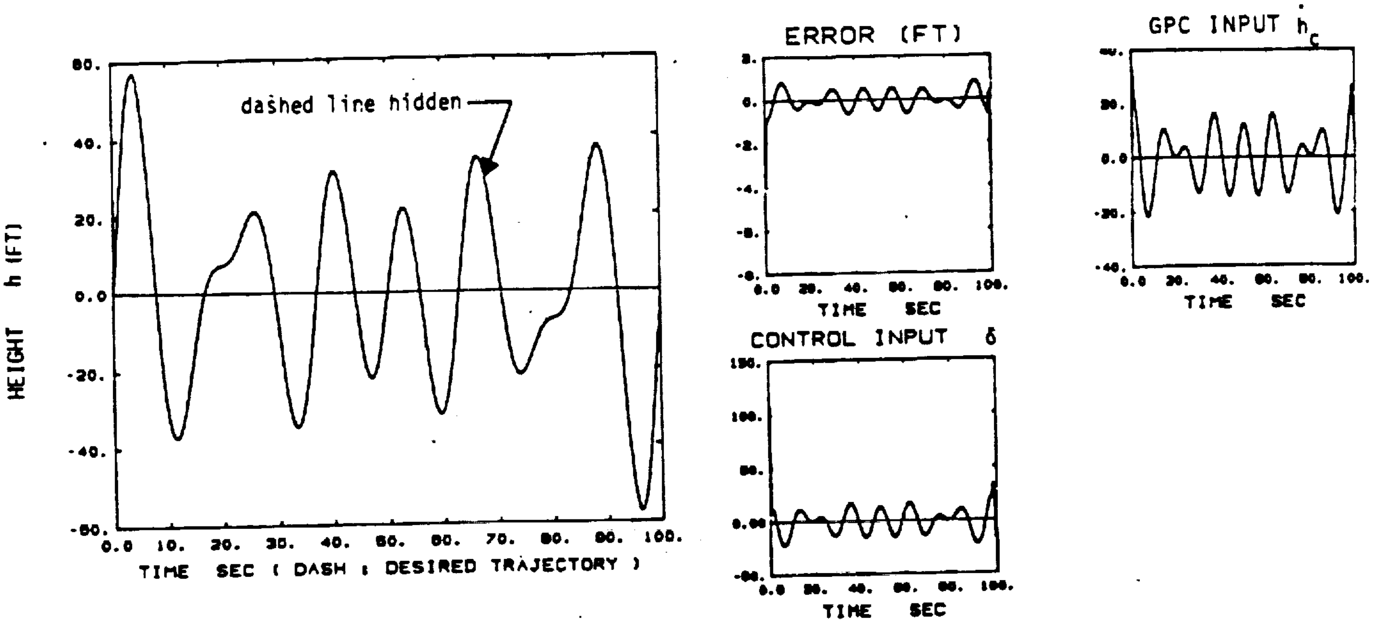


Fig. 6 Terrain following performance of system of Fig. 2 with GPC

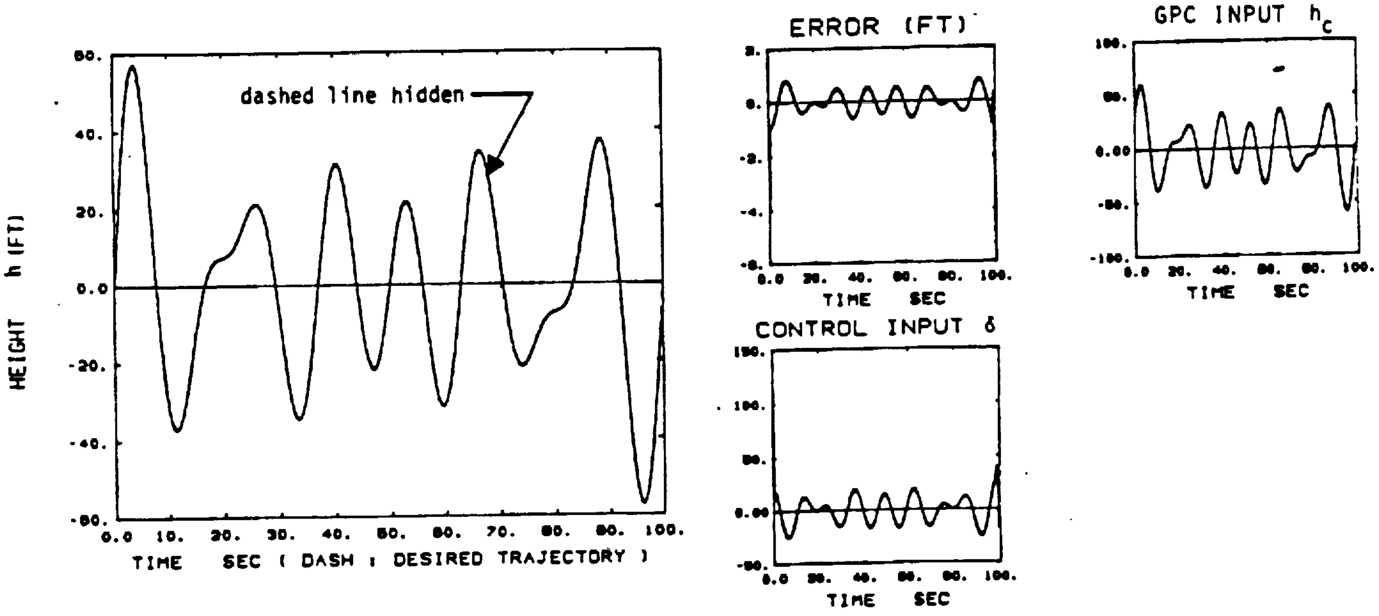


Fig. 7 Terrain following performance of system of Fig. 3 with GPC

ORIGINAL PAGE IS OF POOR QUALITY

An Application of Generalized Predictive Control to Rotorcraft Terrain-Following Flight

RONALD A. HESS, MEMBER, IEEE, AND YOON C. JUNG

Abstract —Generalized predictive control (GPC) describes an algorithm for the control of dynamic systems in which a control input is generated which minimizes a quadratic cost function consisting of a weighted sum of errors between desired and predicted future system output and future predicted control increments. The output predictions are obtained from an internal model of the plant dynamics. The GPC approach is similar in concept to manual preview control. The GPC algorithm is first applied to a simplified rotorcraft terrain-following problem, and GPC performance is compared to that of a conventional compensatory automatic system in terms of flight path following, control activity, and control law implementation. Next, more realistic vehicle dynamics are utilized, and the GPC algorithm is applied to simultaneous terrain following and velocity control in the presence of atmospheric disturbances and errors in the internal model of the vehicle.

I. Introduction

A. Background

In many manual control tasks, the ability of the human operator to "look ahead" or "preview" is a vital strategy in achieving acceptable human/machine performance. In addition, models of human preview control have often employed an "internal model" of the plant dynamics with which the human is presumed to generate predictions of future plant output given current plant state and present and future control inputs, e.g., [1], [2]. Over the past decade, a technique has been introduced for the design of automatic controllers, called variously model predictive heuristic control, model algorithmic control, or output predictive control. This technique for the design of inanimate controllers approximates the activity of the human preview controller [3]–[5].

More recently Clarke and Zhang [6] and Clarke *et al.* [7] have introduced generalized predictive control (GPC) and have related it to the earlier approaches of [3]–[5] and linear quadratic (LQ) designs. It is the GPC approach that is the subject of the research reported herein. This marks the first application of this particular algorithm to a flight control problem. Details of the GPC algorithm can be found in [7]; however a brief review of the salient features of the approach will be undertaken in the following sections.

B. The GPC Algorithm

The plant is modeled in discrete fashion using the so-called controlled autoregressive integrated moving average model [7]:

$$A(q^{-1})y(k)=B(q^{-1})u(k-1)+\xi(k)/\Delta \qquad k=0,1,2,\cdots, \tag{1}$$

where $A(q^{-1})$ and $B(q^{-1})$ are polynomials in the delay operator q^{-1} and $y(k)$ and $u(k)$ are output and control variables, respectively, $\xi(k)$ is an uncorrelated random sequence, and Δ represents the differencing operator $(1-q^{-1})$. The actual sampling interval is T, so that at each sampling instant, the independent variable in (1) is kT. Now a prediction of the plant output, given measured output up to time kT and control input $u(k+i)$ for $i\leqslant -1$, is

$$\hat{y}(k+j|k)=G_j\Delta u(k+j-1)+F_jy(k) \tag{2}$$

where j is the number of future time steps being predicted, $G_j(q^{-1})=E_jB$, and E_j results from a recursive solution of the Diophantine relation

$$1=E_j(q^{-1})A\Delta+q^{-j}F_j(q^{-1}). \tag{3}$$

Here, E_j and F_j are polynomials uniquely defined, given $A(q^{-1})$ and the integer j.

Now a predictive control law can be defined as that which minimizes a cost function given by

$$J(N_1,N_2)=E\left[\sum_{j=N_1}^{N_2}[\hat{y}(k+j)-w(k+j)]^2 + \sum_{j=1}^{N_2}\lambda(j)[\Delta u(k+j-1)]^2\right] \tag{4}$$

Manuscript received September 11, 1988; revised March 3, 1989. This work was supported in part by the Aircraft Guidance and Navigation Branch of NASA Ames Research Center under Grant NAG 2-221.

The material in this paper was presented at the 1988 IEEE International Conference on Systems, Man, and Cybernetics, Beijing and Shenyang, China, August 8–12.

The authors are with the Department of Mechanical Engineering, University of California, Davis, CA 95616.

IEEE Log Number 8928405.

0018-9472/89/0900-0955$01.00 ©1989 IEEE

where

N_1 minimum costing horizon,
N_2 maximum costing horizon,
$w(k)$ desired value of the output y at the kth sampling instant,
$\lambda(k)$ control weighting sequence.

Equation (4) is concerned only with a subset of future time defined $N_2 T$ seconds into the future and is dependent upon data up to time kT. Note how the control is generated: at each sampling instant, an optimal control sequence for N_2 steps into the future is calculated; however only the first of these is applied to the plant. At the next sampling instant a new optimal sequence is calculated that minimizes J for N_2 steps into the future, but again, only the first of these is applied to the plant. This defines a "receding horizon" optimization procedure. The control law so obtained has been classified as open-loop feedback-optimal with an autoregressive disturbance process [7]. The authors feel that this control philosophy is similar to that used by the human operator when controlling plants for which the desired future output can be defined, e.g., automobile driving or aircraft flight path control in near-earth flight.

Significant reductions in the order of the matrices involved in computing the optimal control can be made by requiring that, after an interval $NU < N_2$, projected control increments are assumed to be zero, i.e.,

$$\Delta u(k+j-1) = 0, \qquad j > NU \tag{5}$$

where NU is called the "control horizon." This procedure is equivalent to placing infinite weights on control increments after some future time $NU \cdot T$. In addition to computational simplifications, the introduction of the control horizon allows the stable control of nonminimum phase plants [7].

With the introduction of the control horizon, the prediction equations become

$$\hat{\boldsymbol{y}} = \boldsymbol{G}_1 \tilde{\boldsymbol{u}} + \boldsymbol{f} \tag{6}$$

where

$$\hat{\boldsymbol{y}} = [\hat{y}(k+1), \hat{y}(k+2), \cdots, \hat{y}(k+N)]^T$$

$$\tilde{\boldsymbol{u}} = [\Delta u(k), \Delta u(k+1), \cdots, \Delta u(k+N-1)]^T$$

$$\boldsymbol{f} = [f(k+1), f(k+2), \cdots, f(k+N)]^T$$

N = output horizon = N_2 here.

$$\boldsymbol{G}_1 = \begin{bmatrix} g_0 & 0 & \cdots & 0 \\ g_1 & g_0 & & \\ \cdot & \cdots & \cdots & 0 \\ \vdots & \cdot & & g_0 \\ \vdots & \cdot & & \vdots \\ g_{N-1} & g_{N-2} & \cdots & g_{N-NU} \end{bmatrix} \tag{7}$$

with $f(k+j)$ being that component of $\hat{y}(k+j)$ composed of signals which are known at time kT [7], and the g_i are elements of the polynomial $G_j(q^{-1})$, itself obtained from the recursive Diophantine relation of (3). The corresponding control law is given by

$$\tilde{\boldsymbol{u}} = (\boldsymbol{G}_1^T \boldsymbol{G}_1 + \lambda \boldsymbol{I})^{-1} \boldsymbol{G}_1^T (\boldsymbol{w} - \boldsymbol{f})$$

where

$$\boldsymbol{w} = [w(k+1), w(k+2), \cdots, w(k+N)]^T. \tag{8}$$

The matrix involved in the inversion in (8) is of dimension $NU \times NU$. Equation (8) and the pertinent relations preceding it define the GPC algorithm. Some theoretical stability results can be obtained by relating GPC to state-space LQ control laws. The reader is referred to [7] for details.

A number of parameters are obviously available as design variables or "tuning knobs" in applications of the GPC algorithm. They are the minimum and maximum costing horizons N_1 and N_2, the control horizon NU, and the control weighting sequence $\lambda(k)$. The role played by these parameters is best demonstrated by means of the flight control examples of the following sections.

II. Flight Control Applications

A. Introduction

Terrain following or contour flight is defined as flight at low altitude which conforms generally to the contours of the terrain and gross vegatation features [8]. Each leg of contour flight is typically characterized by a constant vehicle heading but varying velocity and altitude as dictated by vegatation, obstacles, and ambient light. The response requirements of flight path control systems for terrain-following flight involve relatively high bandwidth command-following characteristics. The ability of the human pilot to complete such tasks successfully has led to the investigation of pertinent preview control models for near-earth flight [9]. The similarity between the philosophy of these models and that of the GPC approach led the authors to a consideration of the latter algorithm as a candidate for automatic flight path control in the terrain-following task. Indeed Reid *et al.* [5] have applied an output predictive algorithm to a terrain-following flight control task. Conceptually at least, this algorithm is a special case of GPC, as it considers the control input to be held constant over a multiple of sampling intervals, then provides a least-square control solution that minimizes a cost function similar to (4), but with no weighting on control inputs, a minimum output horizon of zero, and a control horizon matching the maximum output horizon. The necessity of holding the control input constant over a number of sampling intervals was necessary to ensure stability.

In conducting preliminary evaluations of the output predictive algorithm for the height or altitude control task to be considered here, performance was, in general, unsatisfactory. The necessity of holding the control input constant for multiples of the sampling interval coupled with the lack of control weighting in the cost function led to the

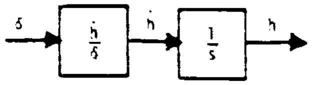


Fig. 1. Bare airframe, single-DOF vehicle.

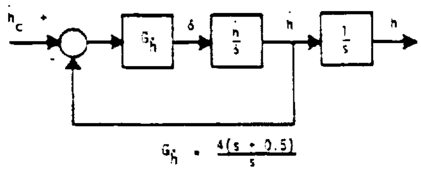


Fig. 2. Bare airframe with vertical velocity augmentation loop.

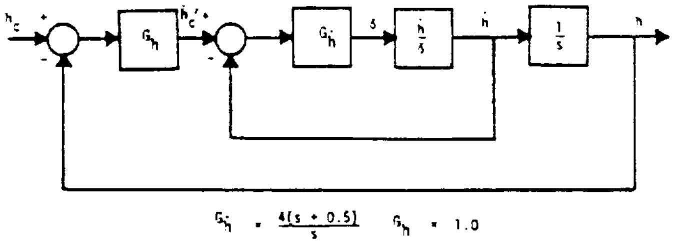


Fig. 3. Bare airframe with vertical velocity and altitude augmentation loops.

generation of unrealistic control inputs for the rotorcraft height control task, i.e., control inputs which resembled relaylike functions alternating between large positive and negative amplitudes. For this reason the output predictive algorithm was eschewed in favor of the GPC approach to be described.

B. Terrain Following — Single-Degree-of-Freedom (DOF) Vehicle Dynamics

Figs. 1–3 show the three "plants" which were utilized in the first example of this study. They involve a simplified rotorcraft "bare airframe" vertical velocity to collective control input transfer function given by

$$\frac{\dot{h}}{\delta}(s) = \frac{-(s-20)}{(s+1)(s+20)} \doteq \frac{e^{-0.1s}}{(s+1)}. \tag{9}$$

The time delay and the Padè approximation have been included to provide a challenge to the control algorithm since nonminimum phase dynamics are involved. Fig. 1 represents the bare airframe in that no stability/control augmentation is included in the plant that is controlled by the GPC algorithm. In this case the control input for GPC is the collective control and the plant is given by (9). Fig. 2 represents the bare airframe with a vertical velocity control loop closed about it. Here the control input for GPC is commanded vertical velocity $\dot{h}_c$. The effective plant for this case is

$$\frac{h}{\dot{h}_c}(s) = \frac{-4(s+0.5)(s-20)}{s(s^3+17s^2+98s+40)} = \frac{-4(s+0.5)(s-20)}{s(s+0.44)\left[s^2+2(0.87)(9.52)s+9.52^2\right]}. \tag{10}$$

Finally Fig. 3 represents the bare airframe with vertical velocity and height control augmentation. Here the control input for GPC is commanded height h_c. The effective plant for this case is

$$\frac{h}{h_c}(s) = \frac{-4(s+0.5)(s-20)}{s^4+17s^3+94s^2+118s+40} = \frac{-4(s+0.5)(s-20)}{(s+1)(s+0.579)\left[s^2+2(0.93)(8.31)s+8.31^2\right]} \tag{11}$$

The rationale for selecting the dynamic systems of Figs. 1–3 was that they represented the range of possible levels of GPC utilization in a typical flight control application from inner loop control actuator commands in Fig. 1 to outer loop flight path guidance commands in Fig. 3. The plants of (9)–(11) were discretized using a 0.1-s sampling interval for the purpose of obtaining the GPC algorithm. In the simulations to be described, the plants were, of course, modeled as continuous systems. The selection of the sampling interval equal to the time delay of (9) was merely out of convenience.

The commanded vertical flight path trajectory for this application was a time history similar to that utilized in [5], represented as a sum of sinusoids

$$h_c = 20\left[\sin(0.05(2\pi t)) + \sin(0.06(2\pi t)) + \sin(0.08(2\pi t))\right] \quad \text{ft.} \tag{12}$$

Equation (12) can be thought of as representing a commanded flight path that would be provided by an on-board computer in a terrain-following task.

In implementing the GPC algorithm, the desired output was a vehicle trajectory that was an exponential curve that continuously defined a smooth capture trajectory from the vehicle's present position to the command of (12). This is graphically portrayed in Fig. 4. Once again this control philosophy was felt to be similar to that employed by the human in path tracking tasks with preview. The capture trajectory was given by

$$h_{\text{cap}}(k+j) = h_c(k+j) - \exp(-\tau_e j)\left[h_c(k+j) - h(k)\right] \qquad j = 1,2,\cdots,N_2. \tag{13}$$

Although the time constant τ_e could serve as another design variable in the GPC algorithm, it was maintained at 0.5 s for this study. Thus the time to 50- and 95-percent amplitudes for the trajectory of (13) was 0.14 and 0.6 s, respectively. The compensation $G_{\dot{h}}$ and G_h were obtained by first selecting inner and outer loop crossover frequencies of 4 and 1 rad/s, respectively, and then ensuring that the open-loop transfer functions in each loop closure of Fig. 3 ($G_{\dot{h}}(\dot{h}/\delta)$ and $G_h(h/\dot{h}_c')$) resembled an integrator in the region of open-loop crossover.

Fig. 5 shows the performance of the system of Fig. 3 without GPC and with the command trajectory of (12) serving as the system input. This serves as a benchmark system for GPC performance comparisons as it represents

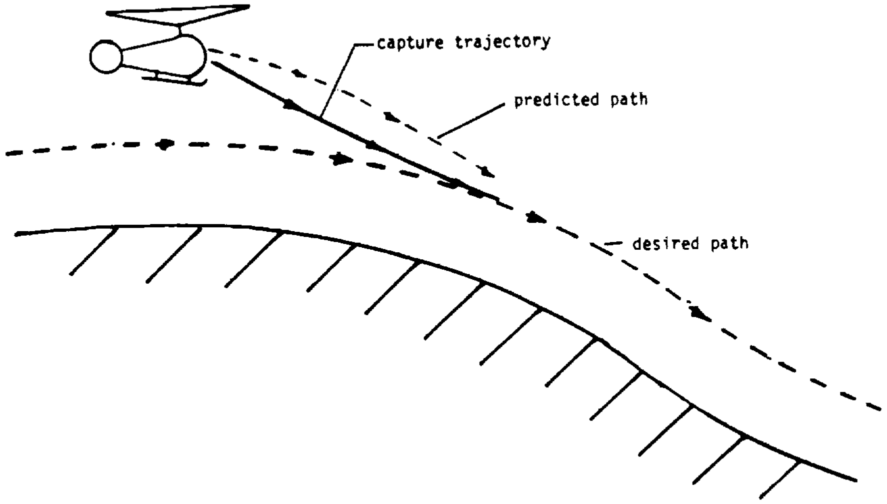


Fig. 4. Predictive control in terrain-following task.

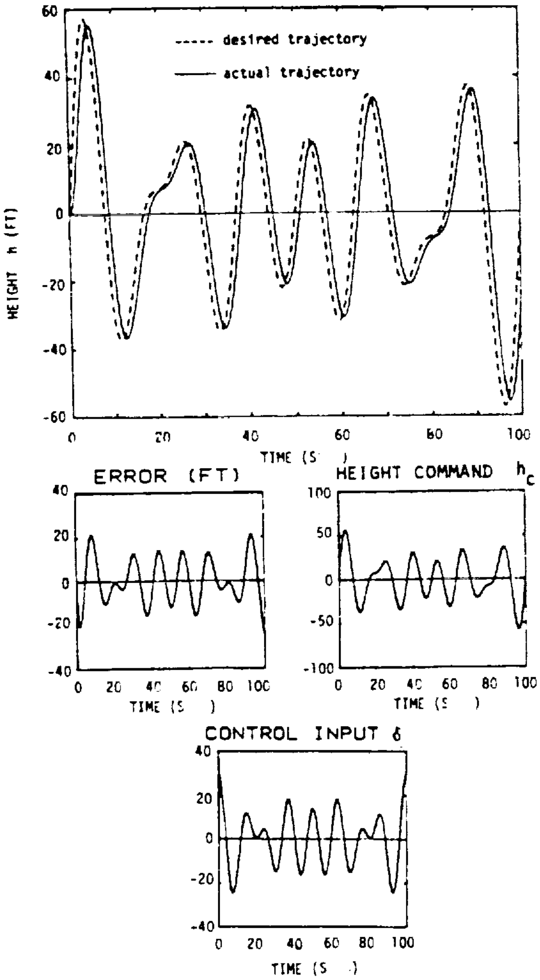


Fig. 5. Terrain-following performance of system of Fig. 3 without GPC (classical design).

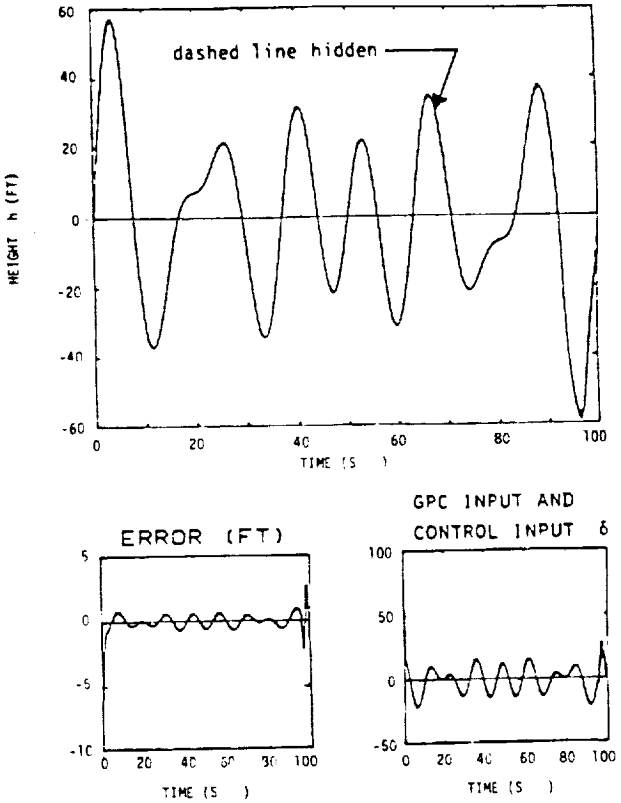


Fig. 6. Terrain-following performance of system of Fig. 1 with GPC.

the performance of a "classical" multiloop control design with fairly high loop bandwidths. Note the height errors exceed 20 ft in some instances. This classical design has been discretized with the same 0.1 s sampling interval as that used in the GPC implementation to be discussed. The GPC parameters for *all* the applications were determined by trial and error as

$$N_1 = \text{minimum output horizon} = 1\ (0.1\ \text{s}),$$
$$N_2 = \text{maximum output horizon} = 50\ (5\ \text{s}),$$
$$NU = \text{control horizon} = 20\ (2\ \text{s}),$$
$$\lambda = \text{control weighting sequence} = 0.2.$$

Figs. 6–8 show the performance of the GPC systems. The commanded and actual vehicle trajectories (dashed and solid lines, respectively) are indistinguishable in these figures because of the excellent tracking performance as

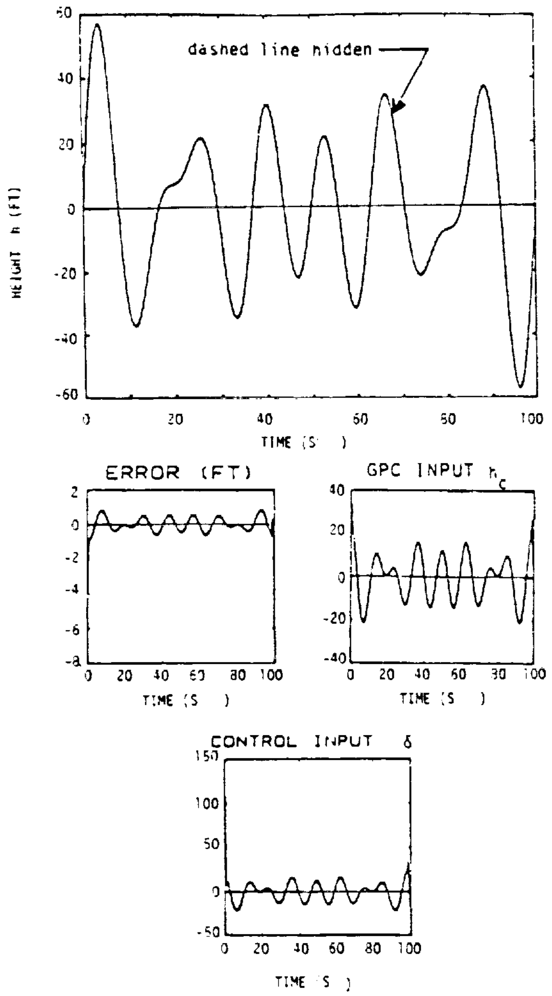


Fig. 7. Terrain-following performance of system of Fig. 2 with GPC.

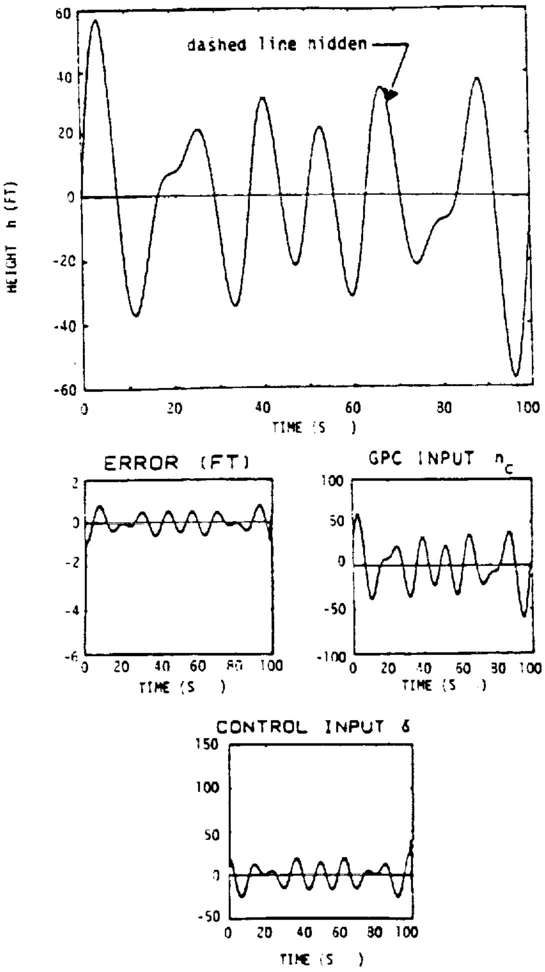


Fig. 8. Terrain-following performance of system of Fig. 3 with GPC.

indicated by the small height errors. With the exception of the initial and final transients, these errors are less than 1 ft in magnitude. The transients are due to the abrupt initiation and termination of the sum of sinusoids at the beginning and end of the simulation. The figure parts labeled "GPC input" represent the control as provided by the GPC algorithm (u in (1)), and this input varies from the systems of Figs. 1–3. The excellent performance of the GPC designs is evident in all the systems with performance deteriorating slightly as one moves from the system of Fig. 1 to that of Fig. 3.

C. Terrain Following and Velocity Control Multi-DOF Vehicle Dynamics

The commanded vertical flight path in this example is identical to that given by (12). In addition however the vehicle was required to follow a sinusoidal varying longitudinal velocity command given by

$$u_c(t) = 20[\sin(0.05(2\pi)t] \qquad \text{ft/s} \qquad (14)$$

where $u_c(t)$ represents an additive command to the trim air speed of $u_o = 101$ ft/s (60 kn). The bare-airframe vehicle dynamics are now described by the following multi-DOF state space equations

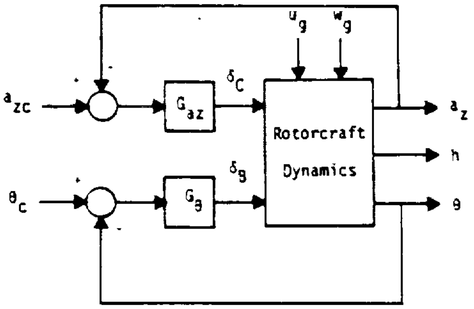


Fig. 9. Multi-DOF vehicle with augmentation.

$$\begin{bmatrix} \dot{u} \\ \dot{w} \\ \dot{q} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} -0.01 & 0 & 0 & -32.2 \\ 0 & -1 & 101 & 0 \\ 0 & 0 & -5.6 & -6.25 \\ 0 & 0 & 1 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 1.5 \\ 0.133 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \delta_B \\ \delta_C \end{bmatrix}. \qquad (15)$$

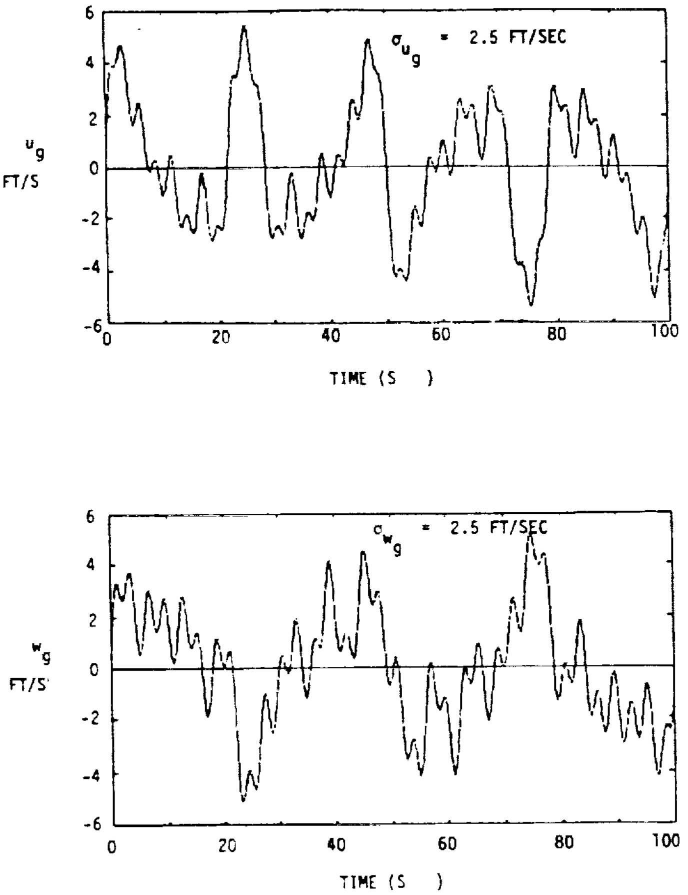


Fig. 10. Longitudinal and vertical velocity disturbances simulating turbulence.

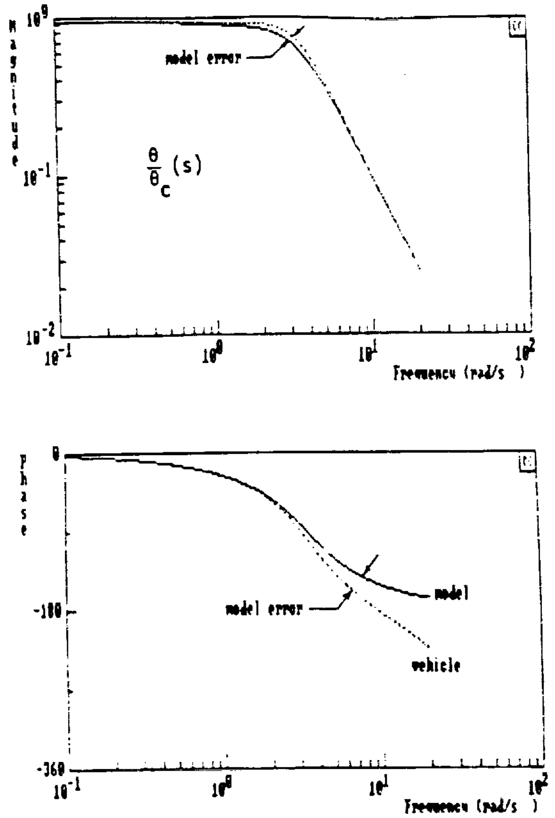


Fig. 11. Effect of internal model error on pitch attitude augmentation loop.

In (15), u and w represent longitudinal and vertical velocity perturbations, q represents pitch rate, θ represents pitch attitude, and δ_B and δ_C represent the longitudinal cyclic and collective inputs, respectively. It is assumed that the rotorcraft in question also possesses a pitch attitude and vertical acceleration stability augmentation system as shown in Fig. 9. Thus the particular level at which the GPC algorithm is introduced here is similar to that of Fig. 2 of the previous example, i.e., only inner stability augmentation loops have been closed around the bare airframe prior to the application of GPC. For the vehicle dynamics of (15), the augmentation transfer functions G_{a_z} and G_θ are given by

$$G_\theta = \frac{909[(s/1.2)+1]}{[(s/0.1)+1]}$$

$$G_{a_z} = \frac{1.39(s+1)}{s^2}. \tag{16}$$

This compensation yields pitch attitude and vertical acceleration systems each with closed-loop bandwidths of 2 rad/s. The requirement for simultaneous control of both altitude and longitudinal velocity offers an interesting challenge to the GPC design because of the inherent dynamic coupling of these variables in a rotorcraft. That is, longitudinal velocity is controlled by changing vehicle pitch attitude that also produces disturbances in vehicle altitude.

In addition to the height and velocity commands, the effects of atmospheric turbulence were simulated by adding sums of sinusoids representing filtered white noise to the vertical and longitudinal vehicle velocities u and w. The time histories of these perturbation velocities are shown in Fig. 10 and possess rms values of 2.5 ft/s. Finally an internal model error was deliberately introduced into the simulation in the form of a 0.05-s time delay in the control inputs δ_B and δ_C. This delay was *not* included in the GPC design, i.e., in calculating the G_1 matrix of (7). Fig. 11 shows the effect of this error on the closed-pitch attitude system. As in the examples of Section II-B, the sampling interval was 0.1 s. The GPC parameters were obtained by trial and error as

$$\begin{aligned}
N_1 &= \text{minimum output horizon} &&= 1\ (0.1\ \text{s}),\\
N_2 &= \text{maximum output horizon} &&= 20\ (2\ \text{s}),\\
NU &= \text{control horizon} &&= 10\ (1\ \text{s}),\\
\lambda &= \text{control weighting sequence} &&= 7\cdot 10^3\ (\Theta_c)\\
& && \quad \text{and } 1.0\ (A_{zc}).
\end{aligned}$$

Figs. 12 and 13 show the simulation results. Once again with the exception of initial transients, system path-follow-

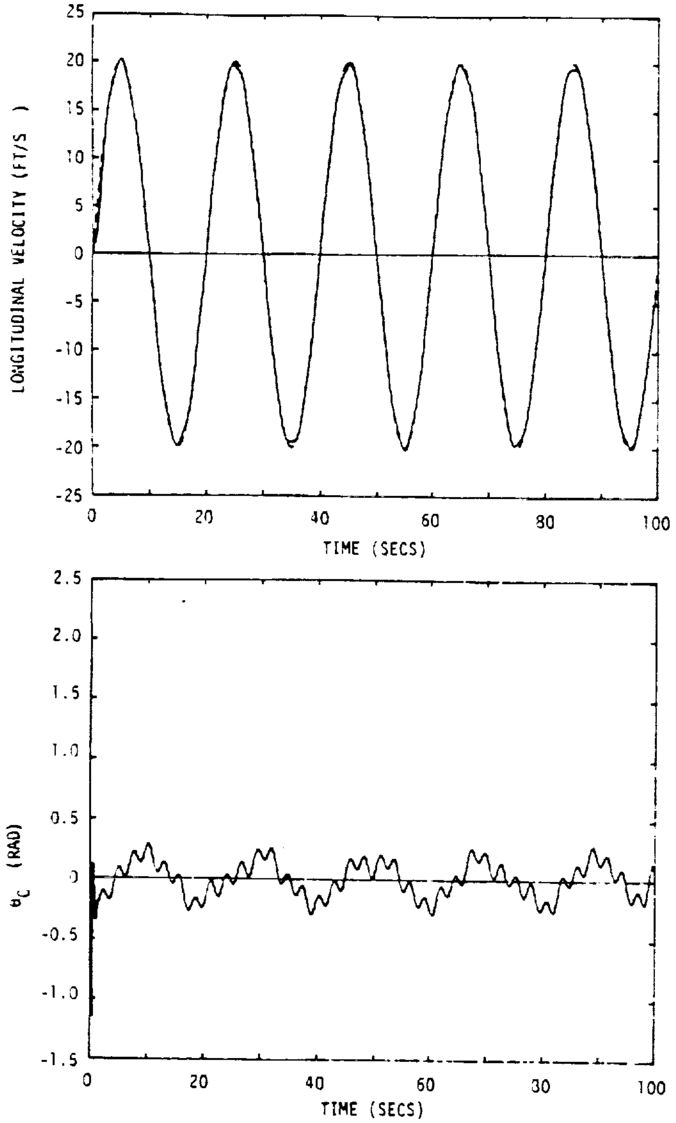


Fig. 12. Longitudinal velocity tracking performance of system of Fig. 9 with GPC.

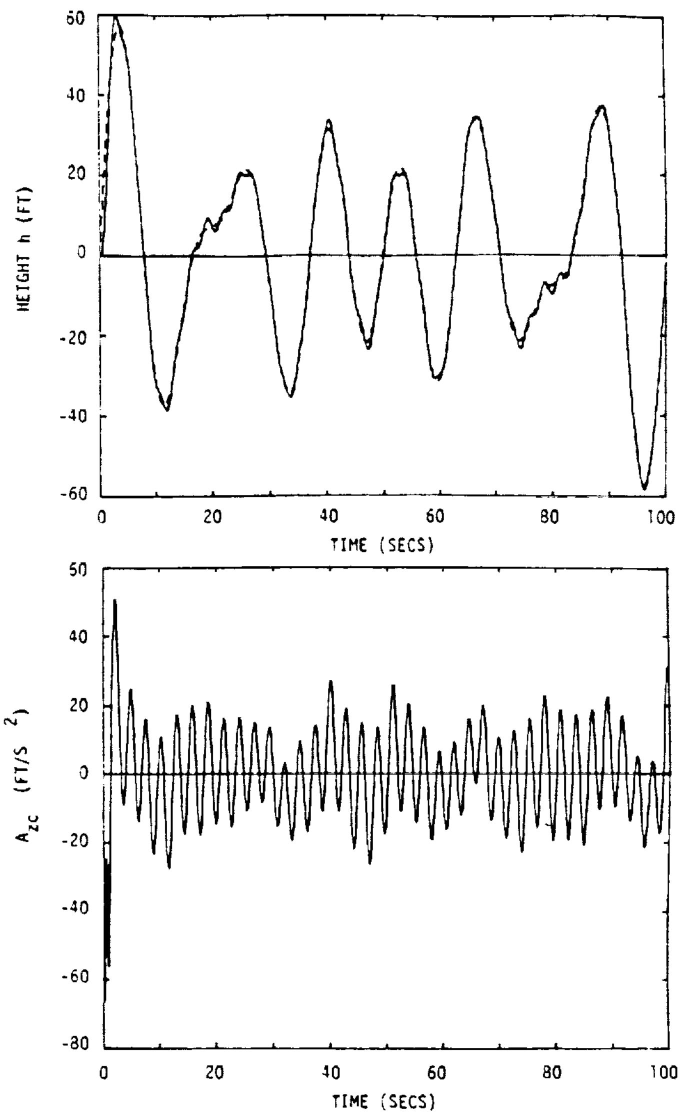


Fig. 13. Terrain-following performance of system of Fig. 9 with GPC.

ing and velocity-tracking performance are excellent. Overall performance is not adversely affected by the aforementioned dynamic coupling, disturbance environment or modeling errors. The transients in A_{zc} and Θ_c that occur at the initiation of the run could be eliminated by allowing a time-varying τ_e in the capture trajectory of (13).

III. Manual Control Applications

Although not pursued in this study, the application of the GPC algorithm to the description of manual control tasks in which desired future output can be defined appears promising. A task that comes to mind immediately is automobile driving. The inclusion of weightings on control *rate* in the cost function of (4) as is typically done in the optimal control model of the human operator [10] can be accomplished by suitable modification of the GPC algorithm [7]. The basic format of the GPC approach, with its output and control horizons, its internal model, and its output (as opposed to state) feedback structure would appear to make it a worthy candidate for future research in the manual control area.

IV. Conclusion

The GPC algorithm has the potential of offering flight path and velocity control performance far superior to that obtainable with classical designs in the demanding environment of terrain-following flight. In addition the GPC algorithm can be successfully introduced at a number of different points in a control hierarchy, from inner loop control actuator commands to outer loop guidance commands.

The on-line computational and sensing requirements for implementing the GPC algorithm are minimal. The gain matrix G_1 can be calculated off-line and only vehicle output need be measured.

Internal model inaccuracies and disturbances require adjustment of the GPC parameters or "tuning knobs" but appear to be quite manageable. Use of the GPC algorithm for manual control models appears worth pursuing.

References

[1] T. B. Sheridan, "Three models of preview control," *IEEE Trans. Human Factors Electron.*, vol. HFE-7, no. 2, pp. 91-102, June 1966.

[2] C. E. Kelley, *Manual and Automatic Control.* New York: Wiley, 1968, ch. 15.

[3] J. Richalet, A. Rault, J. L. Testud, and J. Papon, "Model predictive heuristic control: Applications to industrial processes," *Automatica*, vol. 14, no. 5, pp. 413-428, Sept. 1978.

[4] R. Rouhani and R. K. Mehra, "Model algorithmic control (MAC): Basic theoretical properties," *Automatica*, vol. 18, no. 4, pp. 401-414, July 1982.

[5] J. G. Reid, D. E. Chaffin, and J. T. Silverthorn, "Output predictive algorithmic control: Precision tracking with applications to terrain following," *J. Guidance Contr.*, vol. 4, no. 5, pp. 502-509, Sept.-Oct. 1981.

[6] D. W. Clarke and L. Zhang, "Long-range predictive control using weighting-sequence models," *Proc. Inst. Elec. Eng.*, vol. 134, pt. D., no. 3, May 1987, pp. 187-195.

[7] D. W. Clarke, C. Mohtadi, and P. S. Tuffs, "Generalized predictive control, Parts I and II," *Automatica*, vol. 23, no. 2, Mar. 1987, pp. 137-160.

[8] V. H. L. Cheng and B. Sridhar, "Considerations for automated nap-of-the-earth rotorcraft flight," in *Proc. 1988 American Control Conf.*, Atlanta, GA, pp. 967-976.

[9] R. A. Hess and K. K. Chan, "A preview control pilot model for near-earth maneuvering helicopter flight," *J. Guidance, Contr. Dynamics*, vol. 11, no. 2, pp. 148-152, Mar.-Apr. 1988.

[10] R. A. Hess, "Feedback control models," in *Handbook of Human Factors*, G. Salvendy, Ed. New York: Wiley, 1987, ch. 9.5.

Ronald A. Hess (M'87) received the B.S., M.S., and Ph.D. degrees in aerospace engineering from the University of Cincinnati, Cincinnati, OH, in 1965, 1967, and 1970, respectively.

In 1982 he joined the faculty of the Department of Mechanical Engineering at the University of California, Davis, where he is currently a Professor in the Division of Aeronautical Science and Engineering. His current research interests lie in the areas of automatic and manual control of aircraft and in man/machine systems.

Dr. Hess is an Associate Fellow of the AIAA, a member of Sigma Xi, and is an Associate Editor of the *Journal of Aircraft* and the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS. He is Chairman of the Manual Control Technical Committee of the Systems, Man, and Cybernetics Society of IEEE and is a member of the AIAA Technical Committee on Atmospheric Flight Mechanics.

Yoon C. Jung, was born in Seoul, Korea on November 3, 1958. He received the B.S. and M.S. degrees in mechanics and production engineering from Seoul National University, Seoul, Korea, in 1981 and 1983, respectively.

From 1984 to 1986 he was a research engineer at Korea Power Engineering Company. He is now with the Department of Mechanical Engineering at the University of California, Davis, as a Ph.D. student.

SELF-TUNING GENERALIZED PREDICTIVE CONTROL APPLIED TO TERRAIN FOLLOWING FLIGHT

R. A. Hess[1] and Y. C. Jung[2]
Division of Aeronautical Science and Engineering
Department of Mechanical Engineering
University of California, Davis

Abstract

Generalized Predictive Control (GPC) describes an algorithm for the control of dynamic systems in which a control input is generated which minimizes a quadratic cost function consisting of a weighted sum of errors between desired and predicted future system output and future predicted control increments. The output predictions are obtained from an internal model of the plant dynamics. Self-tuning GPC refers to an implementation of the GPC algorithm in which the parameters of the internal model(s) are estimated on-line and the predictive control law tuned to the parameters so identified. The self-tuning GPC algorithm is applied to a problem of rotorcraft longitudinal/vertical terrain-following flight. The ability of the algorithm to tune to the initial vehicle parameters and to successfully adapt to a stability augmentation failure is demonstrated. Flight path performance is compared to a conventional, classically designed flight path control system.

[1] Professor, Associate Fellow AIAA

[2] Graduate student

Introduction

Over the past decade, a general technique has been introduced for the design of automatic controllers, called variously, Model Predictive Heuristic Control, Model Algorithmic Control, Output Predictive Control, Dynamic Matrix Control, etc. [1-4]. More recently, Clarke and Zhang [4] and Clarke, et al., [5] have introduced Generalized Predictive Control (GPC) and have related it to the earlier approaches of Ref. 1-4 and Linear Quadratic (LQ) designs and have incorporated self-tuning in the control algorithm. Cast in terms of the flight path control problem which will be the subject of the research to be described the GPC algorithm can be summarized as follows [6]:

1.) At each present time instant t_i, a prediction of the vehicle path over a relatively long range time horizon (j sampling periods) is made. This prediction is obtained from a model of the vehicle dynamics.

2.) A control strategy for the next NU sampling intervals is selected which brings the predicted vehicle path back to a desired path in the "best" way according to a specific control objective, i.e, LQ, or model following, etc.

3.) The resulting "best" control is then applied but only over the next sampling interval, and at the next sampling instant the whole procedure is repeated which results in a continuously updated control action with corrections based upon the latest measurements.

Hess and Jung have shown the potential performance improvements which accrue when GPC is used in lieu of so-called "classical" design techniques in a rotorcraft vertical flight path control problem [7]. The research to be described herein will extend the results of Ref. 7 to include self-tuning control.

The Self-Tuning GPC Algorithm

The GPC Algorithm

Details of the GPC algorithm, itself, can be found in Ref. 5,, however a brief review of the salient features of the approach will be undertaken in what follows:

The plant is modeled in discrete fashion using the so-called Controlled Auto-Regressive Integrated Moving Average (CARIMA) model [5]:

$$\begin{aligned} A(q^{-1})y(k) &= B(q^{-1})u(k-1) + \xi(k)/\Delta \\ k &= 0,\ 1,\ 2,\ \text{etc.} \end{aligned} \tag{1}$$

where $A(q^{-1})$ and $B(q^{-1})$ are polynomials in the delay operator q^{-1}, $y(k)$ and $u(k)$ are output and control variables, respectively, $\xi(k)$ is an uncorrelated random sequence, and Δ represents the differencing operator $(1 - q^{-1})$. The actual sampling interval is T, so that, at each sampling instant, the independent variable in Eq. 1 is kT. Now a prediction of the plant output, given measured output up to time kT and control input $u(k + i)$ for $i \leq -1$, is

$$\hat{y}(k + j|k) = G_j\Delta u(k + j - 1) + F_j y(k) \tag{2}$$

where

j = the number of future time steps being predicted

$G_j(q^{-1})$ = E_jB and where E_j results from a recursive solution of the Diophantine relation [8]

$$1 = E_j(q^{-1})A\Delta + q^{-j}F_j(q^{-1}) \tag{3}$$

Here, E_j and F_j are polynomials uniquely defined, given $A(q^{-1})$ and the integer j.

Now a predictive control law can be defined as that which minimizes a cost function given by

$$J(N_1,N_2) = E\left[\sum_{j=N_1}^{N_2}[\hat{y}(k+j) - w(k+j)]^2 + \sum_{j=1}^{N_2}\lambda(j)[\Delta u(k+j-1]^2\right] \tag{4}$$

where

N_1 = the minimum costing horizon

N_2 = the maximum costing horizon

$w(k)$ = the desired value of the output y at the kth sampling instant

$\lambda(j)$ = a control weighting sequence

Equation 4 is concerned only with a subset of future time defined N_2T secs into the future and is dependent upon data up to time kT. As outlined in the Introduction, the control is generated in the following manner: At each sampling instant, an optimal control sequence for N_2 steps into the future is calculated, however only the first of theses is applied to the plant. At the next sampling instant, a new optimal sequence is calculated which minimizes J for N_2 steps into the future, but again, only the first of these is applied to the plant. This defines a "receding horizon" strategy.

Significant reductions in the order of the matrices involved in computing the optimal control can be made by requiring that, after an interval $NU < N_2$, projected control increments are assumed to be zero, i.e.,

$$\Delta u(k + j - 1) = 0 \qquad j > NU \tag{5}$$

where NU is called the "control horizon". This procedure is equivalent to placing infinite weights on control increments after a future time NU•T. With the introduction of the control horizon, the prediction equations become

$$\underline{\hat{y}} = \underline{G}_1\underline{\tilde{u}} + \underline{f} \tag{6}$$

where

$$\begin{aligned}
\underline{\hat{y}} &= [\hat{y}(k + 1), \hat{y}(k + 2), \ldots.\hat{y}(k + N)]^T \\
\underline{\tilde{u}} &= [\Delta u(k), \Delta u(k + 1), \ldots.\Delta u(k + N - 1)]^T \\
\underline{f} &= [f(k + 1), f(k + 2), \ldots.f(k + N)]^T \\
N &= \text{output horizon} = N_2 \text{ here.}
\end{aligned} \tag{7}$$

$$\underline{G}_1 = \begin{bmatrix} g_0 & 0 & \cdots & 0 \\ g_1 & g_0 & & \\ \cdot & \cdot\cdot & \cdots & 0 \\ \vdots & \cdot & & g_0 \\ \vdots & \cdot & & \vdots \\ g_{N-1} & g_{N-2} & \cdots & g_{N-NU} \end{bmatrix}$$

with f(k +j) being that component of $\hat{y}(k + j)$ composed of signals which are known at time kT, and the g_i are elements of the polynomial $G_1(q^{-1})$, itself obtained from the recursive Diophantine relation of Eq. 3. The corresponding control law is given by:

$$\tilde{\underline{u}} = (\underline{G}_1^T \underline{G}_1 + \lambda \underline{I})^{-1} \underline{G}_1^T (\underline{w} - \underline{f}) \tag{8a}$$

where

$$\underline{w} = [w(k + 1), w(k + 2), \ldots . w(k + N)]^T \tag{8b}$$

In implementing the GPC algorithm, the "desired" output is usually described as an exponential curve which continuously defines a smooth capture trajectory from the present output to the commanded output defined over future time. The capture trajectory can be given by

$$h_c'(k + j) = h_c(k + j) - \exp(-\tau_e j)[h_c(k + j) - h(k)] \tag{9}$$
$$j = 1, 2, \ldots, N_2$$

Self-Tuning

As implemented herein, self-tuning control refers to the on-line identification of the elements of the polynomials $A(q^{-1})$ and $B(q^{-1})$ in the CARIMA model of Eq. 1. Both standard Recursive Least Squares (RLS) and Extended Least Squares (ELS) algorithms were mechanized as identification algorithms with a "forgetting factor" β where $0 < \beta < 1$ [9], and relying upon UDU covariance factorization [10]. In the example of this study, $\beta = 1.0$ and only the results for RLS will be discussed. The ELS technique can remove estimation biases which can occur with the RLS technique operating in a low signal to noise environment, however ELS is not as robust as RLS in practice [9].

Rotorcraft Terrain Following Example

Terrain-following or contour flight is defined as flight at low altitude which conforms generally to the contours of the terrain and gross vegetation features [11]. Each leg of contour flight is typically characterized by a constant vehicle heading but varying velocity and altitude as dictated by vegetation, obstacles, and ambient light. The response requirements of flight path control systems for terrain following flight involve relatively high bandwidth command following characteristics and provide a challenging test for the self-tuning GPC algorithm. Figure 1 is a graphical portrayal of GPC as applied to the terrain following problem.

The bare-airframe rotorcraft dynamics to be used in this study are given by the following set of linear longitudinal state equations

$$\begin{bmatrix} \dot{u} \\ \dot{w} \\ \dot{q} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} -0.01 & 0 & 0 & -32.2 \\ 0 & -1 & 101 & 0 \\ 0 & 0 & -5.6 & -6.25 \\ 0 & 0 & 1 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 1.5 \\ 0.133 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \delta_B \\ \delta_C \end{bmatrix} \qquad (10)$$

where

u = longitudinal velocity perturbation in x body axis, ft/sec

w = vertical velocity perturbation in z body axis, ft/sec

q = pitch rate, rad/sec

θ = pitch attitude, rad

U_o = trim airspeed, 101 ft/sec (60 kts)

δ_b = longitudinal cyclic control, % of full deflection

δ_c = collective control, % of full deflection

Although it is possible to apply the self-tuning GPC algorithm to the bare-airframe dynamics, the authors felt is more realistic to apply the algorithm to a vehicle which already possessed a stability augmentation system. Figure 2 shows the stability augmentation systems that will be utilized herein. For the vehicle dynamics of Eq. 10 the augmentation transfer functions G_{az} and G_{θ} are given by

$$G_{\theta} = \frac{909[(s/1.2) + 1]}{[(s/0.1) + 1]}$$

$$G_{az} = \frac{1.39(s + 1)}{s^2} \qquad (11)$$

This compensation yields pitch attitude and vertical acceleration systems each with closed-loop bandwidths of 2 rad/sec.

The commanded vertical flight path trajectory is the same as used in Ref. 7 and is represented as a sum of sinusoids

$$h_c(t) = 20[1 - \exp(-.05t)][\sin(.05(2\pi t)) + \sin(.06(2\pi t)) + \sin(.08(2\pi t))] \text{ ft} \tag{12}$$

The [1 - exp(-.05t)] term is included to prevent the initial transients from causing unrealistically large control inputs in the simulation. In addition, to the vertical flight path command of Eq. 12, the vehicle is required to follow a sinusoidal longitudinal velocity command given by

$$u_c(t) = 20[1 - \exp(-.05t)][\sin(.05(2\pi t))] \qquad \text{ft/sec} \tag{13}$$

The [1 - exp(-.05t)] term is again included to prevent initial transients. The requirement for simultaneous control of both altitude and longitudinal velocity is a challenging task because of the inherent dynamic coupling of these variables in the rotorcraft model of Eq. 10, i.e., longitudinal velocity is controlled by changing vehicle attitude which also produces disturbances in vehicle altitude.

Finally, in addition to the height and velocity commands, the effects of atmospheric turbulence were simulated by adding gust terms u_g and w_g to the perturbation velocities u and w, respectively. Time histories of these gust terms are shown in Fig. 3 and are represented by sums of sinusoids approximating filtered white noise possessing RMS values of 2.5 ft/sec.

Finally, the following nonlinear kinematic equations were employed to describe the vehicle flight path:

$$\dot{h} = (U_o + u)\sin\theta - w\cos\theta$$

$$A_{zc} = \dot{w} - (U_o + u)q \qquad (14)$$

Interpreting the self-tuning procedure in the z-domain, the internal model of the vehicle used in the self-tuning was represented by the following two transfer functions

$$u/\theta_c = [a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3} + a_4 z^{-4}]/[1 - b_1 z^{-1} - b_2 z^{-2} - b_3 z^{-3} - b_4 z^{-4}]$$

$$h/Azc = [c_1 z^{-1} + c_2 z^{-2} + c_3 z^{-3} + c_4 z^{-4}]/[1 - d_1 z^{-1} - d_2 z^{-2} - d_3 z^{-3} - d_4 z^{-4}] \qquad (15)$$

where the values of the coefficients a_i, b_i, c_i, and d_i were the end product of the RLS/ELS techniques.

Simulation results for the rotorcraft terrain following problem were obtained under the following conditions:

(1) The GPC activity begins with a control law developed from a nominal model of the vehicle. This nominal model employed reduced-order system transfer functions which capitalized upon near pole-zero cancellation in the system model. These simplifications are shown in Table 1. Although the GPC is calculated initially using the reduced-order models, the vehicle simulation

always uses the complete state space model of Eq. 10 with the stability augmentation system of Eq. 11 and Fig. 2. In this example the following GPC parameter values were selected on the basis of simulation trial and error:

$$N1 = 1 \text{ (0.1 secs)}$$

$$N2 = 20 \text{ (2 secs)}$$

$$NU = 10 \text{ (1 sec)}$$

$$\tau_e = 0.5 \text{ secs}$$

$$\lambda_{a_{zc}} = 10$$

$$\lambda_{\theta_c} = 7 \cdot 10^4$$

(2) During the first 20 seconds of flight, the self-tuning algorithm identifies the coefficients in Eqs. 15 on-line, starting from initial estimates of zero, i.e. not using the nominal model. This 20 second period does not represent a minimum tuning time, and was simply chosen with an eye toward allowing enough tuning time to adequately identify low frequency dynamics.

(3) Using the results of the system identification in step (2), a revised control law is implemented using Eq. 8a.

(4) Forty seconds into the run, the gain on the transfer function G_{az} in Eq. 11 was halved, simulating a "soft" failure in the stability augmentation system responsible for vertical acceleration control. The self-tuning algorithm was allowed to identify the modified vehicle dynamics on-line for 10 seconds. At the end of this period, a second, revised control law was

implemented using Eq. 8a. As in the case in step (2), the 10 secs does not represent a minimum tuning period, and was a tradeoff between accurate but rapid identification.

The results of a simulation of the self-tuning GPC system are shown in Figs. 4-7. In Figs. 4 and 6, the dashed lines represent the commanded altitude and velocity values. By way of comparison, Figs. 8-11 show the same simulation without the gust perturbations. It should be noted that, without self-tuning, the stability augmentation "failure" produced an unstable system with GPC. However, the time to double amplitude was sufficiently large (approximately 6 secs) so that the 10 sec self-tuning period did not produced excessively large perturbations. The instability is demonstrated in Fig. 12 which shows the altitude response of the system with the augmentation failure but without self-tuning being initiated at 40 secs. Note the altitude scale.

Finally, for the sake of comparison a classical control system design was implemented in the terrain following task. Referring to the diagram of Fig. 2, this classical design was implemented by allowing

$$\begin{aligned} \theta_c &= -.0182\theta_e \quad \text{rad/(ft/sec)} \\ A_{zc} &= -(\dot{h}_e + 0.1h_e) \quad (\text{ft/sec}^2)/\text{ft} \end{aligned} \tag{18}$$

These outer-loop compensators resulted in velocity and altitude loops with bandwidths near 1 rad/sec, which is quite high. Figures 13-16 show the simulation results for this classical design with turbulence. As in the case with the self-tuning GPC design, the augmentation failure was introduced at 40 secs. In this case, of course, no adaptation is occurring and the natural robustness of the classical design maintains stability. A comparison of Figs. 4 and 13, and 6 and 15, clearly demonstrates the performance superiority of the self-tuning GPC design. The performance of the classical design could, of course, be improved by feeding forward weighted derivatives of the future desired path and velocity commands to the stability augmentation inputs Θ_c and A_{zc}. However, this modification makes use of desired path information which is not required by the GPC algorithm, and so was not included in the comparison.

As an example of the quality of the RLS identification technique, Figs. 18 and 19 show the actual and identified u/Θ_c and h/A_{zc} transfer functions interpreted in the w' plane for the initial self-tuning period. Here, the actual transfer function refers to that obtained with the complete state space model of Eq. 10 and the stability augmentation system of Eq. 11 and Fig. 2. It is encouraging to note that the GPC design is robust enough to tolerate the errors in the identified dynamics.

Conclusions

1.) A self-tuning capability added to a GPC algorithm as applied to a problem of rotorcraft terrain following flight has yielded a flight path control system with exceptional performance.

2.) The sensor requirements of the self-tuning GPC algorithm are minimal. The only measurements beyond that required for operation of the stability augmentation system are instantaneous vehicle altitude and velocity.

3.) On-board computational requirements also are quite manageable. The RLS and ELS techniques for system identification use an efficient computational algorithms. Once the tuning process is complete, the optimal control law is generated via the matrix multiplications called out in Eq. 8a.

4.) The remaining significant theoretical development currently being pursued is a method for relating system robustness to GPC design parameter selection.

Acknowledgement

This research was supported by Grant No. NAG 2-221 from the Aircraft Guidance and Navigation Branch and the Flight Dynamics and Controls Branch of NASA Ames Research Center, Moffett Field, CA.

References

[1] Richalet, J., Rault, A., Testud, J. L., Papon, J., "Model Predictive Heuristic Control: Applications to Industrial Processes," Automatica, Vol. 14, No. 5, Sept. 1978, pp. 413-428.

[2] Rouhani, R., and Mehra, R. K., "Model Algorithmic Control (MAC): Basic Theoretical Properties," Automatica, Vol. 18, No., 4, July 1982, pp. 401-414.

[3] Reid, J. G., Chaffin, D. E., and Silverthorn, J. T., "Output Predictive Algorithmic Control: Precision Tracking with Application to Terrain

Following," Journal of Guidance and Control, Vol. 4, No. 5, Sept-Oct., 1981, pp. 502-509.

[4] Clarke, D. W., and Zhang, L., "Long-Rage Predictive Control Using Weighting Sequence Models," IEE Proceedings, Vol. 134, Pt. D., No. 3, May 1987, pp. 187-195.

[5] Clarke, D. W., Mohtadi, C., and Tuffs, P. S., "Generalized Predictive Control, Parts I and II," Automatica, Vol., 23, No. 2, March 1987, pp. 137-160.

[6] De Keyser, R. M. C., Van De Velde, G. A., Dumortier, F. A. G., "A Comparative Study of Self-Adaptive Long-Range Predictive Control Methods," Automatica, Vol. 24, No. 2, March 1988, pp. 149-163.

[7] Hess, R. A., and Jung, Y. C., "An Application of Generalized Predictive Control to Rotorcraft Terrain-Following Flight," submitted to IEEE Transactions on Systems, Man, and Cybernetics.

[8] Astrom, K. J., and Wittenmark, B., Adaptive Control, Addison-Wesley,

[9] Clarke, D. W., Self-Tuning and Adaptive Control, H. Nicholson, Ed., Chaps. 2 and 6, Peter Perengrinus, 1981.

[10] Thornton, C. L., and Bierman, G. J., "Filtering and Error Analysis Via the UDU Covariance Factorisation," IEEE Transactions on Automatic Control,

Vol. AC-23, No. 5, Oct. 1978, pp. 901-907.

Figure Captions

Fig. 1 The terrain following task.

Fig. 2 The stability augmentation system (SAS).

Fig. 3 The turbulence inputs.

Fig. 4 Terrain following performance, self-tuning GPC, turbulence.

Fig. 5 GPC normal acceleration input to SAS, self-tuning, turbulence.

Fig. 6 Velocity command following performance, self-tuning GPC, turbulence.

Fig. 7 GPC pitch attitude input to SAS, self-tuning, turbulence.

Fig. 8 Terrain following performance, self-tuning GPC, no turbulence.

Fig. 9 GPC normal acceleration input to SAS, self-tuning, no turbulence.

Fig. 10 Velocity command following performance, self-tuning GPC, no turbulence.

Fig. 11 GPC pitch attitude input to SAS, self-tuning, no turbulence.

Fig. 12 Unstable terrain following performance with no self-tuning after SAS failure.

Fig. 13 Terrain following performance, classical design, turbulence.

Fig. 14 Normal acceleration input to SAS, classical design, turbulence.

Fig. 15 Velocity command following performance, classical design, turbulence.

Fig. 16 Pitch attitude input to SAS, classical design, turbulence.

Fig. 17 Comparison of w' plane transfer functions for u/Θ_c, actual vehicle and RLS identification in initial self-tuning.

Fig. 18 Comparison of w' plane transfer functions for h/A_{zc}, actual vehicle and RLS identification in initial self-tuning.

Table 1 System transfer functions

system transfer functions (with SAS)

$$\frac{u}{\theta_c} = \frac{-356.8476s^2 - 356.8476s}{s^6 + 6.71s^5 + 12.577s^4 + 18.6423s^3 + 11.8924s^2 + 0.1171s} \qquad \frac{h}{a_{zc}} = \frac{-2.085s^4 - 13.782s^3 - 24.845s^2 - 13.278s - 0.13}{s^7 + 8.695s^6 + 25.698s^5 + 31.213s^4 + 13.341s^3 + 0.13s^2}$$

reduced-order transfer functions

$$\frac{u}{\theta_c} = \frac{-356.8476}{s^4 + 5.71s^3 + 6.867s^2 + 11.775s + 0.1171} \qquad \frac{h}{a_{zc}} = \frac{-2.085}{s^2(s + 2.0847)}$$

z-transfer functions of reduced-order system

$$\frac{u}{\theta c} = \frac{-0.0013z^{-1} - 0.0131z^{-2} - 0.0117z^{-3} - 0.0009z^{-4}}{1 - 3.5081z^{-1} + 4.59z^{-2} - 2.6469z^{-3} + 0.565z^{-4}} \qquad \frac{h}{a_{zc}} = \frac{-0.0003z^{-1} - 0.0013z^{-2} - 0.0003z^{-3}}{1 - 2.8118z^{-1} + 2.6236z^{-2} - 0.8118z^{-3}}$$

(sampling time = 0.1 sec) (sampling time = 0.1 sec)

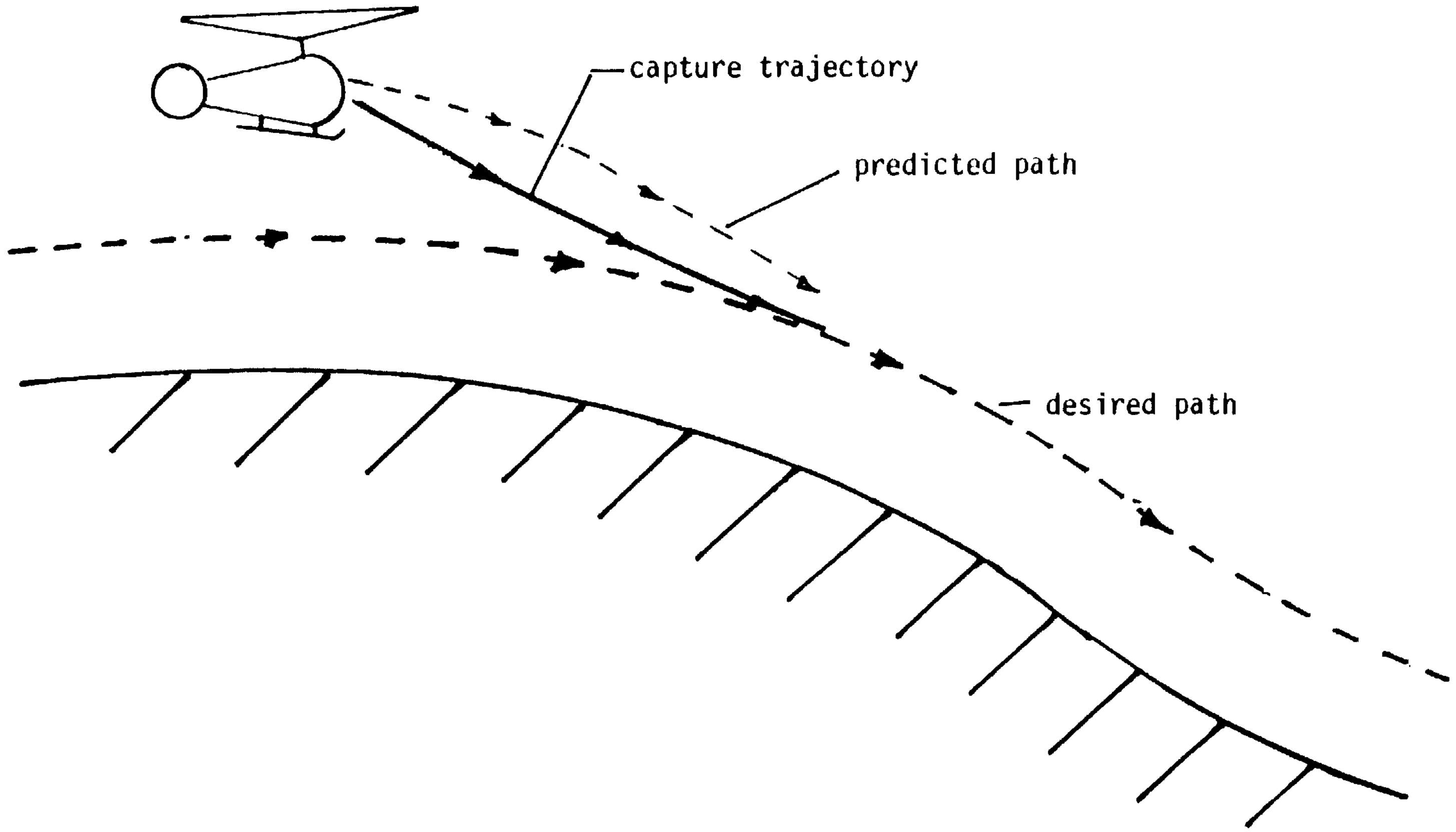
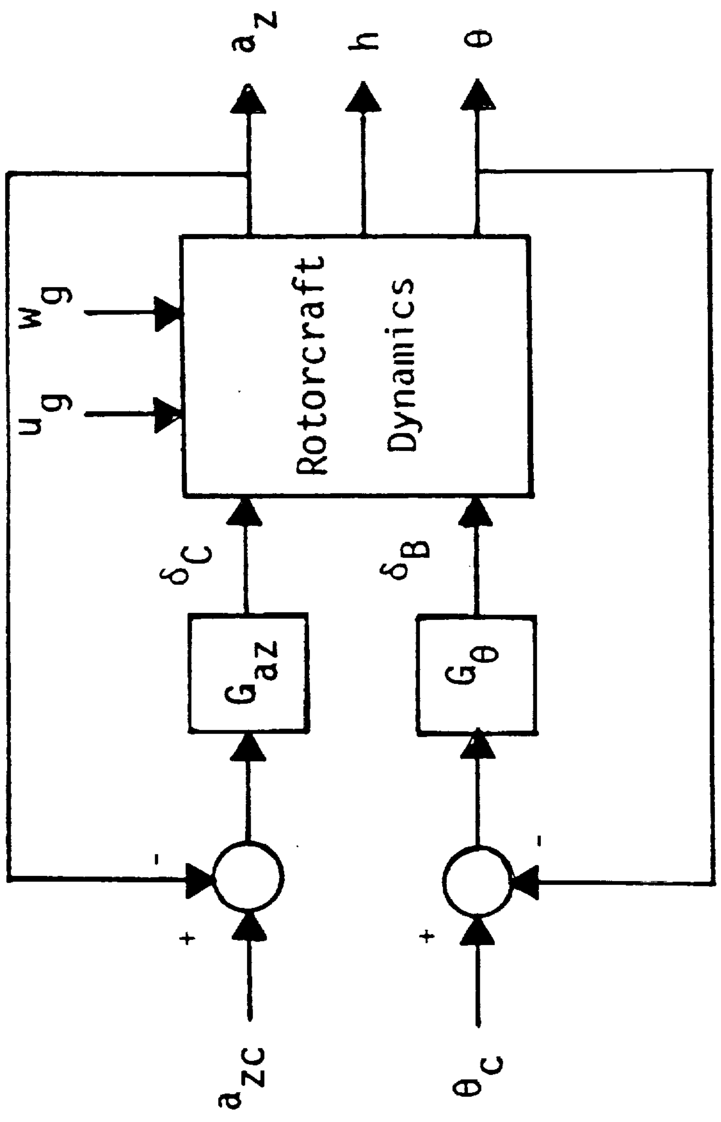
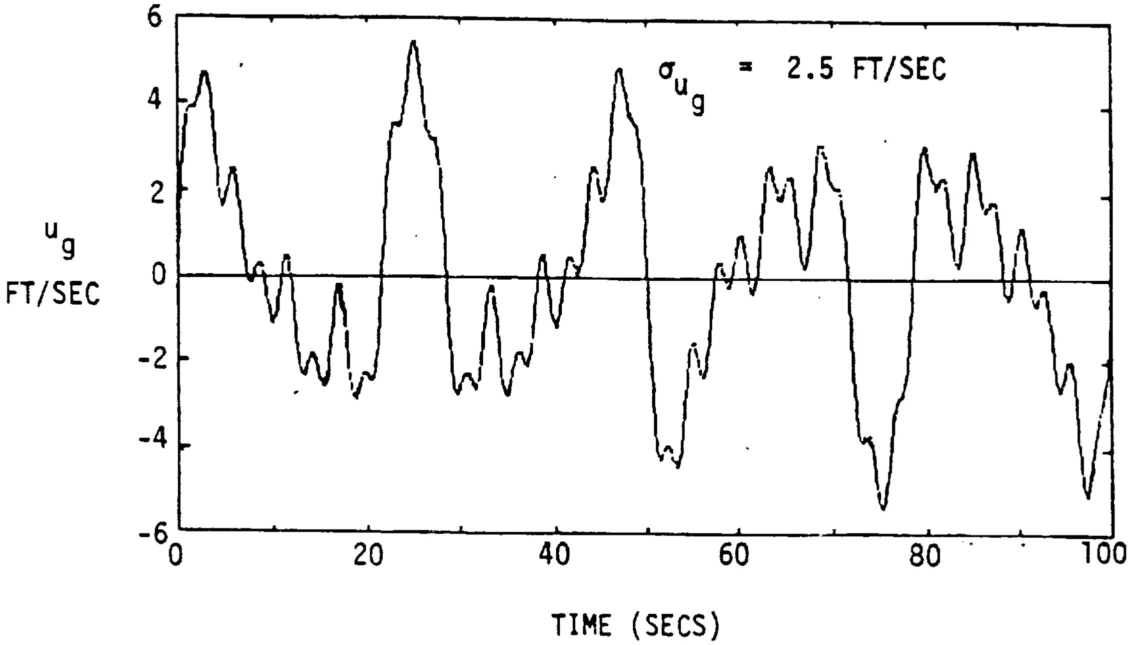


Fig. 1

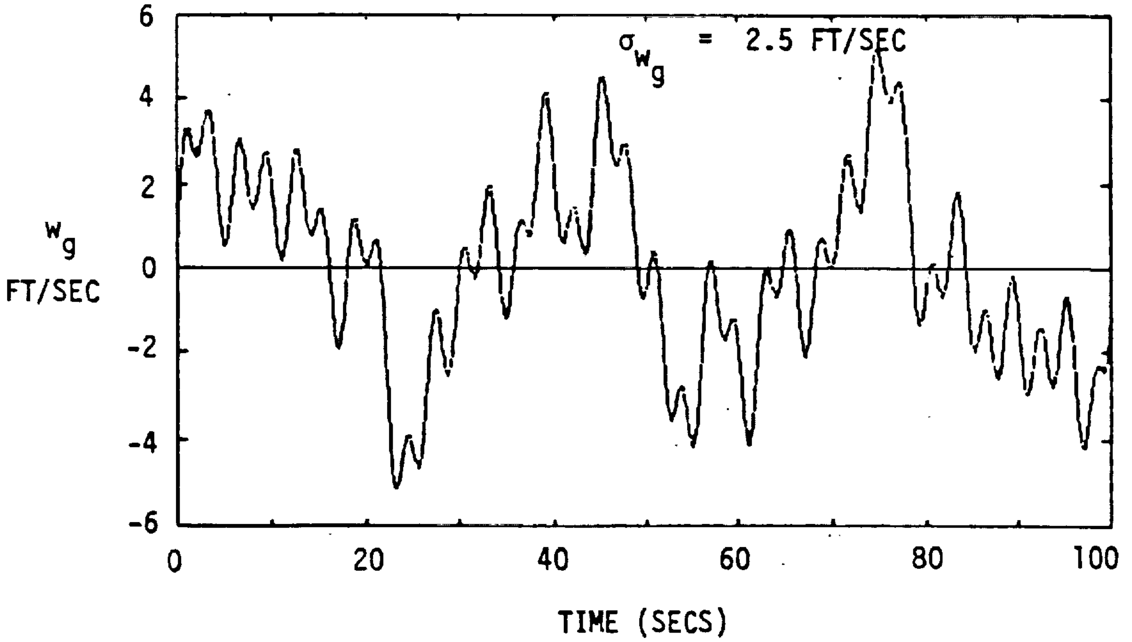


a_{zc}
θ_c
+
-
+
-
G_{az}
G_θ
δ_C
δ_B
u_g
w_g
Rotorcraft Dynamics
a_z
h
θ

Fig. 2



6
4
2
0
-2
-4
-6
u_g
FT/SEC
σ_{u_g} = 2.5 FT/SEC
0
20
40
60
80
100
TIME (SECS)



6
4
2
0
-2
-4
-6
w_g
FT/SEC
σ_{w_g} = 2.5 FT/SEC
0
20
40
60
80
100
TIME (SECS)

Fig. 3

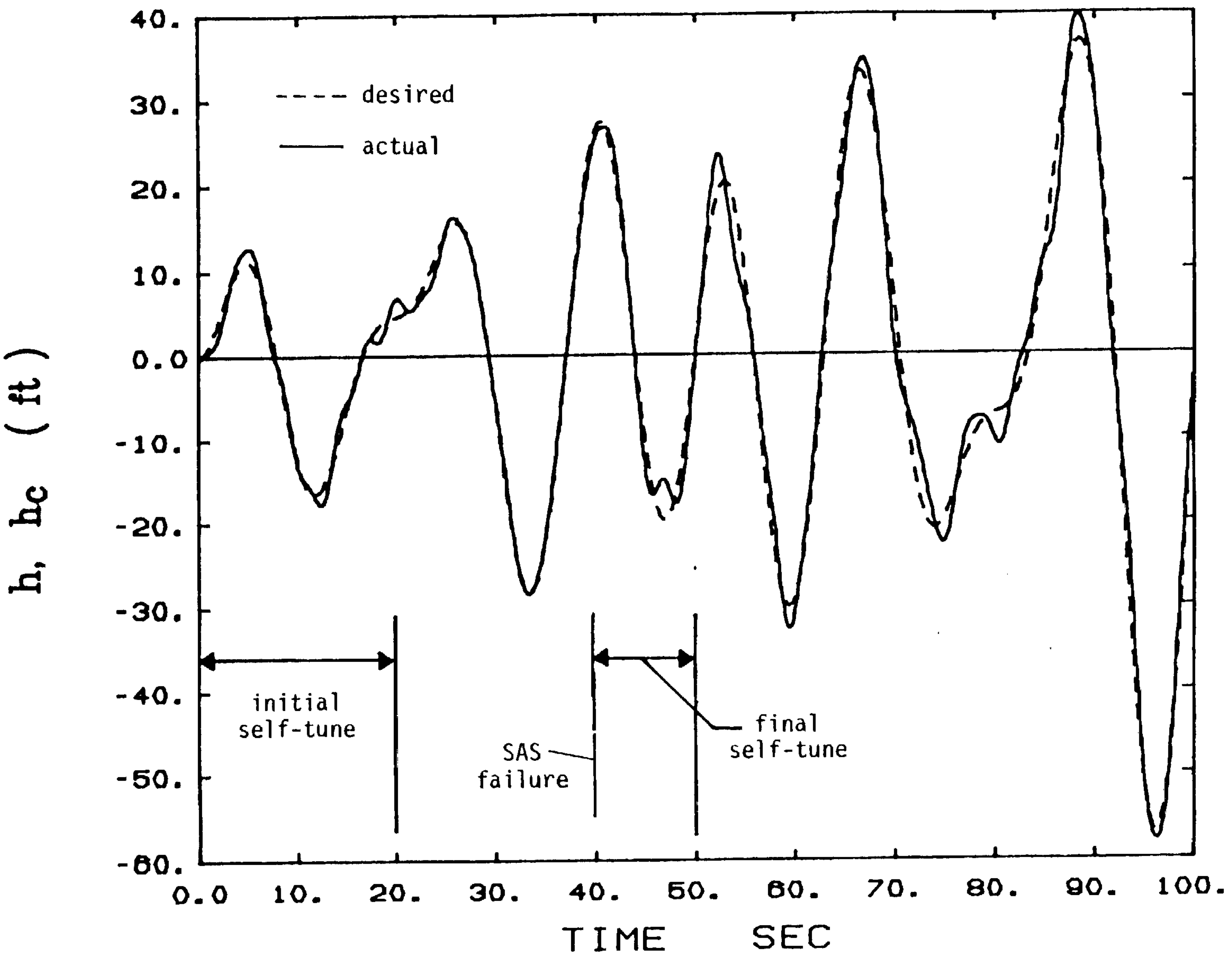

---- desired
—— actual
h, h_c (ft)
40.
30.
20.
10.
0.0
-10.
-20.
-30.
-40.
-50.
-60.
0.0
10.
20.
30.
40.
50.
60.
70.
80.
90.
100.
TIME SEC
initial
self-tune
SAS
failure
final
self-tune

Fig. 4

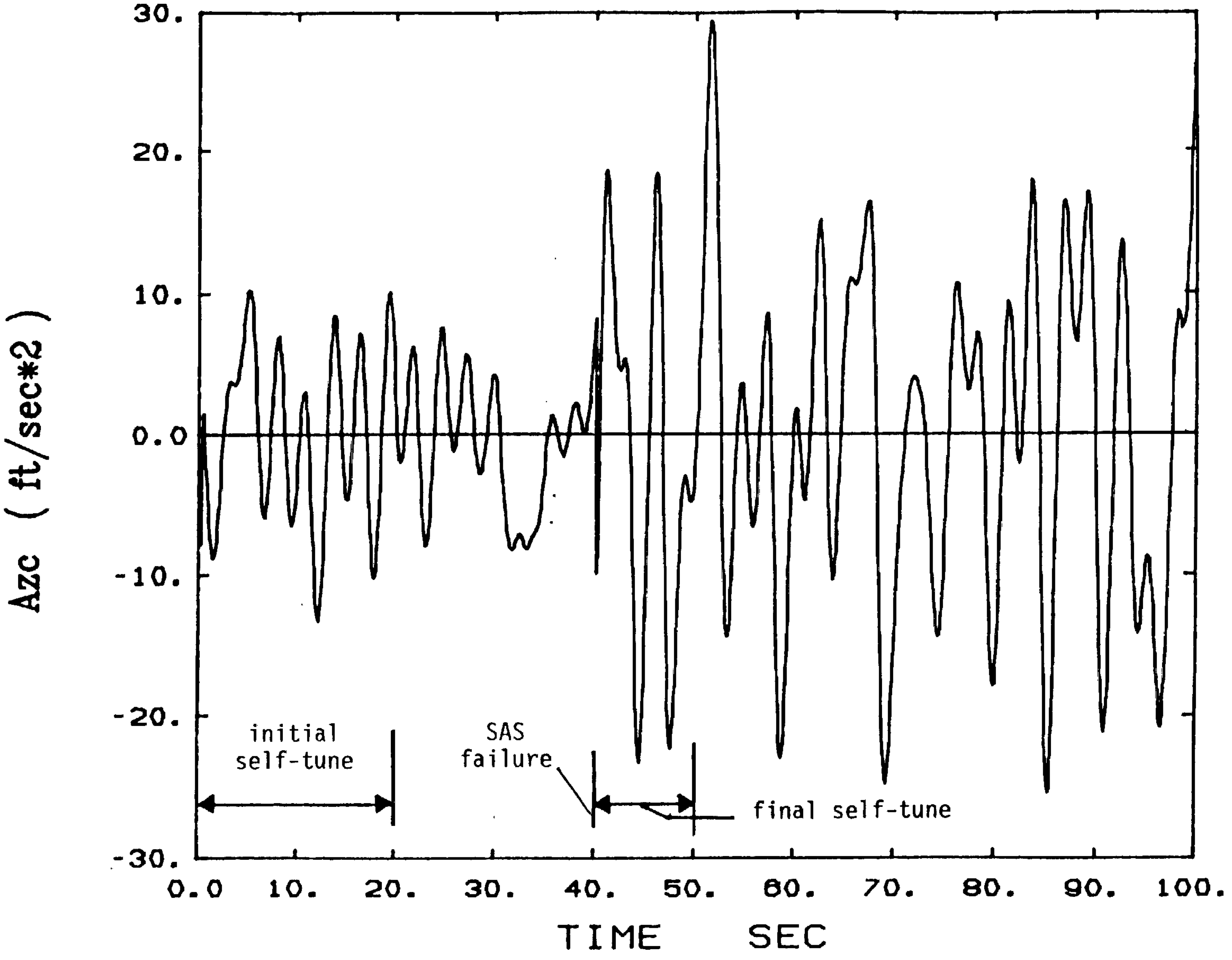
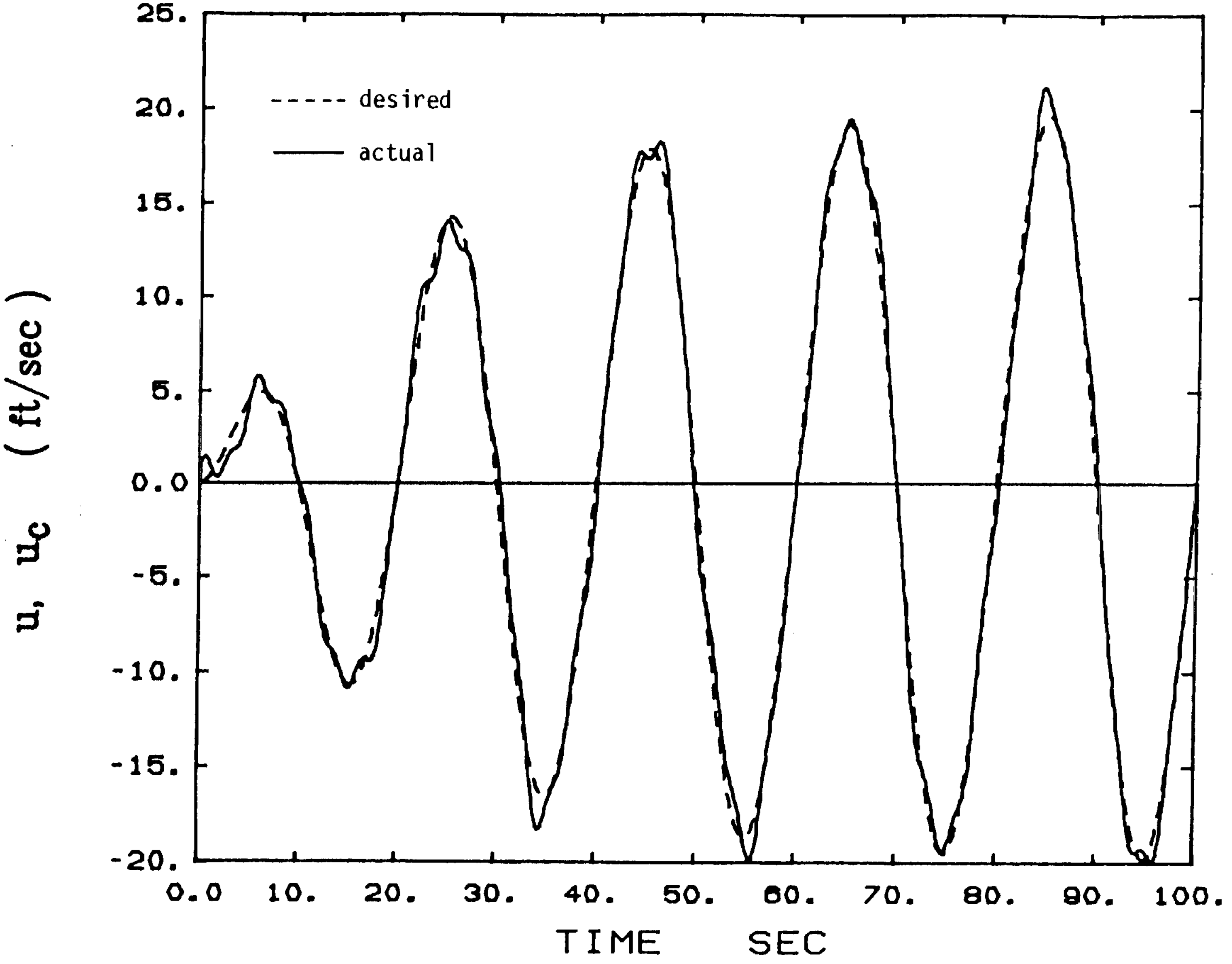

30.
20.
10.
0.0
-10.
-20.
-30.
Azc (ft/sec*2)
0.0
10.
20.
30.
40.
50.
60.
70.
80.
90.
100.
TIME SEC
initial
self-tune
SAS
failure
final self-tune

Fig. 5



----- desired
—— actual
25.
20.
15.
10.
5.
0.0
-5.
-10.
-15.
-20.
0.0
10.
20.
30.
40.
50.
60.
70.
80.
90.
100.
u, u_c (ft/sec)
TIME SEC

Fig. 6

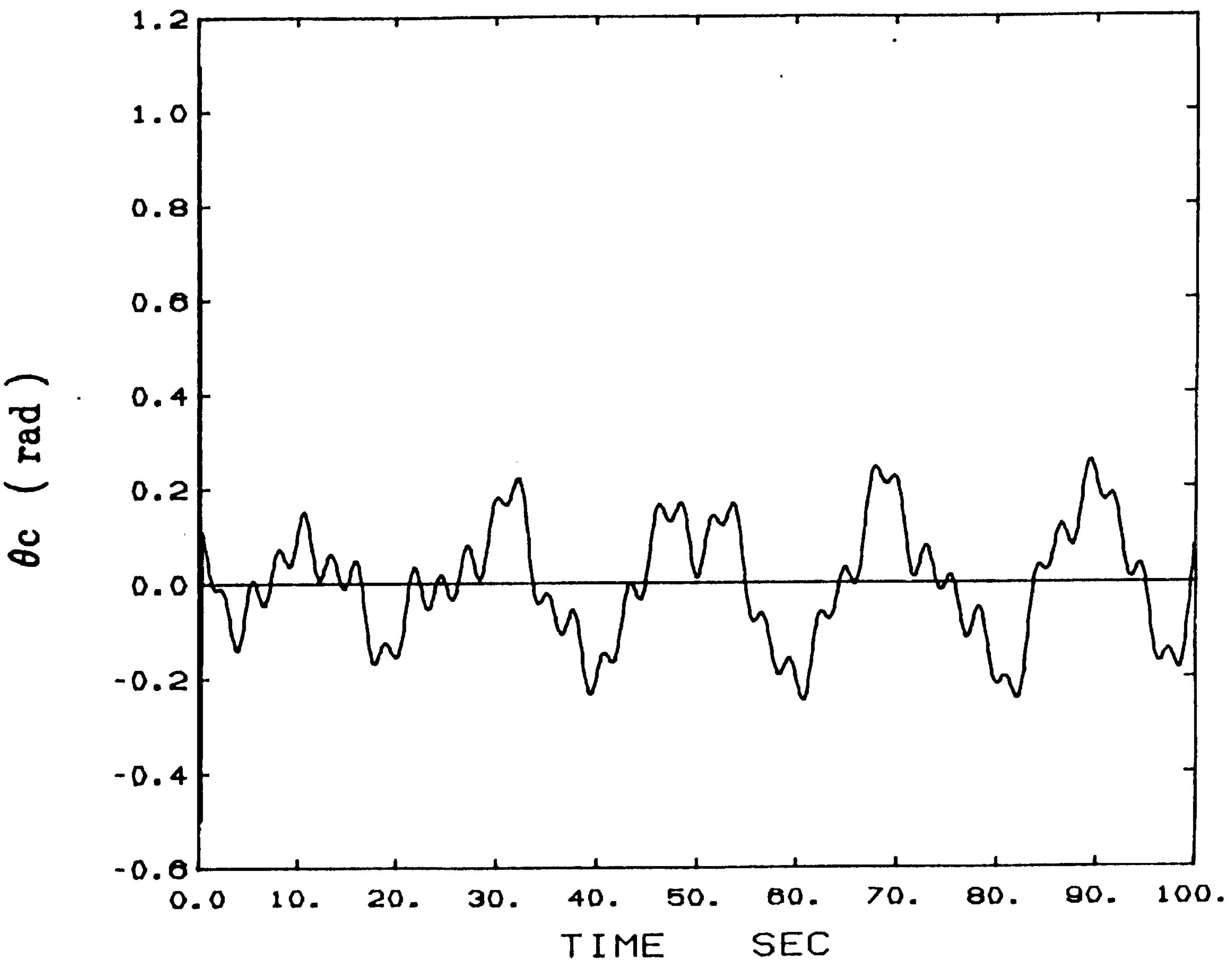
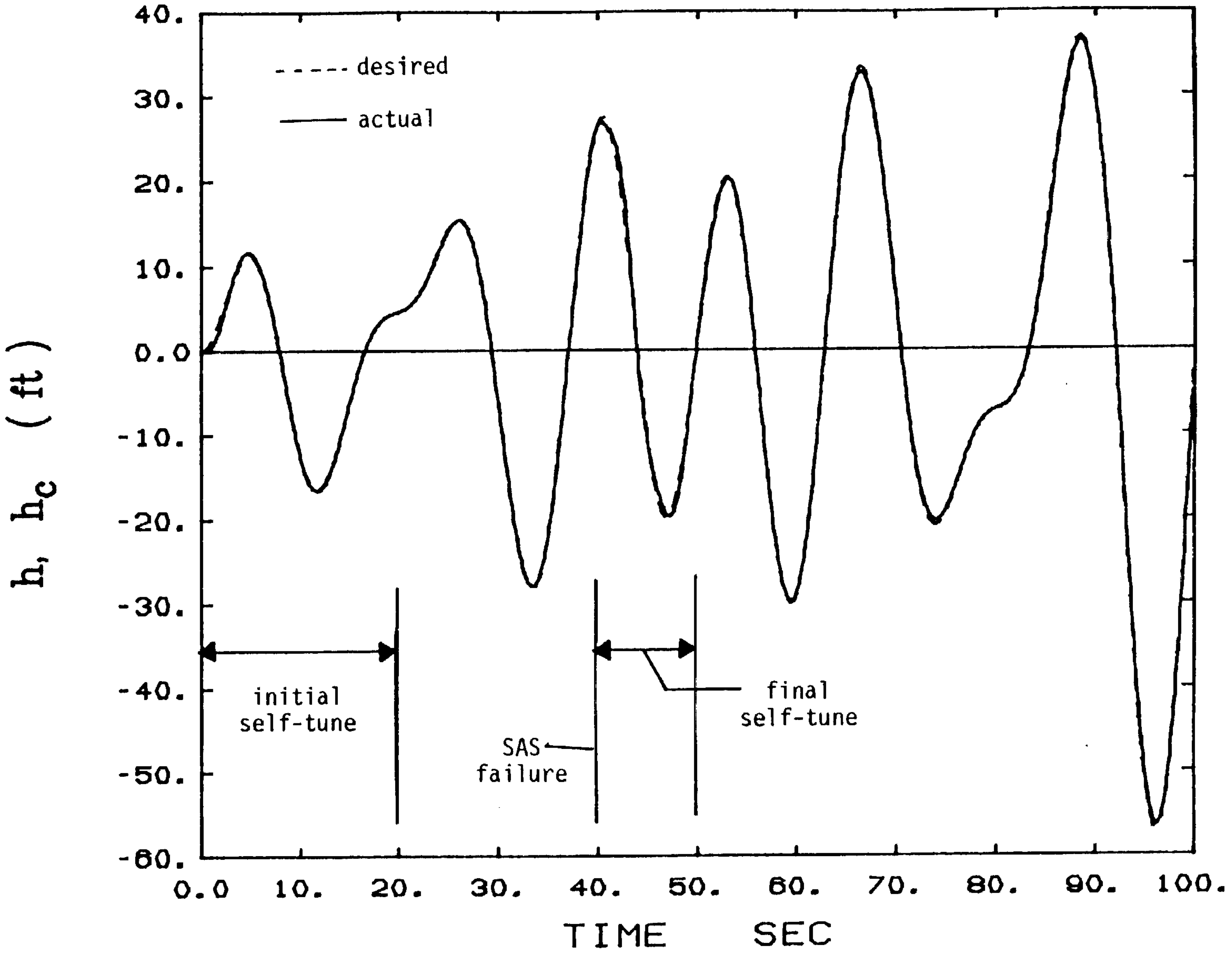

1.2
1.0
0.8
0.6
0.4
0.2
0.0
-0.2
-0.4
-0.6
θc (rad)
0.0
10.
20.
30.
40.
50.
60.
70.
80.
90.
100.
TIME SEC

Fig. 7



----- desired
—— actual
40.
30.
20.
10.
0.0
-10.
-20.
-30.
-40.
-50.
-60.
h, h_c (ft)
0.0
10.
20.
30.
40.
50.
60.
70.
80.
90.
100.
TIME SEC
initial
self-tune
SAS
failure
final
self-tune

Fig. 8

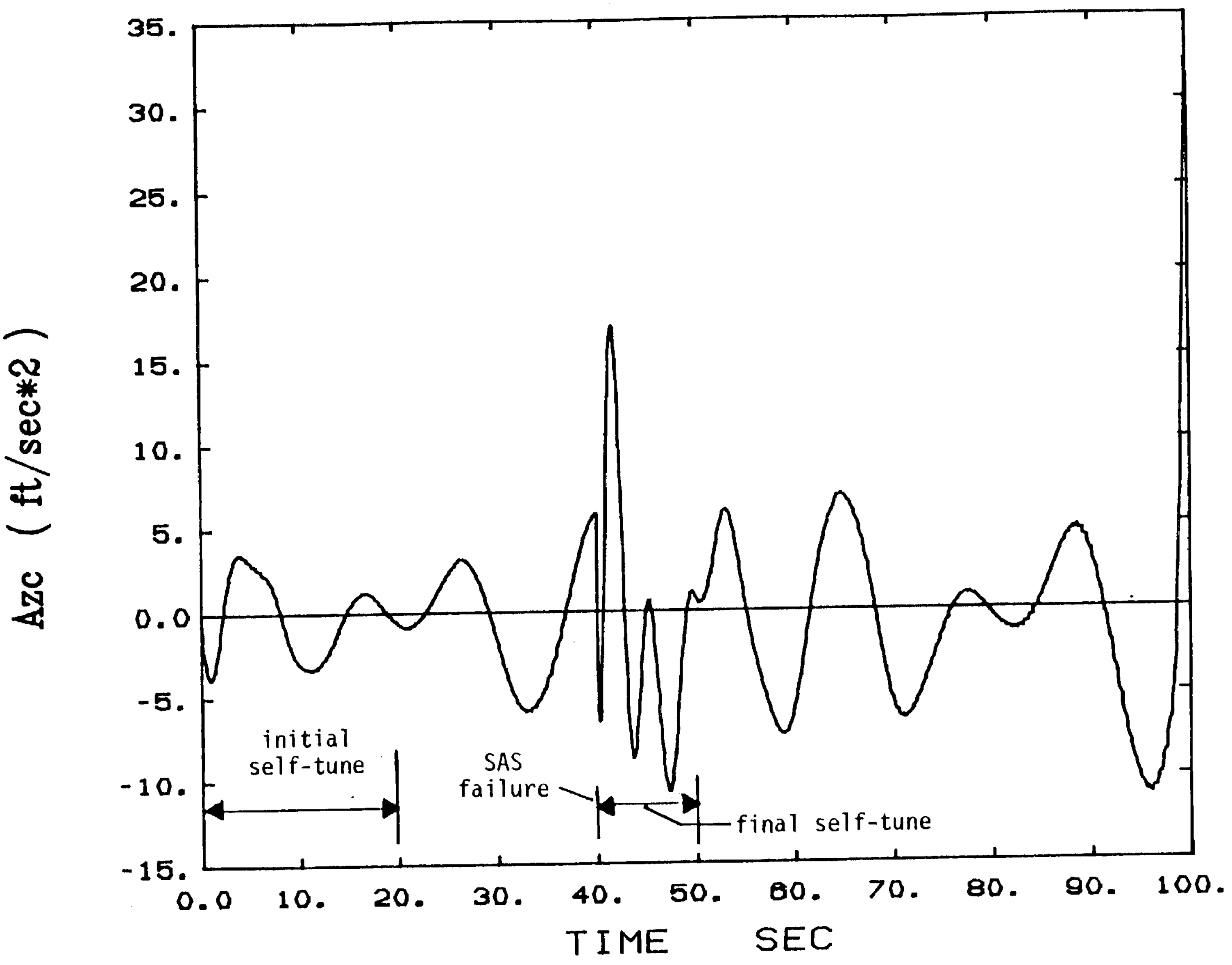

35.
30.
25.
20.
15.
10.
5.
0.0
-5.
-10.
-15.
Azc (ft/sec*2)
0.0
10.
20.
30.
40.
50.
60.
70.
80.
90.
100.
TIME SEC
initial
self-tune
SAS
failure
final self-tune

Fig. 9

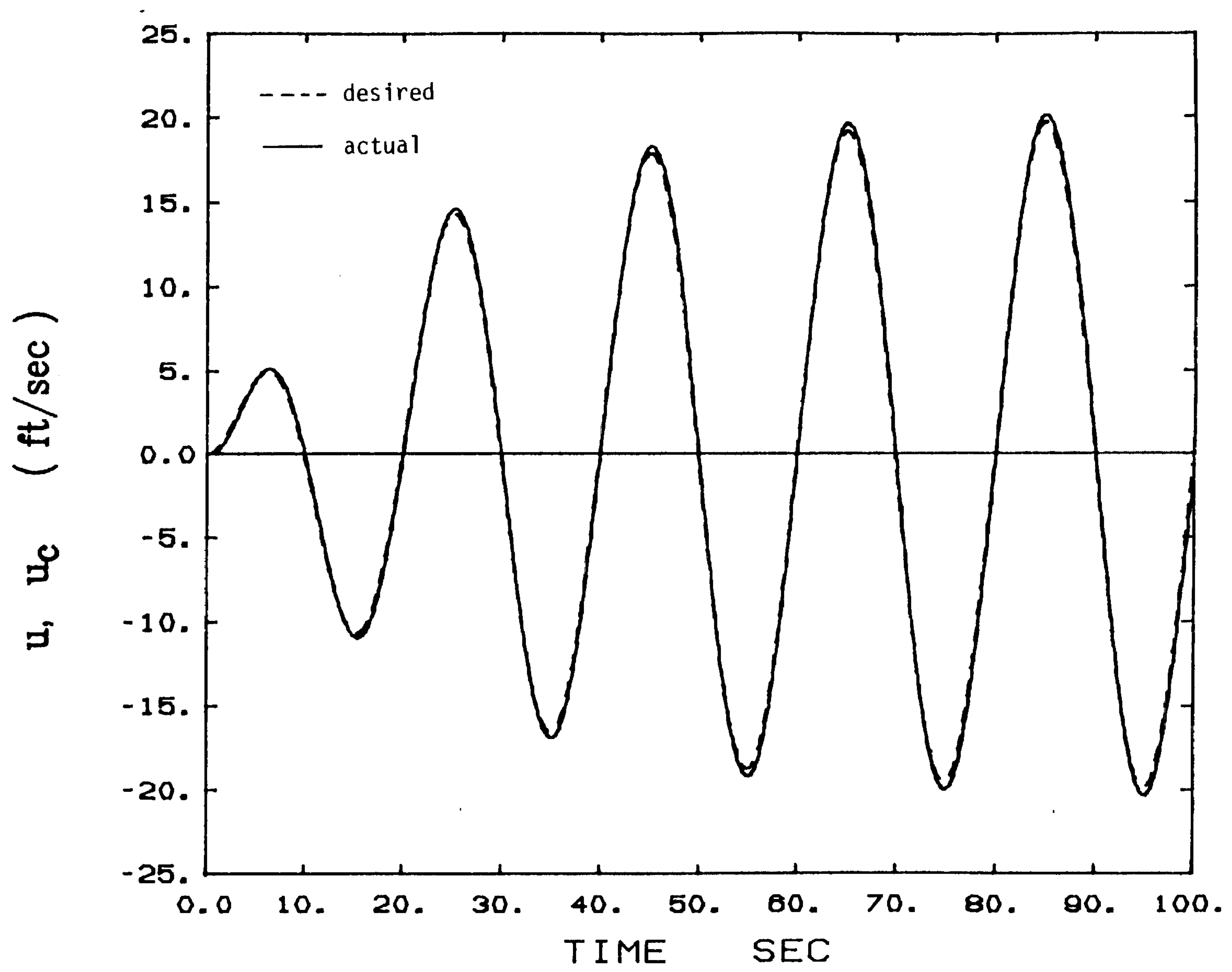

25.
20.
15.
10.
5.
0.0
-5.
-10.
-15.
-20.
-25.
---- desired
—— actual
u, u_c (ft/sec)
0.0
10.
20.
30.
40.
50.
60.
70.
80.
90.
100.
TIME SEC

Fig. 10

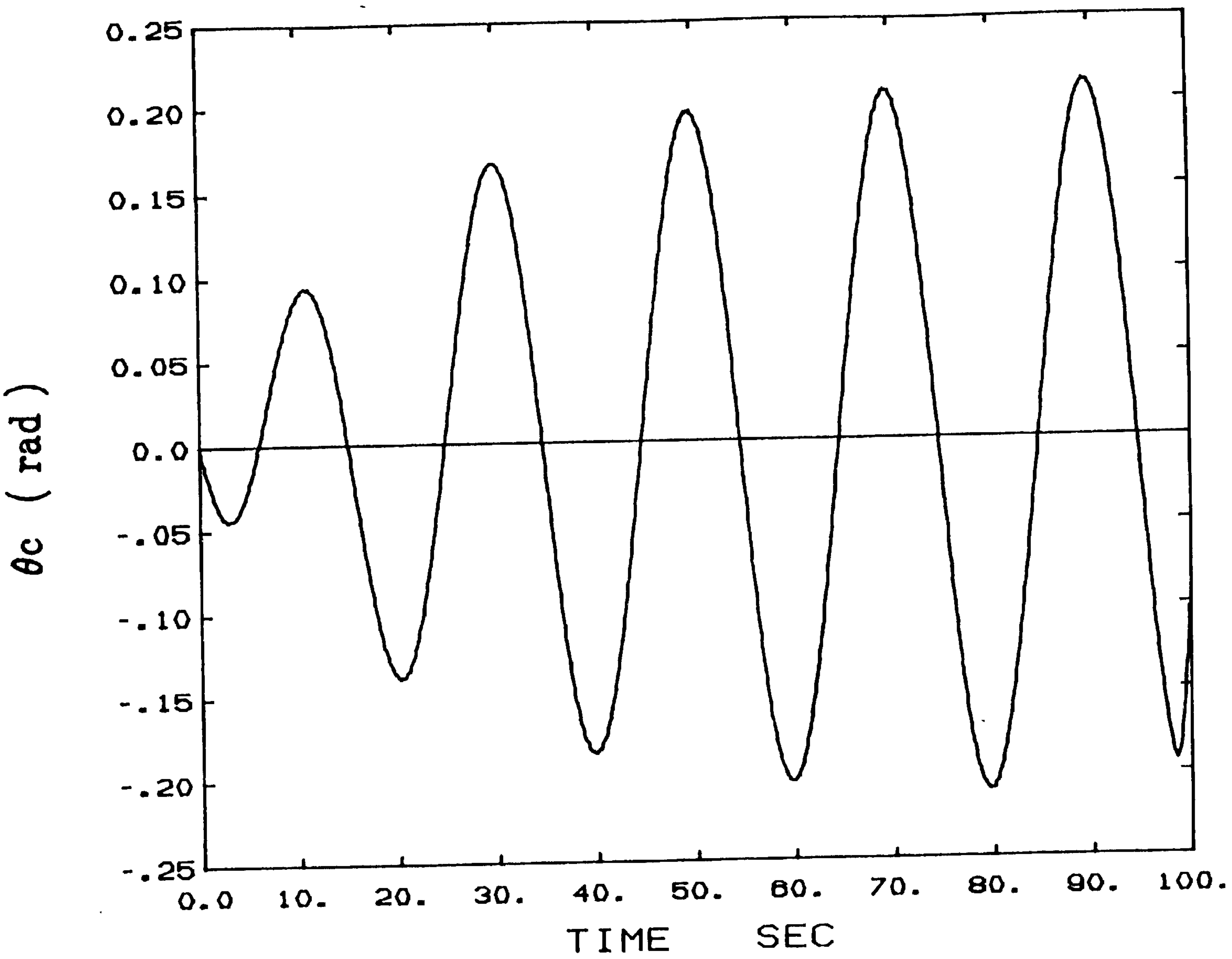

0.25
0.20
0.15
0.10
0.05
0.0
-.05
-.10
-.15
-.20
-.25
θc (rad)
0.0
10.
20.
30.
40.
50.
60.
70.
80.
90.
100.
TIME SEC

Fig. 11

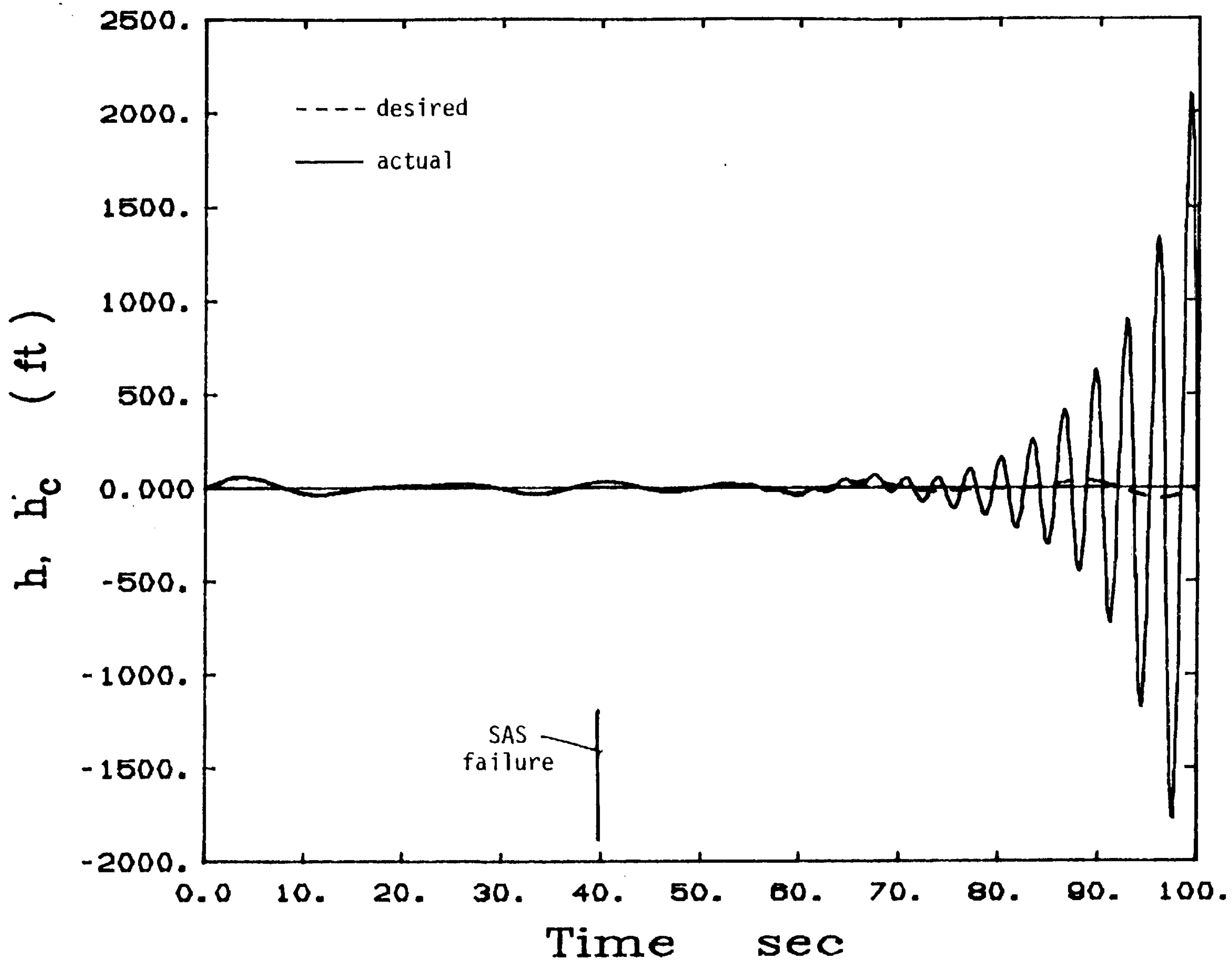

2500.
2000.
1500.
1000.
500.
0.000
-500.
-1000.
-1500.
-2000.
h, h_c (ft)
0.0
10.
20.
30.
40.
50.
60.
70.
80.
90.
100.
Time sec
---- desired
—— actual
SAS failure

Fig. 12

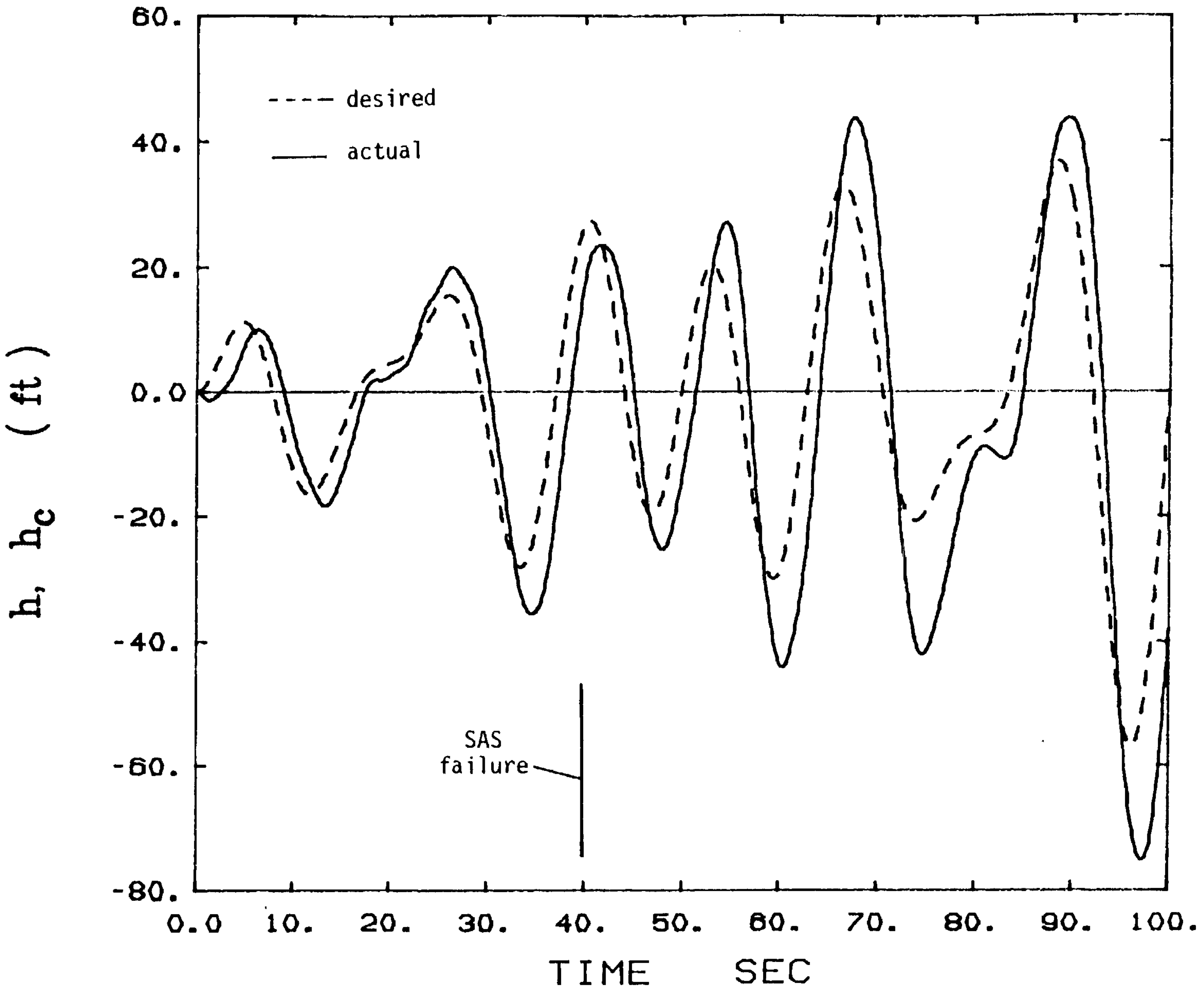

60.
40.
20.
0.0
-20.
-40.
-60.
-80.
h, h_c (ft)
---- desired
—— actual
SAS failure
0.0
10.
20.
30.
40.
50.
60.
70.
80.
90.
100.
TIME SEC

Fig. 13

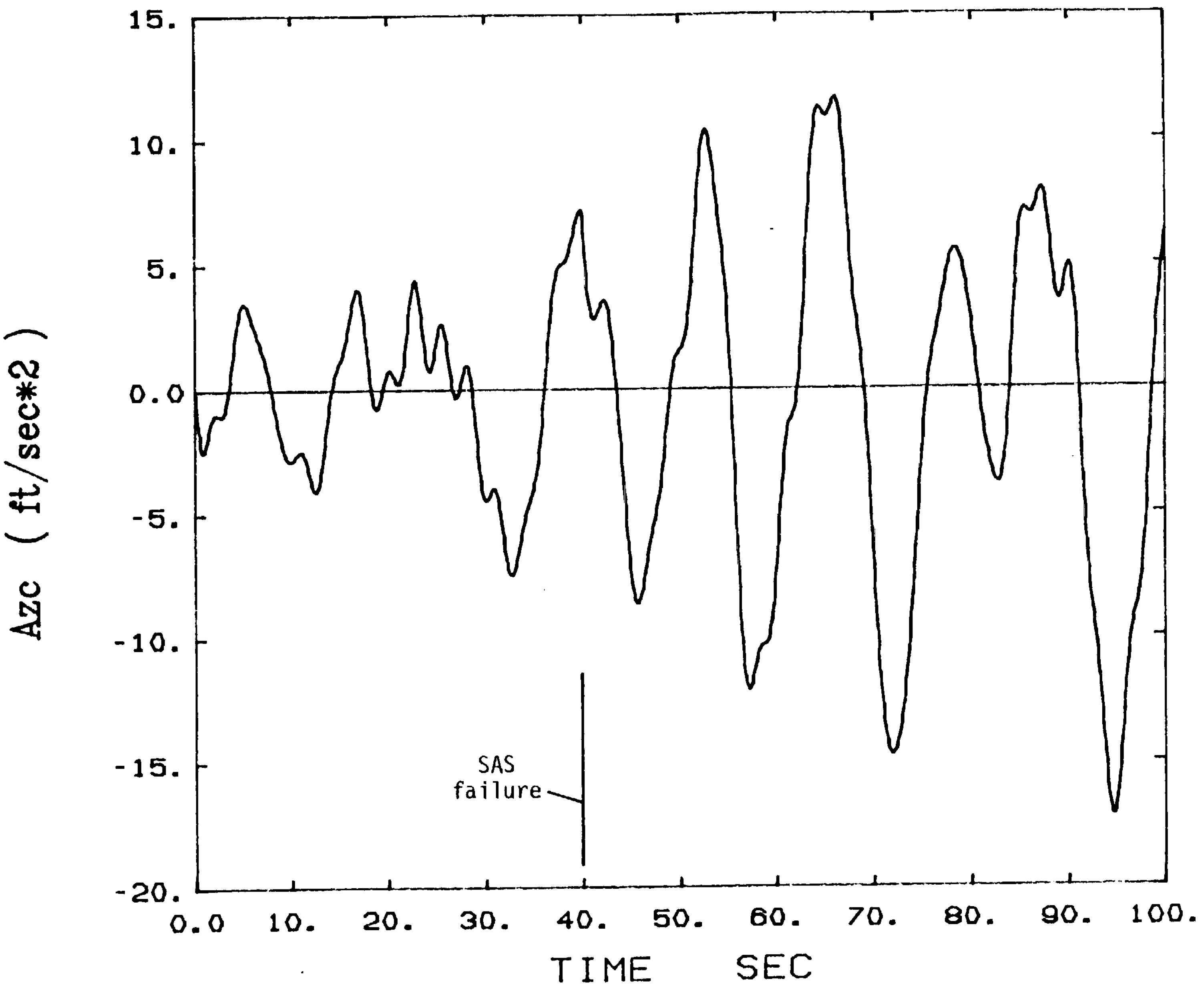
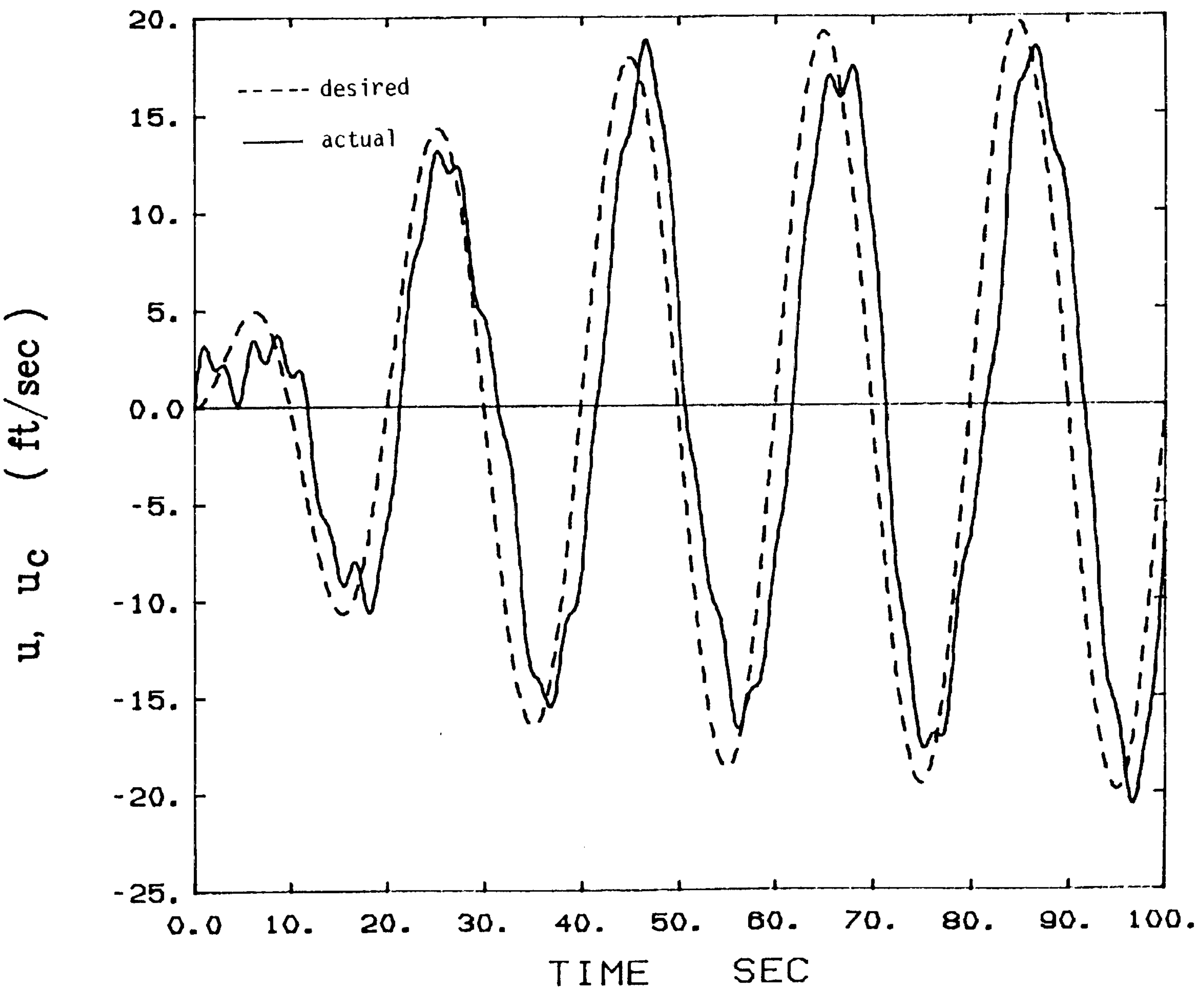

15.
10.
5.
0.0
-5.
-10.
-15.
-20.
Azc (ft/sec*2)
0.0
10.
20.
30.
40.
50.
60.
70.
80.
90.
100.
TIME SEC
SAS
failure

Fig. 14



----- desired
—— actual
20.
15.
10.
5.
0.0
-5.
-10.
-15.
-20.
-25.
0.0
10.
20.
30.
40.
50.
60.
70.
80.
90.
100.
TIME SEC
u, u_c (ft/sec)

Fig. 15

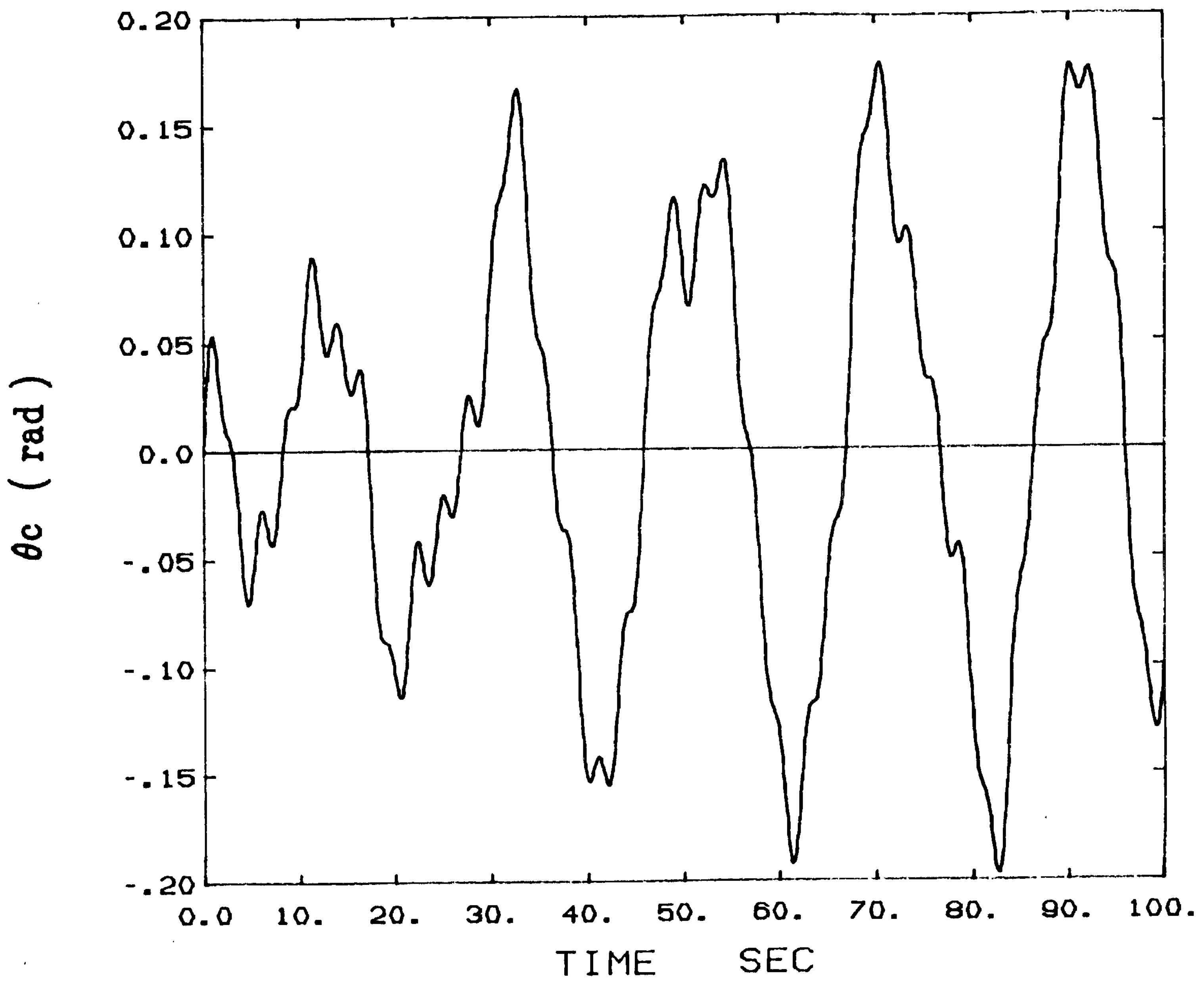

0.20
0.15
0.10
0.05
0.0
-.05
-.10
-.15
-.20
θc (rad)
0.0
10.
20.
30.
40.
50.
60.
70.
80.
90.
100.
TIME SEC

Fig. 16

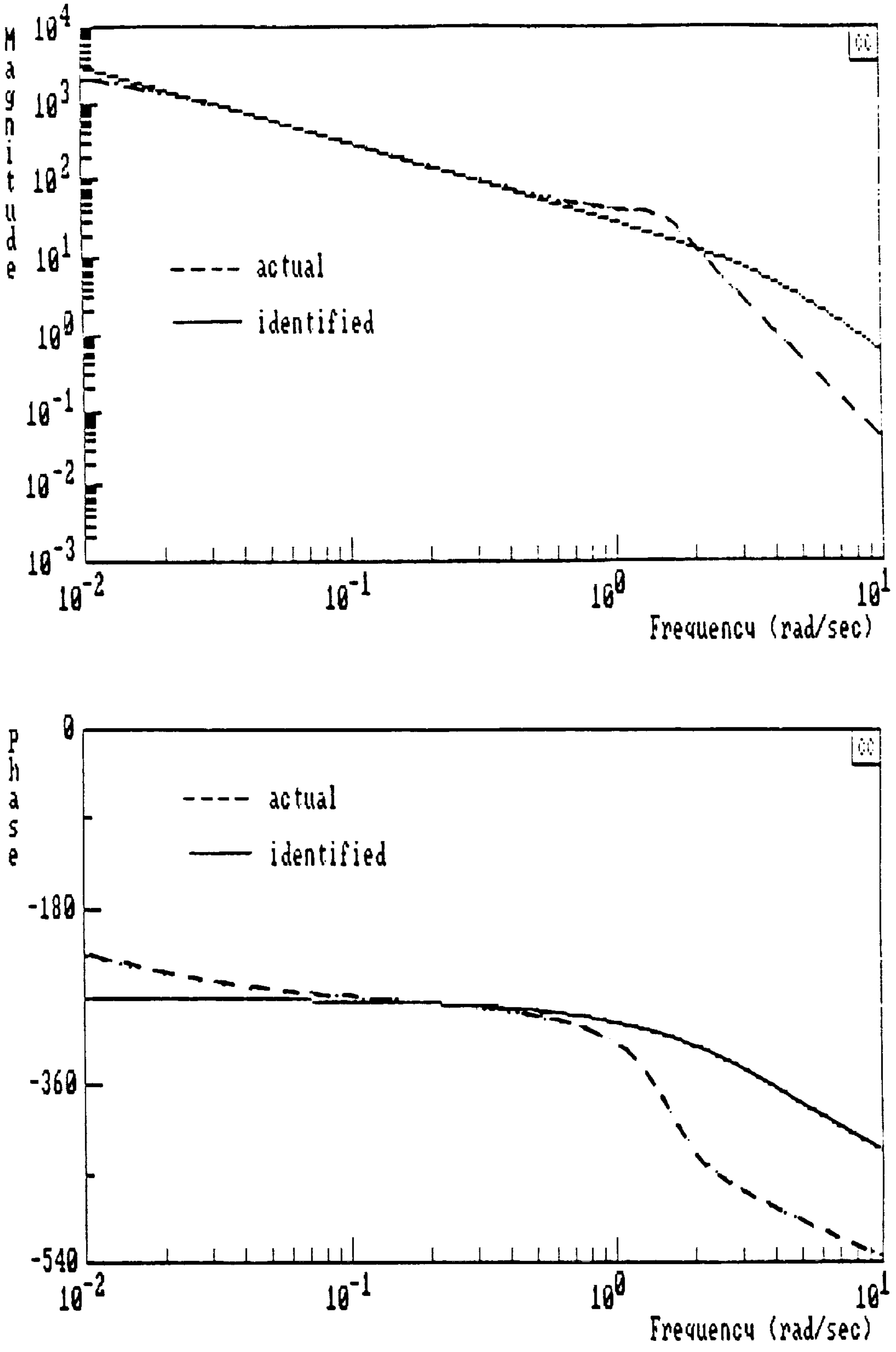

Magnitude
10^4
10^3
10^2
10^1
10^0
10^{-1}
10^{-2}
10^{-3}
actual
identified
10^{-2}
10^{-1}
10^0
10^1
Frequency (rad/sec)
Phase
0
-180
-360
-540
actual
identified
10^{-2}
10^{-1}
10^0
10^1
Frequency (rad/sec)

Fig. 17

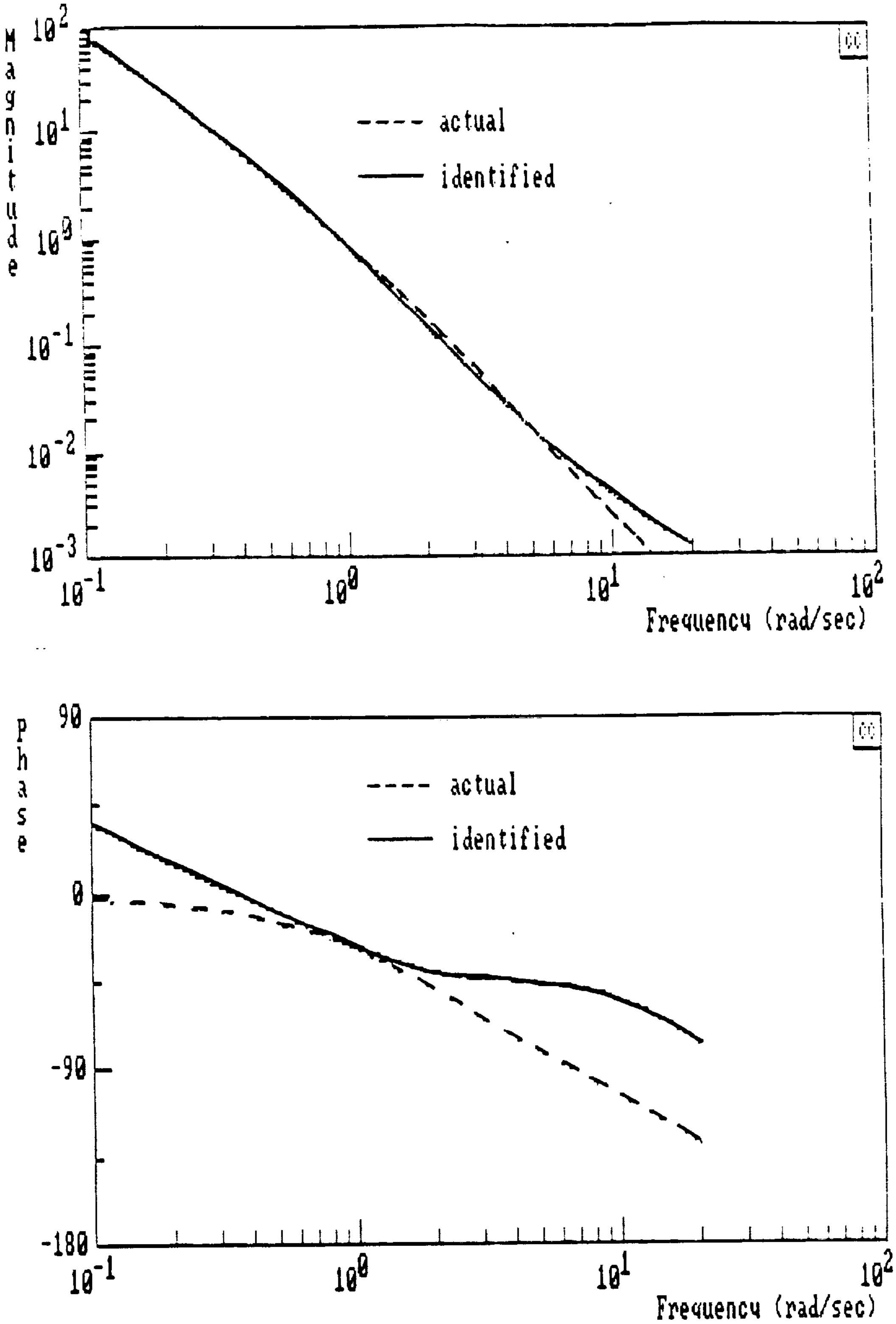

Magnitude
10^2
10^1
10^0
10^-1
10^-2
10^-3
10^-1
10^0
10^1
10^2
Frequency (rad/sec)
---- actual
—— identified
Phase
90
0
-90
-180
10^-1
10^0
10^1
10^2
Frequency (rad/sec)
---- actual
—— identified

Fig. 18

PRECISE FLIGHT PATH CONTROL USING A PREDICTIVE ALGORITHM

Y.C. Jung[1] and R. A. Hess[2]
Department of Mechanical, Aeronautical, and Materials Engineering
University of California, Davis CA 95616

Abstract

Generalized Predictive Control describes an algorithm for the control of dynamic systems in which a control input is generated which minimizes a quadratic cost function consisting of a weighted sum of errors between desired and predicted future system output and future predicted control increments. The output predictions are obtained from an internal model of the plant dynamics. A design technique is discussed for applying the single-input, single-output Generalized Predictive Control algorithm to a problem of longitudinal/vertical terrain-following flight of a rotorcraft. By using the Generalized Predictive Control technique to provide inputs to a classically designed stability and control augmentation system, it is demonstrated that a robust flight path control system can be created which exhibits excellent tracking performance.

Introduction

Over the past decade, a general technique has been introduced for the design of automatic controllers, called variously, Model Predictive Heuristic

[1] Graduate Student

[2] Professor, Associate Fellow, AIAA

Control, Model Algorithmic Control, Output Predictive Control, Dynamic Matrix Control, etc.[1-4] More recently, Clarke and Zhang[4] and Clarke, et al.,[5] have introduced Generalized Predictive Control (GPC) and have related it to the earlier approaches of Ref. 1-4 and Linear Quadratic (LQ) designs and have incorporated self-tuning in the control algorithm. Cast in terms of the flight path control problem which will be the subject of the research to be described the GPC algorithm, formulated as a discrete control problem, can be summarized as follows:[6]

1.) At each present time instant t_k, a prediction of the vehicle path j sampling periods into the future, is made. This prediction is obtained from a model of the vehicle dynamics.

2.) A control strategy for the next N_U sampling intervals is selected which brings the predicted vehicle path back to a desired path in the "best" way according to a specific control objective, i.e, LQ, or model following, etc.

3.) The resulting "best" control is then applied but only over the next sampling interval, and at the next sampling instant the procedure is repeated which results in a continuously updated control action with corrections based upon the latest measurements.

Hess and Jung have shown the potential performance of a GPC design in a rotorcraft longitudinal vertical flight path control problem using simplified vehicle dynamics.[7] The research to be described herein demonstrates how the single-input, single-output (SISO) GPC algorithm might be incorporated in a flight path control system to obtain a robust design with excellent performance.

The GPC Algorithm

Details of the GPC algorithm, itself, can be found in Ref. 5, however a brief review of the salient features of the approach will be undertaken in what follows:

The plant is modeled in discrete fashion using the so-called Controlled Auto-Regressive Integrated Moving Average (CARIMA) model:[5]

$$\begin{aligned} A(q^{-1})y(t) &= B(q^{-1})u(t-1) + \xi(t)/\Delta \\ A(q^{-1}) &= 1 + a_1 q^{-1} + \cdots + a_{na} q^{-na} \\ B(q^{-1}) &= b_0 + b_1 q^{-1} + \cdots + b_{nb} q^{-nb} . \end{aligned} \tag{1}$$

where q^{-1} is the delay operator and $y(t)$ and $u(t)$ are output and control variables, respectively, $\xi(t)$ is an uncorrelated random sequence, and Δ represents the differencing operator $(1 - q^{-1})$. The actual sampling interval is T, so that, at each sampling instant, the independent variable in Eq. 1 is kT. Now a prediction of the plant output, given measured output up to time kT and known control input $u(t + i)$ for $i \leqslant -1$, is

$$\hat{y}(t+j|t) = E_j(q^{-1})B(q^{-1})\Delta u(t+j-1) + F_j(q^{-1})y(t) \tag{2}$$

where

j = the number of future time steps being predicted

$E_j(q^{-1})$, $F_j(q^{-1})$ = result from a recursive solution of the Diophantine identity[8]

$$1 = E_j(q^{-1})A(q^{-1})\Delta + q^{-j}F_j(q^{-1}) \tag{3}$$

Also,

$$E_j(q^{-1})B(q^{-1}) = G_j(q^{-1}) + q^{-j}\Gamma_j(q^{-1}) \tag{4}$$

where

$$\begin{aligned} G_j(q^{-1}) &= g_0 + g_1 q^{-1} + \cdot\cdot\cdot + g_{j-1}q^{-j+1} \\ \Gamma_j(q^{-1}) &= \gamma_{j,o} + \gamma_{j,1}q^{-1} + \cdot\cdot\cdot + \gamma_{j,nb-1}q^{-nb+1} \end{aligned} \tag{5}$$

Substitution of Eq. 4 into Eq. 2 results in

$$\begin{aligned} \hat{y}(t+j|t) &= G_j(q^{-1})\Delta u(t+j-1) + \Gamma_j(q^{-1})\Delta u(t-1) + F_j(q^{-1})y(t) \\ &= G_j(q^{-1})\Delta u(t+j-1) + y^{OL}(t+j) \end{aligned} \tag{6}$$

with

$$y^{OL}(t+j) = \Gamma_j(q^{-1})\Delta u(t-1) + F_j(q^{-1})y(t) \tag{7}$$

Now a predictive control law can be defined as that which minimizes a cost function given by

$$J(N_1,N_2) = E\{\sum_{j=N_1}^{N_2}[y(t+j) - w(t+j)]^2 + \sum_{j=1}^{N_2}\lambda(j)[\Delta u(t+j-1)^2]\} \tag{8}$$

where

N_1 = the minimum costing horizon

N_2 = the maximum costing horizon

$w(k)$ = the desired value of the output y at the kth sampling instant

$\lambda(j)$ = a control weighting sequence

Equation 8 is concerned only with a subset of future time defined N_2T secs into the future and is dependent upon data up to time kT. As outlined in the Introduction, the control is generated in the following manner: At each sampling instant, an optimal control sequence for N_2 steps into the future is calculated, however only the first of these is applied to the plant. At the next sampling instant, a new optimal sequence is calculated which minimizes J for N_2 steps into the future, but again, only the first of these is applied to the plant. This defines a "receding horizon" strategy.

Significant reductions in the order of the matrices involved in computing the optimal control can be made by requiring that, after an interval $N_U < N_2$, projected control increments are assumed to be zero, i.e.,

$$\Delta u(t+j-1) = 0 \quad j > N_u \tag{9}$$

where N_U is called the "control horizon". This procedure is equivalent to placing infinite weights on control increments after a future time $N_U \cdot T$. With the introduction of the control horizon, the prediction equations become

$$\hat{\underline{y}} = \underline{G}_1 \tilde{u} + \underline{y}^{OL} \tag{10}$$

where

$$\underline{G}_1 = \begin{bmatrix} g_0 & 0 & 0 & & 0 \\ g_1 & g_0 & 0 & & 0 \\ . & . & . & \cdots & . \\ . & . & . & \cdots & . \\ g_{N-1} & g_{N-2} & g_{N-3} & \cdots & g_{N-N_u} \end{bmatrix}_{(N \times N_u)} \tag{11}$$

$$\hat{\underline{y}} = [\hat{y}(t+1), \hat{y}(t+2), \cdots, \hat{y}(t+N)]^T$$

$$\tilde{\underline{u}} = [\Delta u(t),\ \Delta u(t+1), \cdots, \Delta u(t+N_u-1)]^T$$

$$\underline{y}^{OL} = [y^{OL}(t+1), y^{OL}(t+2), \cdots, y^{OL}(t+N)]^T$$

For simplicity of notation, it was assumed in Eq. 11 that $N_1 = 1$ and N_2 was referred to simply as N. The corresponding control law is given by:

$$\tilde{\underline{u}} = (\underline{G}_1^T \underline{G}_1 + \lambda \underline{I})^{-1} \underline{G}_1^T (\underline{w} - \underline{y}^{OL}) \tag{12}$$

and

$$\Delta u(t) = \underline{k}^T(\underline{w} - \underline{y}^{OL})$$

$$\underline{k}^T = [1,0,0,\cdots,0](\underline{G}_1^T\underline{G}_1 + \lambda\underline{I})^{-1}\underline{G}_1^T \tag{13}$$

The current control law, u(k) is thus

$$u(t) - u(t-1) + \underline{k}^T(\underline{w} - \underline{y}^{OL}) \tag{14}$$

The incremental controller ensures zero offset even with non-zero disturbances, and, as such, is equivalent to integral control.

The final products of the entire GPC design are contained in the coefficients of the $\Gamma_j(q^{-1})$ and $G_j(q^{-1})$ polynomials of Eqs. 5 and in the elements of **k** matrix of Eq. 14, all of which can be precomputed.

The choice of parameters N_1, N_2, N_U, T and λ , determines the performance and stability of the GPC algorithm. It will be demonstrated how guidelines for selecting these parameters[9], can be incorporated into an overall design procedure. As regards stability, the following theorem is of interest[9]:

Given: A state-space model of the plant of Eq. 1, augmented by an integrator:

$$\underline{x}(t+1) = \underline{A}\underline{x}(t) + \underline{b}\Delta u(t)$$

$$y(t) = \underline{c}^T\underline{x}(t) \tag{15}$$

The state-space model of Eq. 15 is stable under GPC control if:

1.) The n-state model (A,b,c) is stabilizable and detectable, and if

2.) $NU = N_1 \geq n$, $N_2 - N_1 \geq n-1$, and $\lambda = \varepsilon \to 0$.

A proof of this theorem is offered in Ref. 9. As will be seen, the conditions (1-2) will, in general, be met by the proposed application of the GPC algorithm.

Analysis of SISO GPC Designs

Substituting Eq. 7 into the first of Eqs. 13 gives the following:

$$\Delta u(t) = \sum_{i=N_1}^{N_2} k_i q^i w(t) - \sum_{i=N_1}^{N_2} k_i \Gamma_i(q^{-1}) \Delta u(t-1) - \sum_{i=N_1}^{N_2} k_i F_i(q^{-1}) y(t) \qquad (16)$$

or

$$[1 + q^{-1} \sum_{N_1}^{N_2} k_i \Gamma_i(q^{-1})] \Delta u(t) = \sum_{N_1}^{N_2} k_i q^i w(t) - \sum_{N_1}^{N_2} k_i F_i(q^{-1}) y(t) \qquad (17)$$

Equation 17 can be represented in the form of the block diagram shown Fig. 1.[10] The closed-loop transfer function can be obtained directly as:

$$\frac{y}{w}(q^{-1}) = \frac{\dfrac{\sum k_i q^i q^{-1} B(q^{-1})}{(1-q^{-1})[1+q^{-1}\sum k_i \Gamma_i] A(q^{-1})}}{1 + \dfrac{q^{-1} B(q^{-1}) \sum k_i F_i(q^{-1})}{(1-q^{-1})[1+q^{-1}\sum k_i \Gamma_i] A(q^{-1})}} \qquad (18)$$

with the loop transmission given by

$$L(q^{-1}) = \frac{q^{-1}B(q^{-1})\sum k_i F_i(q^{-1})}{(1-q^{-1})[1+q^{-1}\sum k_i \Gamma_i]A(q^{-1})} \qquad (19)$$

By transforming to the w' plane, the Bode plot of Eq. 19 allows the phase and gain margins of the SISO GPC controlled system to be examined just as with a conventional, non-predictive SISO design.

A Design Procedure

Figure 2 shows how the SISO GPC algorithm could be incorporated into a Multi-Input, Multi-Output (MIMO) flight control system. Here, is it assumed that mission/task requirements demand very precise tracking performance for one of the output variables, here shown as y. An example would be vertical flight path deviations in a terrain-following flight task. It is this hypothesized stringent performance requirement which justifies the use of the GPC algorithm as part of the flight control system.

The proposed design procedure would require, in most applications, that the vehicle possess a stability augmentation system (SAS). We would include in the definition of SAS here, control of other pertinent output variables not subject to GPC control. From the standpoint of the GPC design, the purpose of this SAS is to: 1) provide stabilizable and detectable dynamics, i.e. to ensure that modes which may not be controllable or observable, are at least asymptotically stable, 2) reduce the variations in the dynamics of the "effective vehicle" over the flight regime in which the GPC design is to be used, 3) reduce the effects of any nonlinearities in the vehicle dynamics, and 4) simplify the vehicle dynamics, i.e., reduce the apparent order of the transfer function for the effective vehicle which is used in determining the

GPC control law. Thus, in the overall design, the SAS will provide robustness, and the GPC will provide performance. Since most high performance flight vehicles now include a full-authority SAS for acceptable handling qualities, this approach appears quite reasonable. As will be seen, a properly designed SAS can allow a single, fixed-parameter GPC design to control a vehicle over a flight regime in which the unaugmented dynamics are subject to considerable variation.

The design procedure can be summarized as follows:

1.) Create an "effective vehicle" which possesses the desirable dynamic characteristics just outlined. This effective vehicle will typically be obtained using linear feedback principles associated with any acceptable feedback design technique, e.g. H^{∞}[11], or QFT[12]. Note that system <u>performance</u> is not the object of this design.

2.) Referring to Fig. 2, form the $y(s)/u_{GPC}(s)$ transfer function, with the feedback loops obtained in step (1), closed.

3.) If possible, approximate the $y(s)/u_{GPC}(s)$ of step 2, with a lower-order transfer function. Discretize this transfer function, including a zero-order hold. The discretization interval, T, is here assumed to be dictated by constraints other than the control system design, e.g., minimum cycle time of the digital computer implementing the GPC law.

4.) Select the initial GPC parameters as follows:[9]

$N_1 = n$, the order of the discretized transfer function from step (3), plus one, to account for the integral action of the GPC design,

$N_2 = 2n - 1$

$N_U = n$

$\lambda = \varepsilon$, a value large enough to ensure invertibility in the matrix $G_1^T G_1 + \lambda I$ in Eq. 12. Thus, the weighting sequence of Eq. 8 is a constant value, here.

5.) With these selections, and using unit step responses, adjust λ by trial and error, to ensure stability, desirable transient performance and adequate gain and phase margins. Of course, this trial and error is equivalent to a pole-placement procedure and the problem can be approached as such. For this step use the simplified transfer function of step (3).

6.) Evaluate the GPC law obtained in step (5), again using unit step responses, however, now using the complete vehicle/SAS model. If the GPC design to this point is acceptable, simulate using more realistic command inputs over the entire flight regime of interest. If the design is not acceptable, repeat steps 3-6, with a more accurate simplified transfer function. It may be necessary to modify the values of N_2, and N_U, so that $N_2 > 2n-1$, $N_U > n$.[9] Finally, of course, the flight regime may involve changes in vehicle characteristics of such magnitude that GPC control law scheduling may be necessary.

A Design Example

The Task and Vehicle Model

The example to be presented involves a rotorcraft terrain-following task. This task is also often referred to as "contour flight" and is characterized by low altitude flight conforming generally to the contours of the terrain and gross vegetation features. Each leg of contour flight is characterized by constant heading, varying airspeed and flight path as close to the earth's surface as vegetation, obstacles, and ambient light will permit.[13] The rotorcraft model for this study is a rigid-body model of the BO-105C.[14] To provide a challenging task, the command airspeed ranged from 20 kts to 100 kts while the vehicle was to follow a vertical flight path emulating a pre-computed profile described by a sum of three sinusoids:

$$h_c(t) = w(t) = 20[\sin(.05(2\pi t)) + \sin(.06(2\pi t)) + \sin(.08(2\pi t))] \ \text{ft} \qquad (20)$$

The appendix describes the vehicle model. Basically, the dynamics were obtained through linear interpolation between five equilibrium flight conditions at 20, 40, 60, 80 and 100 kts. The interpolation was based upon low-pass filtered vehicle airspeed. The command airspeed profile consisted of a series of constant accelerations of 3.375 ft/sec^2 for 10 secs each, followed by 10 secs of constant velocity. The constant velocity sections occurred at 20, 40, 60 80, and 100 kts, respectively. The unaugmented vehicle dynamics were linear, but time-varying, highly coupled, and, depending upon the airspeed, unstable and/or non-minimum phase in nature. The coupling referred to here is that between attitude and vertical velocity.

The Design

1.) Figure 3 shows the stability augmentation system (SAS) which was designed to meet the aforementioned criteria. We include airspeed control as part of this SAS design. The design was a "classical" frequency domain approach involving successive loop closures, beginning with pitch attitude, then altitude rate, and finally airspeed. The airspeed loop possessed a bandwidth of approximately 0.4 rad/sec. The design was based upon the vehicle dynamics at 60 kts. Figures 4-5 show the resulting closed-loop transfer function for altitude rate ($\dot{h}/\dot{h}_c'$) and airspeed (u/u_c). The latter transfer function was calculated with the altitude-rate loop closed. The prime notation on h_c' serves to indicate that $h_c' \neq d(h_c)dt$. This feedback system possessed sufficient robustness to be employed for the entire flight regime studied here. Given this fact, the GPC design was also based upon the 60 kt vehicle/SAS dynamics (the effective vehicle).

2,3.) Figure 6 compares the actual and reduced-order h/u_{GPC} transfer functions for the vehicle plus SAS, i.e. all the feedback loops in Fig. 3 were closed in computing h/u_{GPC}. As can be seen, the reduced order transfer function compares quite favorably with that of the actual vehicle. The reduced-order function is of order 2, while that of the actual vehicle/SAS dynamics are of order 8. The dynamics of both the actual and simplified vehicle are controllable and observable, and hence meet the conditions of the stability theorem stated previously. The reduced-order transfer function is given by:

$$\frac{h}{u_{GPC}}(s) = \frac{0.8446}{s(s/4.72+1)} \qquad (21)$$

Note that, in terms of the GPC design, n = 3, including the additional order arising from the inherent integral action of the GPC design. The discretization interval here was selected as T = 0.1 secs, and was not considered a design parameter. With this interval and including the effects of a zero-order hold, the dynamics of Eq. 21 become,

$$\frac{h}{u_{GPC}}(q^{-1}) = \frac{q^{-1}(0.01713+0.14644q^{-1})}{1-1.6238q^{-1}+0.62375} \qquad (22)$$

4,5) The initial GPC parameters are $N_1 = n = 3$, $N_2 = 2n - 1 = 5$, $N_U = n = 3$, $\lambda = 0$. Figure 7 shows the locus of w'-plane closed-loop characteristics roots for different λ values. Based upon Fig. 7 and corresponding step responses, λ was selected as $\lambda = 0.039$. Figure 8 shows the w'-plane Bode plot for the loop transmission given by Eq. 19. The gain and phase margins are seen to be 38 deg, and 11 dB, respectively, which were judged acceptable for this design. The maximum costing horizon of 0.5 secs is quite modest. An examination of the effective vehicle dynamics for the 20 kt and 100 kt conditions revealed that the transfer function of Eq. 21 still provided an excellent approximation.

6.) Figure 9 shows the step responses for h and u for the actual system, wherein the airspeed command has been set to 60 kts (no change). As can be seen, the transient responses are well-damped.

Simulation

Figures 10-13 show the altitude, airspeed, collective and longitudinal cyclic time histories which result when the rotorcraft is commanded to follow the altitude trajectory of Eq. 20 with the aforementioned airspeed command profile. The units on the collective and longitudinal cyclic refer to equivalent control displacement at the pilot's hand. For convenience in plotting, the control inputs were assumed to be zero when positioned at the 20 kt trim values (see Appendix). The flight path tracking performance is such that it is difficult to distinguish the command from the output time histories in Fig. 10, thus, the dashed curve in Fig. 14 shows the altitude errors. With the exception of the transients at the beginning and end of the run, the maximum altitude errors are seen to be less than 1.5 ft in magnitude. The increasing amplitude of the longitudinal cyclic input in Fig. 13 reflects the monotonically increasing airspeed.

It is interesting to compare the performance of the GPC system with a more conventional design. To this end, the system of Fig. 15 was simulated. This control system is identical to the SAS design of Fig. 3, with the addition of an altitude loop, with equalization G_h, and a prefilter G_{hf}. As can be seen from the figure, the prefilter essentially provides a low frequency lead command to improve altitude tracking performance. In the absence of the prefilter, the altitude loop possessed a bandwidth of approximately 0.6 rad/sec. With the prefilter, the bandwidth exceeded 10 rad/sec. Of course, the tracking improvements which result from the prefilter are completely dependent upon the the availability of precise commanded altitude rate information. Noise or other errors in this signal will significantly compromise the performance of this conventional design. The solid curve in Fig. 14 shows the altitude errors for this conventional design. As can be seen, the maximum altitude errors are on the order of 5 ft in magnitude, considerably larger than those for the GPC design.

Conclusions

Based upon the research described herein, the following conclusions can be drawn:

1.) The single-input, single-output Generalized Predictive Control algorithm can be applied to precise flight path control as part of a multiloop flight control system.

2.) A design procedure can be offered in which a stability augmentation system creates a simplified effective vehicle which is then subjected to GPC control. For the purposes of design, the effective vehicle is approximated by lower-order, linear, time-invariant dynamics. The lower-order nature of the effective vehicle (as compared to the unaugmented vehicle) permits the implementation of a simple GPC controller.

3.) Given the order of the effective vehicle dynamics, all but one of the parameters which determine the GPC controller can be selected. The final GPC parameter, the control increment weighting coefficient, λ , is selected on a trail and error basis using the system step response with the simplified dynamics, or, equivalently, via pole placement.

4.) The simulation of a rotorcraft with highly coupled, time varying, unstable and/or nonminimum phase dynamics in a longitudinal terrain avoidance task demonstrated the potential of the GPC algorithm to provide excellent flight path tracking performance with adequate stability margins.

Acknowledgement

This research was supported by Grant No. NAG 2-221 from the Aircraft Guidance and Navigation Branch and the Flight Dynamics and Controls Branch of NASA Ames Research Center, Moffett Field, CA.

References

[1]Richalet, J., Rault, A., Testud, J. L., Papon, J., "Model Predictive Heuristic Control: Applications to Industrial Processes," Automatica, Vol. 14, No. 5, Sept. 1978, pp. 413-428.

[2]Rouhani, R., and Mehra, R. K., "Model Algorithmic Control (MAC): Basic Theoretical Properties," Automatica, Vol. 18, No., 4, July 1982, pp. 401-414.

[3]Reid, J. G., Chaffin, D. E., and Silverthorn, J. T., "Output Predictive Algorithmic Control: Precision Tracking with Application to Terrain Following," Journal of Guidance and Control, Vol. 4, No. 5, Sept-Oct., 1981, pp. 502-509.

4Clarke, D. W., and Zhang, L., "Long-Rage Predictive Control Using Weighting Sequence Models," IEE Proceedings, Vol. 134, Pt. D., No. 3, May 1987, pp. 187-195.

[5]Clarke, D. W., Mohtadi, C., and Tuffs, P. S., "Generalized Predictive Control, Parts I and II," Automatica, Vol., 23, No. 2, March 1987, pp. 137-160.

[6] De Keyser, R. M. C., Van De Velde, G. A., Dumortier, F. A. G., "A Comparative Study of Self-Adaptive Long-Range Predictive Control Methods," Automatica, Vol. 24, No. 2, March 1988, pp. 149-163.

[7] Hess, R. A., and Jung, Y. C., "An Application of Generalized Predictive Control to Rotorcraft Terrain-Following Flight," IEEE Transactions on Systems, Man, and Cybernetics, Vol. 19 No. 5, Sept/Oct. 1989, pp. 955-962.

[8] Astrom, K. J., and Wittenmark, B., Adaptive Control, Addison-Wesley, 1989.

[9] Clarke, D. W., and Mohtadi, C., "Properties of Generalized Predictive Control," Automatica, Vol. 25, No. 6, Nov. 1989, pp. 859-875.

[10] Crisalle, O. D., Seborg, D. E., and Mellichamp, D. A., "Theoretical Analysis of Long-Range Predictive Controllers," Proceedings of the 1989 American Control Conference, pp. 570-576.

[11] Francis, B. A., Helton, J. W., and Zames, G., H^{∞}-Optimal Feedback Controllers for Linear Multi-Variable Systems," IEEE Transactions on Automatic Control, Vol. AC-29, No. 10, 1984, pp. 888-900.

[12] Horowitz, I., "Quantitative Feedback Theory, (QFT)," Proceedings of the 1988 American Control Conference, June 1988, pp. 2032-2037.

[13] Cheng. V. H. L., and Sridhar, B., "Considerations for Automated Nap-of-the-Earth Rotorcraft Flight," *Proceedings of the 1988 American Control Conference*, June 15-17, 1988, pp. 967-976.

[14] Heffley, R. K., et al, "A Compilation and Analysis of Helicopter Handling Qualities Data, Vol. 1: Data Compilation," NASA CR-3144, Aug. 1979.

BO-105C

Longitudinal Linearized Equations of Motion

Body Axes

Airspeeds 20 kts-100 kts

$$\begin{aligned}
\dot{u} &= (X_u)u + (X_w)w + [-W_o + X_q]q - (g\cos\theta_o)\theta + X_{\delta c}\delta_c + X_{\delta B}\delta_B \\
\dot{w} &= (Z_u)u + (Z_w)w + [U_o + Z_q]q - (g\sin\theta_o)\theta + Z_{\delta c}\delta_c + Z\delta_B\delta_B \\
\dot{q} &= (M_u)u + (M_w)w + (M_q)q + M_{\delta c}\delta_c + M\delta_B\delta_B \\
\dot{\theta} &= q
\end{aligned}$$

$$\begin{aligned}
\dot{x} &= [A]\{x\} + [B]\{u\} \\
x &= \{u,\ w,\ q,\ \theta\}^T \\
u &= \{\delta_c,\ \delta_B\}^T
\end{aligned}$$

Units

u (ft/sec)

w (ft/sec)

q (rad/sec)

θ (rad)

δ_c (inches)

δ_B (inches)

20 kts (U_o = 33.73 ft/sec)

θ_o = 0.0361 rad $\quad$ δ_{co} = 8.72 in $\quad$ δ_{Bo} = 0.4 in

$$[A] = \begin{bmatrix} -.0154 & .0193 & .6176 & -32.15 \\ -.1978 & -.4699 & 33.79 & -1.162 \\ .0204 & .0017 & -3.4423 & 0 \\ 0 & 0 & 1.0 & 0 \end{bmatrix} \qquad [B] = \begin{bmatrix} .2412 & .7813 \\ -9.3763 & .5217 \\ .0823 & -.9712 \\ 0 & 0 \end{bmatrix}$$

40 kts (U_o = 67.49 ft/sec)

Θ_o = 0.0284 rad δ_{co} = 8.03 in δ_{Bo} = 1.31 in

$$[A] = \begin{bmatrix} -.0245 & .0253 & .1898 & -32.16 \\ -.1277 & -.6648 & 67.41 & -.915 \\ .0223 & .01 & -3.4724 & 0 \\ 0 & 0 & 1.0 & 0 \end{bmatrix} \qquad [B] = \begin{bmatrix} .1651 & .7363 \\ -10.165 & 1.111 \\ .2566 & -.9717 \\ 0 & 0 \end{bmatrix}$$

60 kts (U_o = 101.26 ft/sec)

Θ_o = 0.0103 rad δ_{co} = 7.89 in δ_{Bo} = 2.39 in

$$[A] = \begin{bmatrix} -.0338 & .0311 & 1.044 & -32.17 \\ -.0564 & -.7886 & 101.45 & -.331 \\ .0179 & .0129 & -3.6151 & 0 \\ 0 & 0 & 1.0 & 0 \end{bmatrix} \qquad [B] = \begin{bmatrix} .1583 & .7037 \\ -11.436 & 1.797 \\ .5163 & -.9962 \\ 0 & 0 \end{bmatrix}$$

80 kts (U_o = 135.01 ft/sec)

$\Theta_o = -0.015$ rad $\quad \delta_{co} = 8.13$ in $\quad \delta_{Bo} = 3.26$ in

$$[A] = \begin{bmatrix} -.0423 & .0292 & 4.03 & -32.17 \\ -.0158 & -.8734 & 135.03 & .483 \\ .0153 & .0170 & -3.63 & 0 \\ 0 & 0 & 1.0 & 0 \end{bmatrix} \qquad [B] = \begin{bmatrix} .0515 & .6957 \\ -12.714 & 2.563 \\ .7645 & -1.038 \\ 0 & 0 \end{bmatrix}$$

100 kts (U_o = 168.58 ft/sec)

$\Theta_o = -0.0489$ rad $\quad \delta_{co} = 8.85$ in $\quad \delta_{Bo} = 4.47$ in

$$[A] = \begin{bmatrix} -.0524 & .0269 & 10.12 & -32.13 \\ .0026 & -.9411 & 168.43 & 1.57 \\ .0183 & .0250 & -3.60 & 0 \\ 0 & 0 & 1.0 & 0 \end{bmatrix} \qquad [B] = \begin{bmatrix} -.1082 & .7361 \\ -13.90 & 3.362 \\ 1.011 & -1.095 \\ 0 & 0 \end{bmatrix}$$

Figure Captions

Fig. 1 A block diagram representation of the GPC algorithm.

Fig. 2 Incorporating the SISO GPC algorithm in a MIMO flight control system.

Fig. 3 A stability augmentation system for the BO-105C vehicle.

Fig. 4 The $\dot{h}/\dot{h}_c$' transfer function for the system of Fig. 3.

Fig. 5 The u/u_c transfer function for the system of Fig. 3, altitude rate loop closed.

Fig. 6 Comparison of actual and reduced order h/u_{GPC} transfer functions, calculated with all loops in Fig. 3 closed.

Fig. 7 The w'-plane locus of closed-loop roots of h/h_c of Fig. 3 as a function of GPC control increment weighting λ.

Fig. 8 The w'-plane Bode diagram of loop transmission of GPC design.

Fig. 9 Altitude and airspeed responses of system of Fig. 3 to unit step altitude command, h_c.

Fig. 10 Altitude tracking performance of system of Fig. 3.

Fig. 11 Airspeed tracking performance of system of Fig. 3.

Fig. 11 Airspeed tracking performance of system of Fig. 3.

Fig. 12 Collective inputs of system of Fig. 3.

Fig. 13 Longitudinal cyclic inputs of system of Fig. 3.

Fig. 14 A comparison of altitude tracking errors for systems of Fig. 3 and 15.

Fig. 15 A conventional control system design.

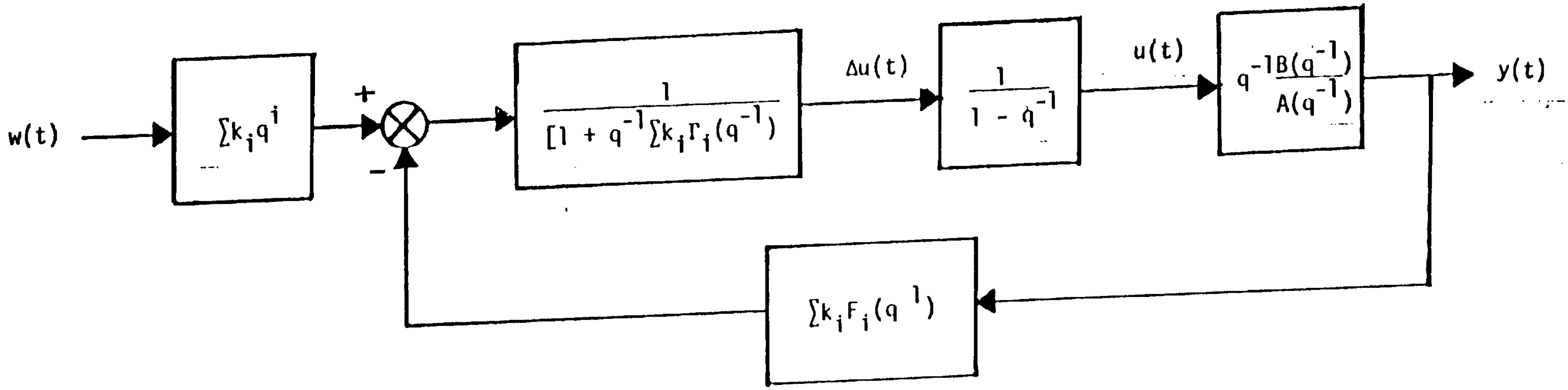

w(t)
$\sum k_i q^i$
+
−
$\frac{1}{[1 + q^{-1}\sum k_i \Gamma_i(q^{-1})}$
Δu(t)
$\frac{1}{1 - q^{-1}}$
u(t)
$q^{-1}\frac{B(q^{-1})}{A(q^{-1})}$
y(t)
$\sum k_i F_i(q^{-1})$

Fig. 1 A block diagram representation of the GPC algorithm.

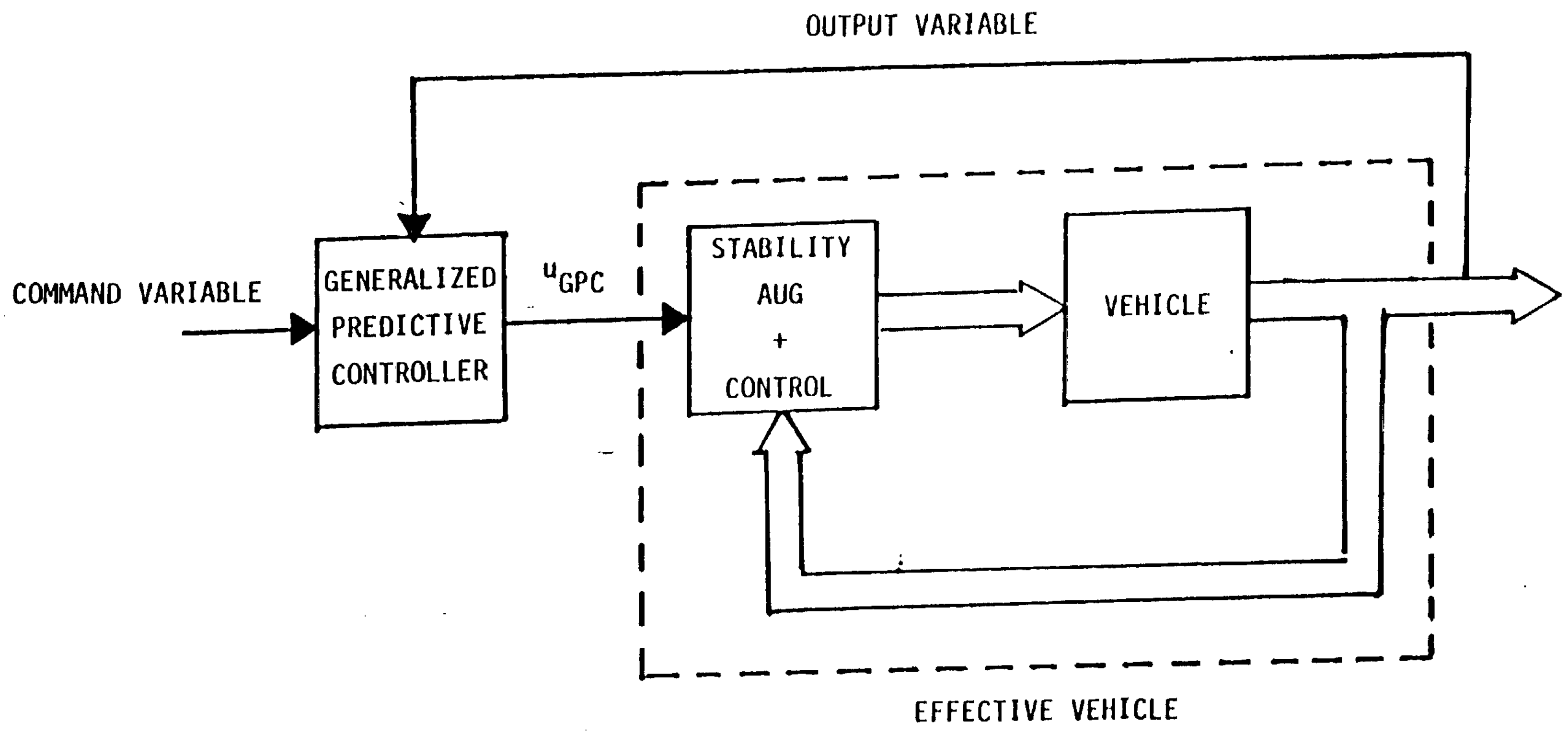
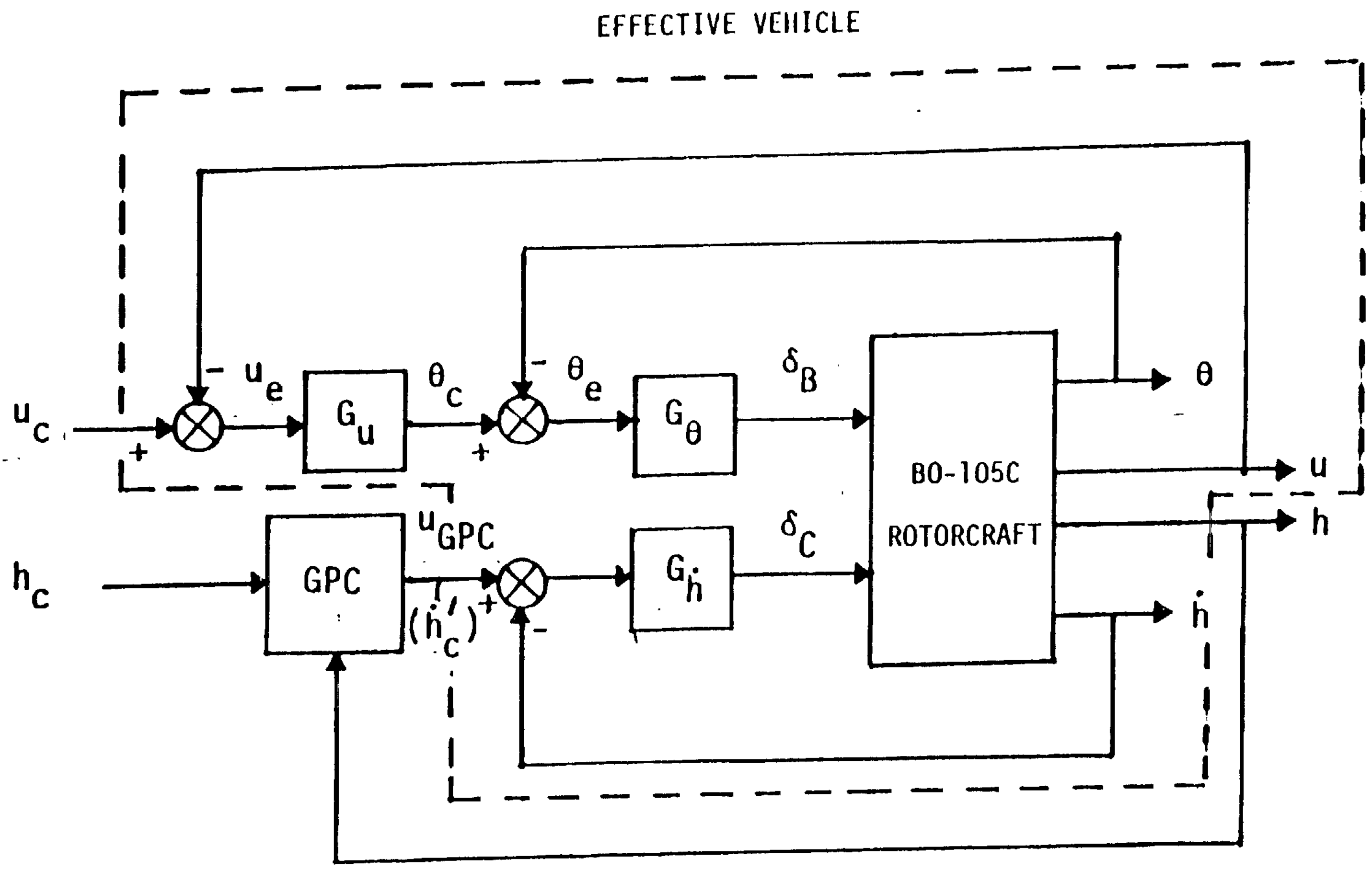


Fig. 2 Incorporating the SISO GPC algorithm in a MIMO flight control system.



$$G_\theta = - \frac{3.44(s+1)(s+.5)}{s(s/20+1)} \text{ in/rad} \qquad G_u = -.0222 \text{ rad/(ft/sec)}$$

$$G_{\dot{h}} = 0.4 \text{ in/(ft/sec)}$$

Fig. 3 A stability augmentation system for the BO-105C vehicle.

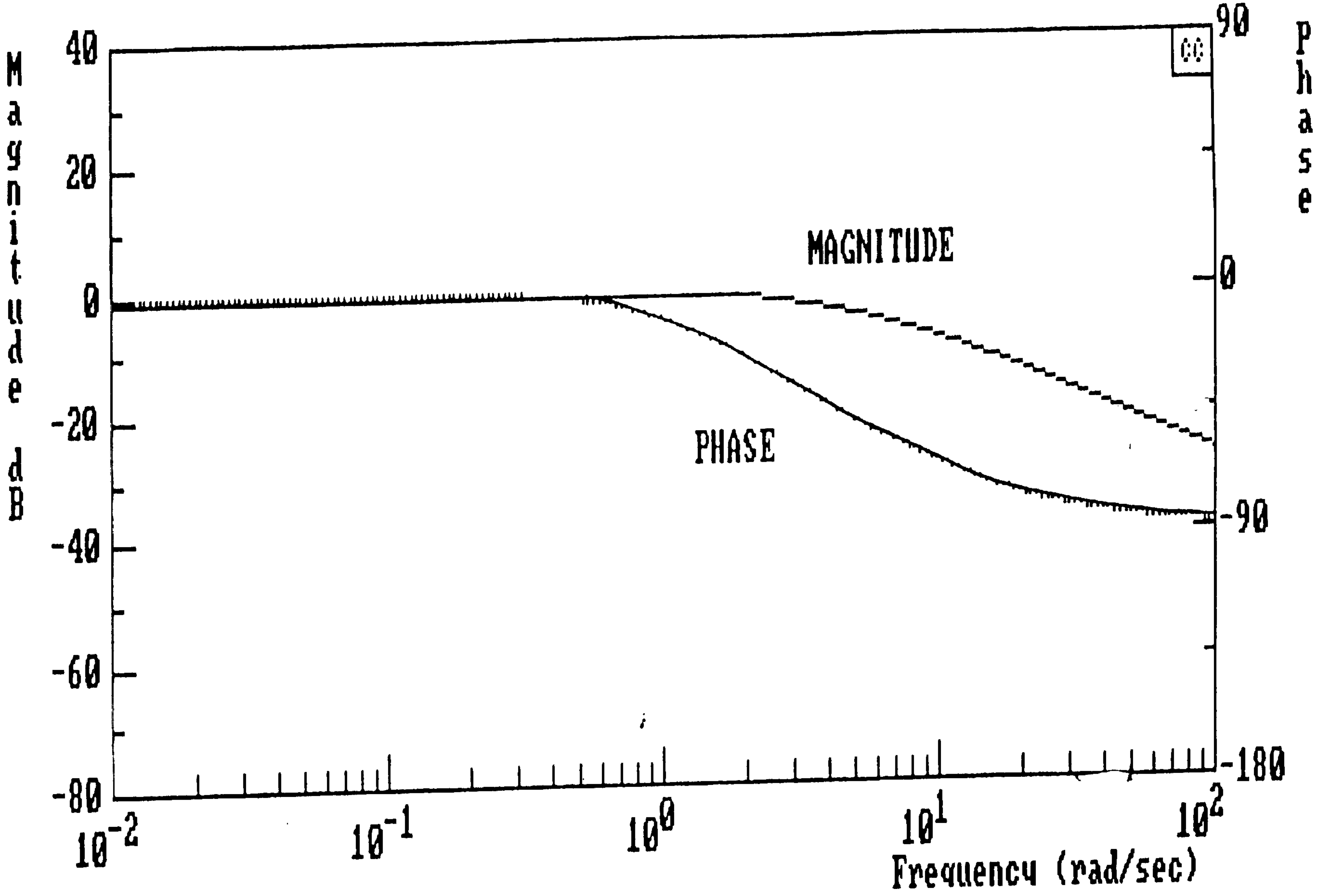


Fig. 4 The $\dot{h}/\dot{h}_o$' transfer function for the system of Fig. 3.

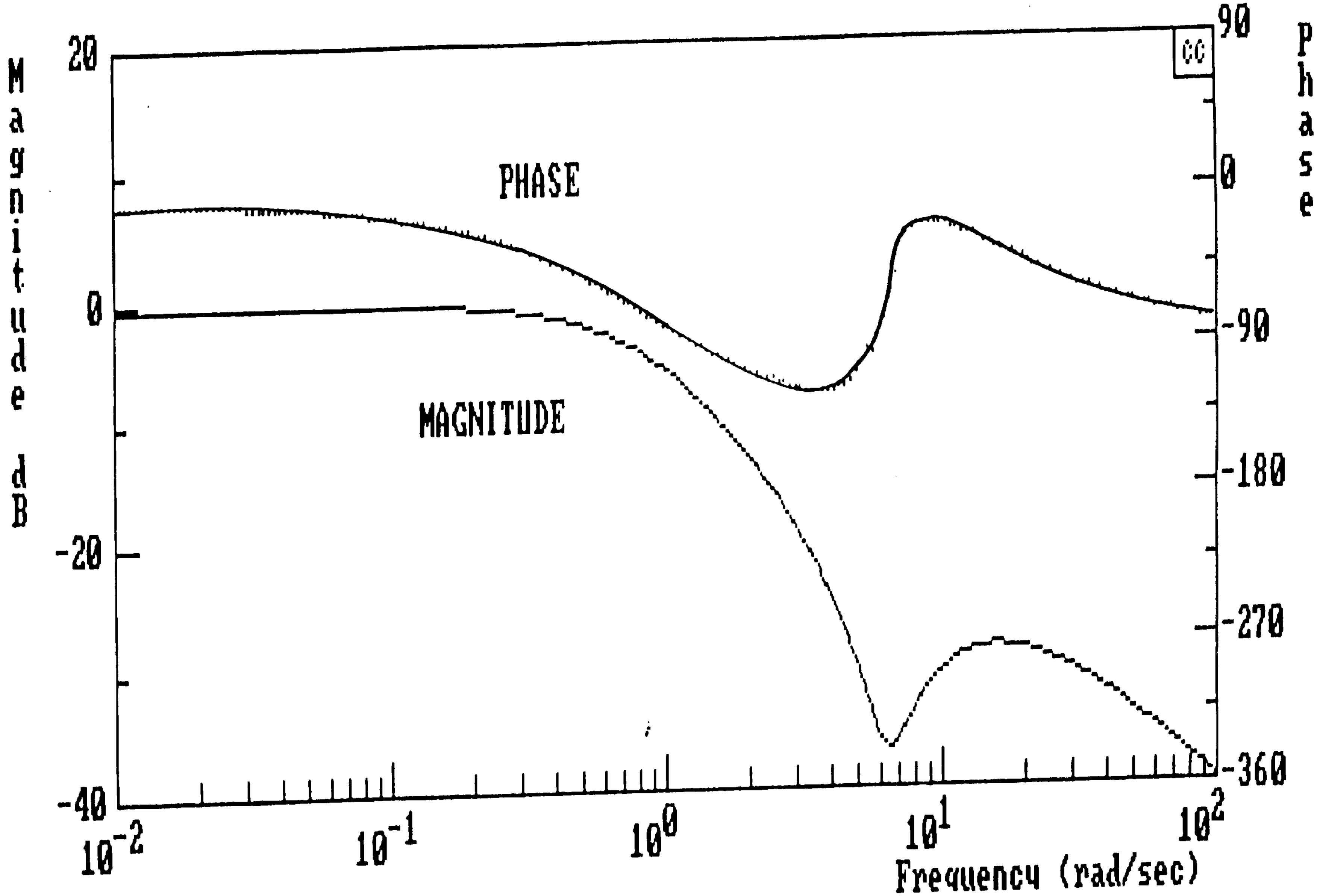


Fig. 5 The u/u_a transfer function for the system of Fig. 3, altitude rate loop closed.

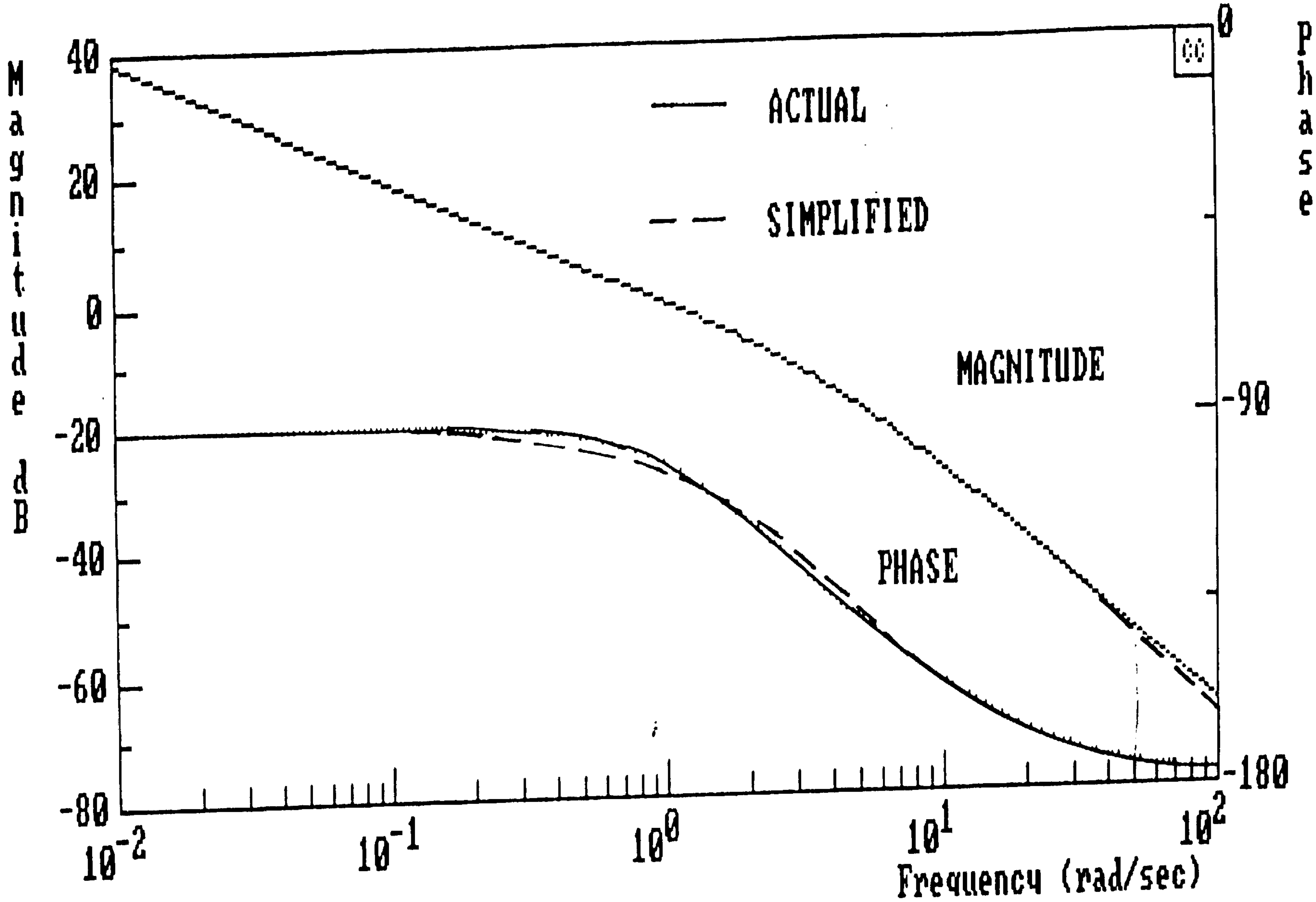


Fig. 6 Comparison of actual and reduced order h/u_{aro} transfer functions, calculated with all loops in Fig. 3 closed.

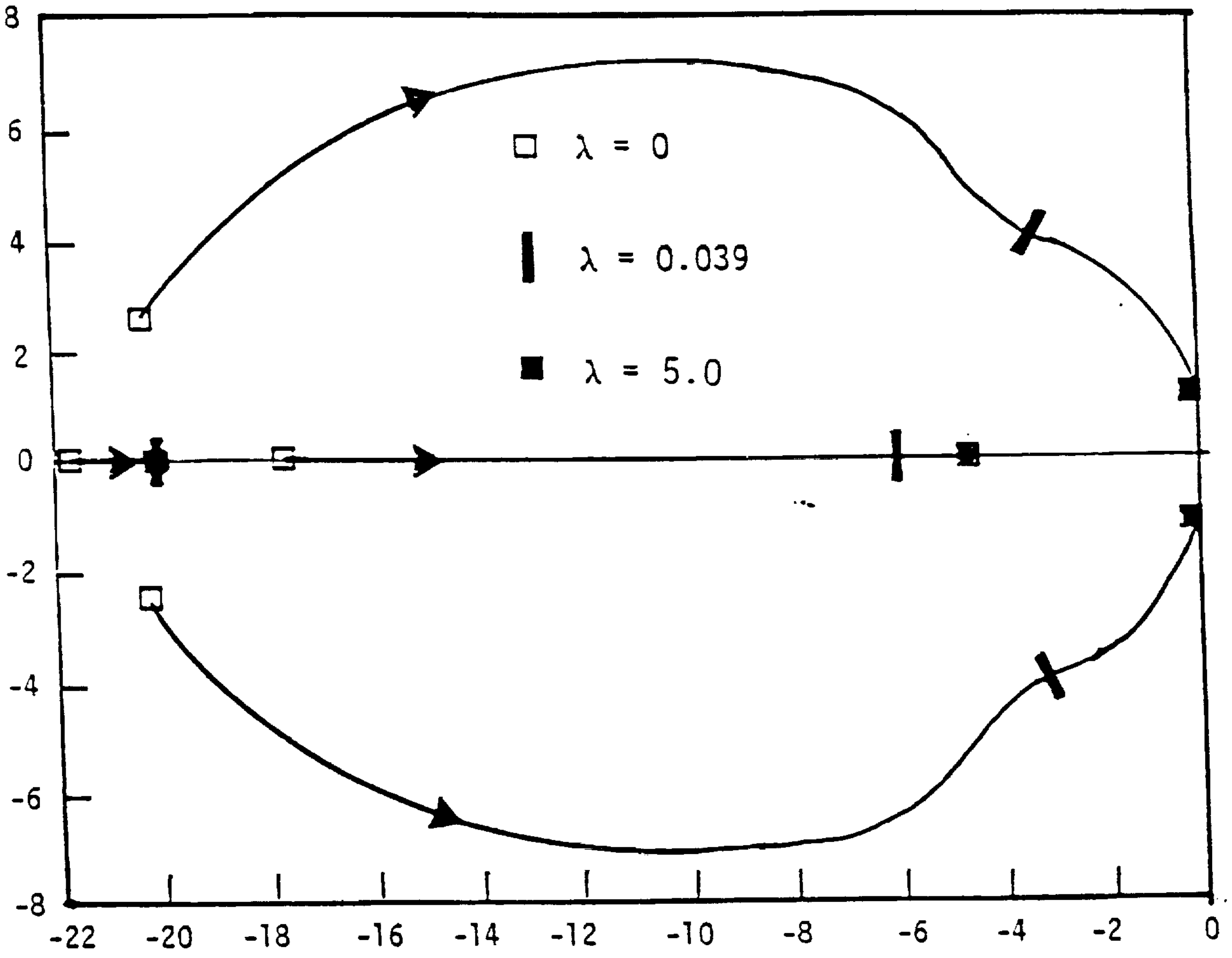


Fig. 7 W'-plane locus of closed-loop roots of h/h_c of Fig. 3 as a function of GPC control increment weighting λ .

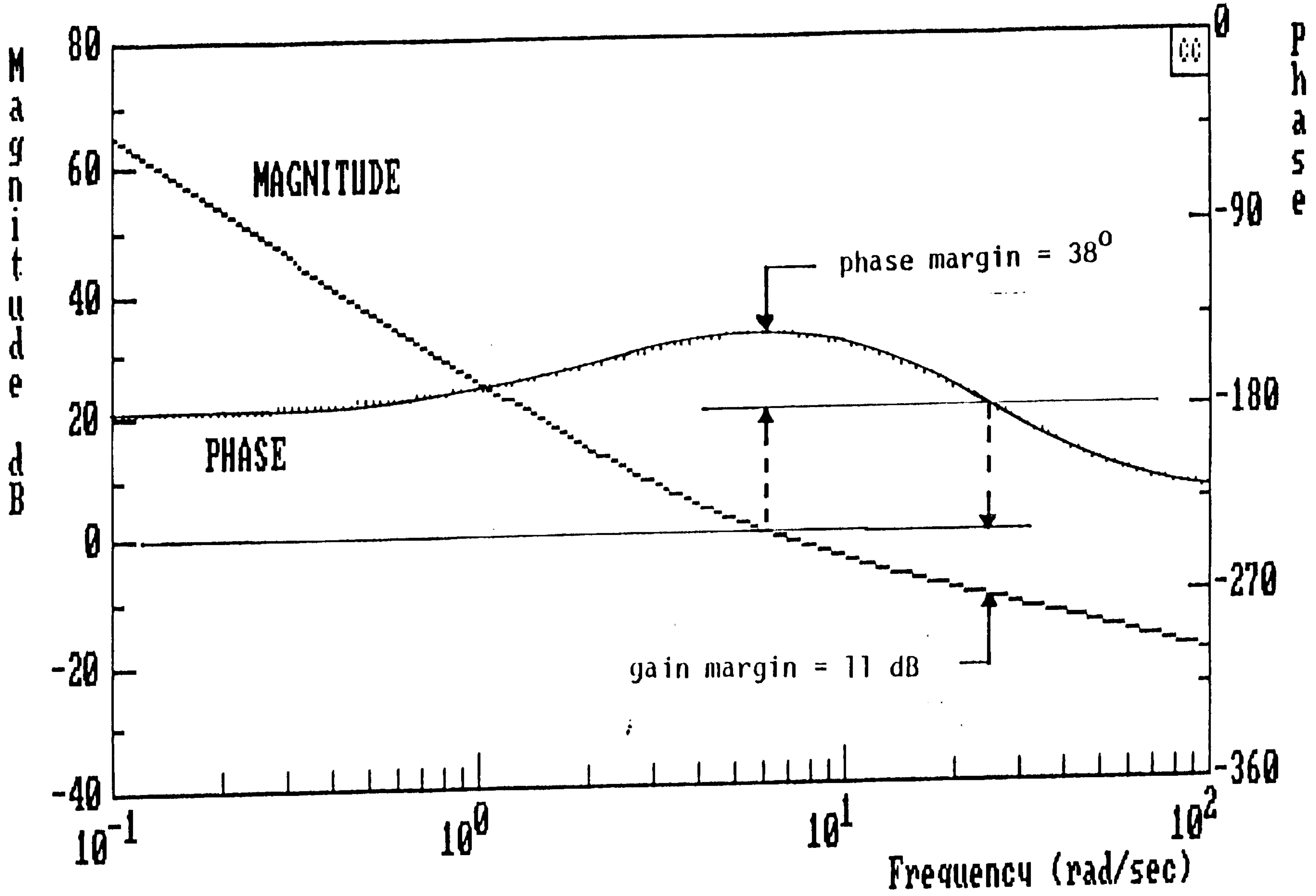


Fig. 8 W'-plane Bode diagram of loop transmission of GPC design.

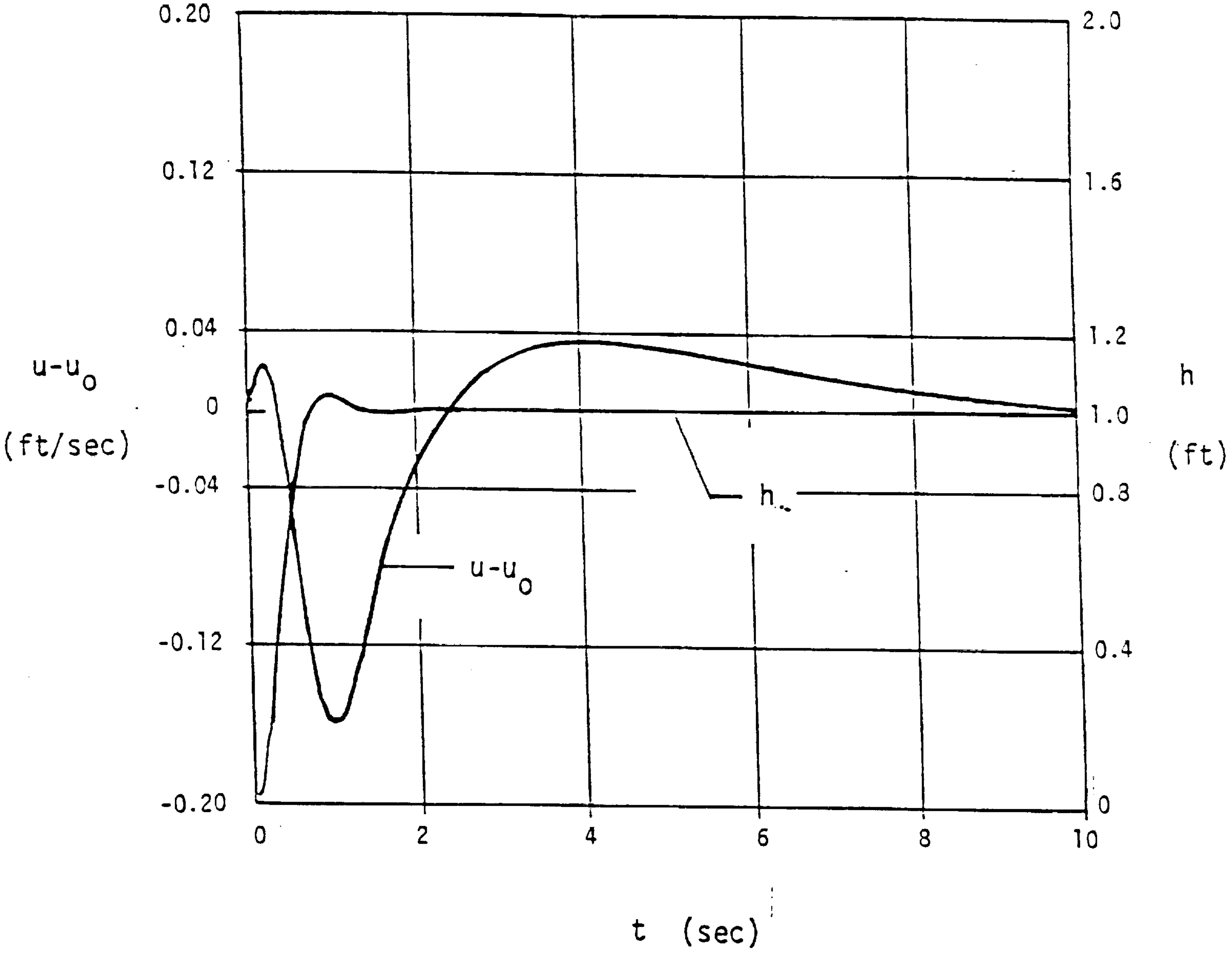


Fig. 9 Altitude and airspeed responses of system of Fig. 3 to unit step altitude command, h_c.

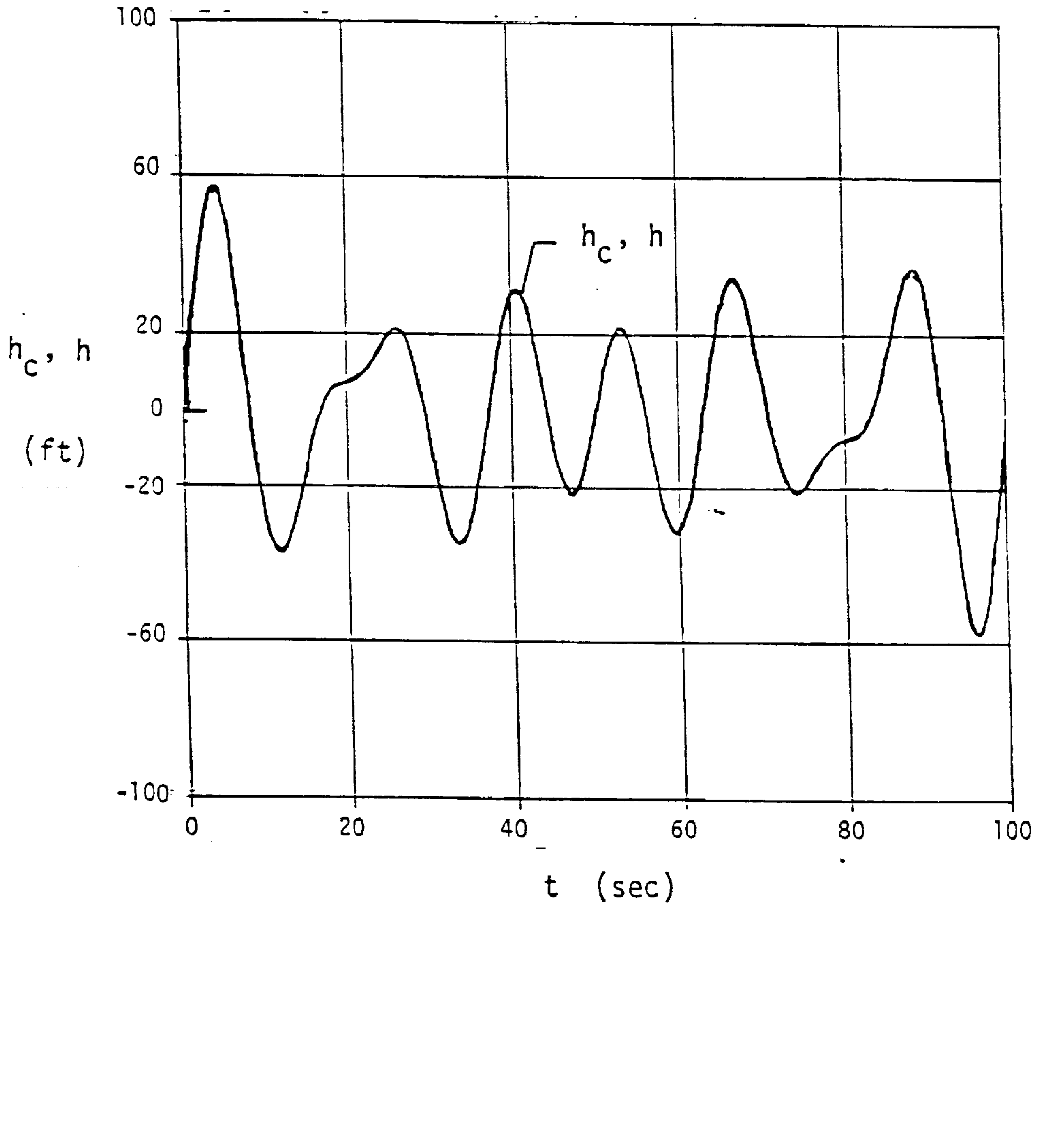


Fig. 10 Altitude tracking performance of system of Fig. 3.

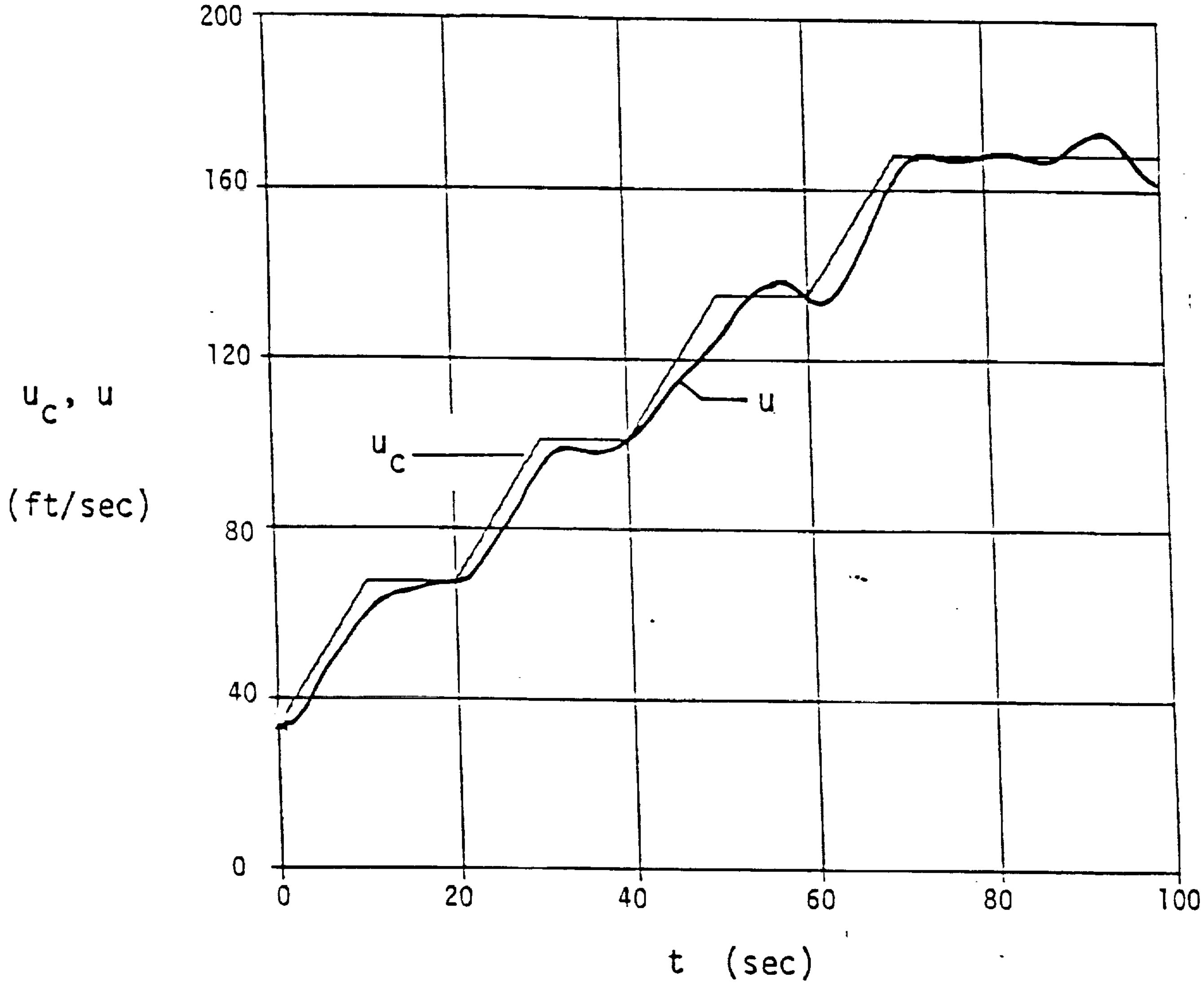


Fig. 11 Airspeed tracking performance of system of Fig. 3.

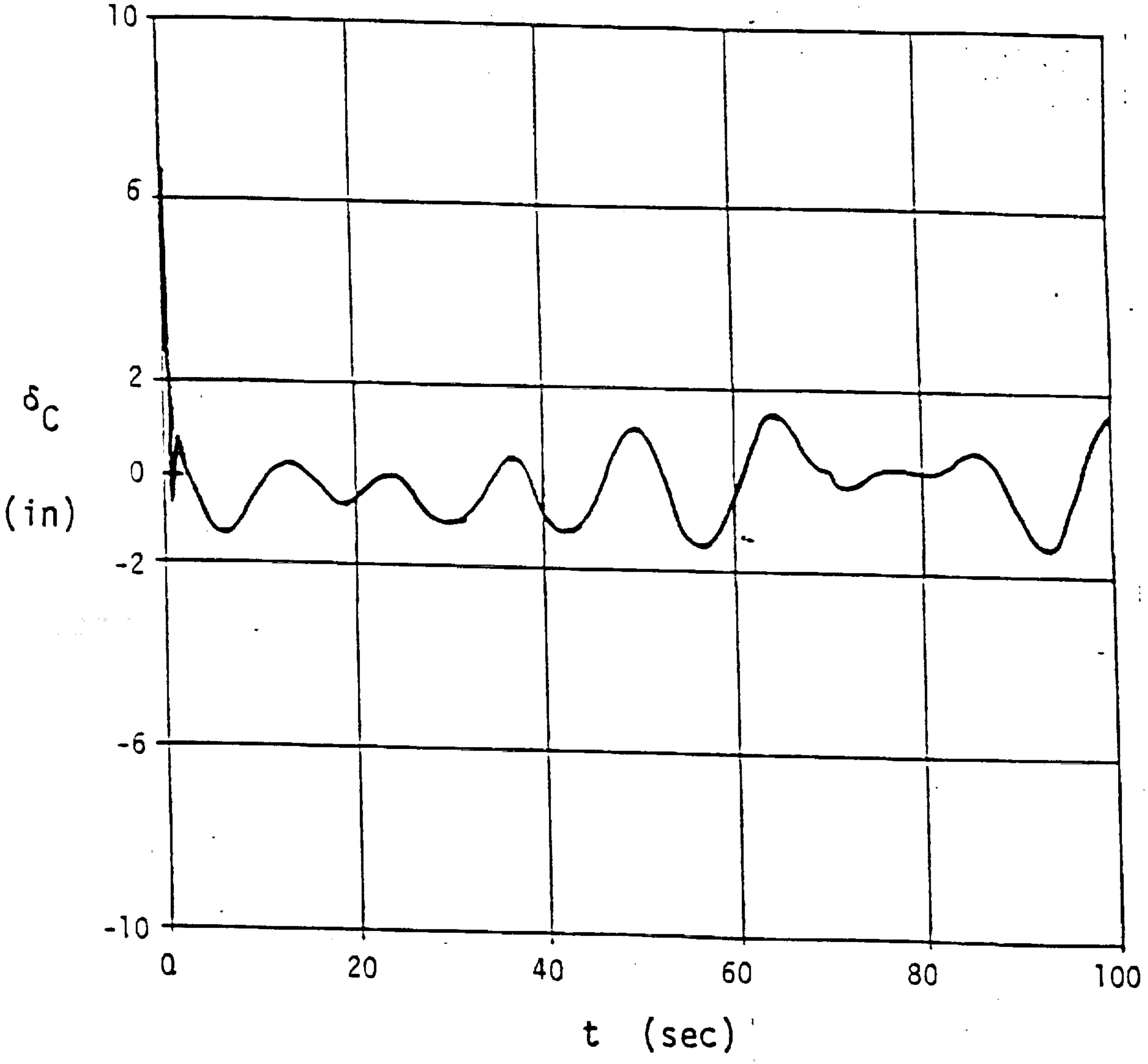


Fig. 12 Collective inputs of system of Fig. 3.

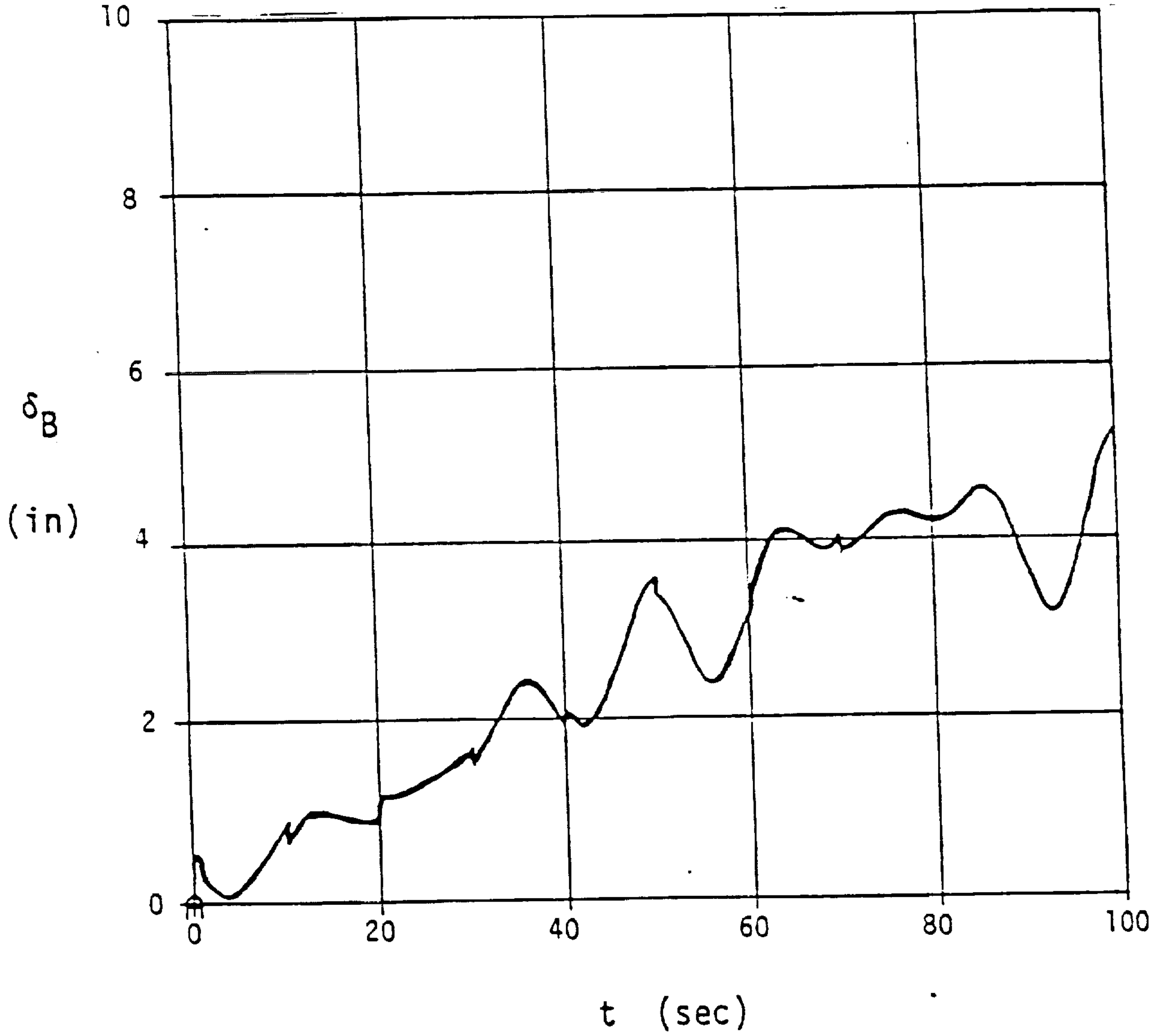


Fig. 13 Longitudinal cyclic inputs of system of Fig. 3.

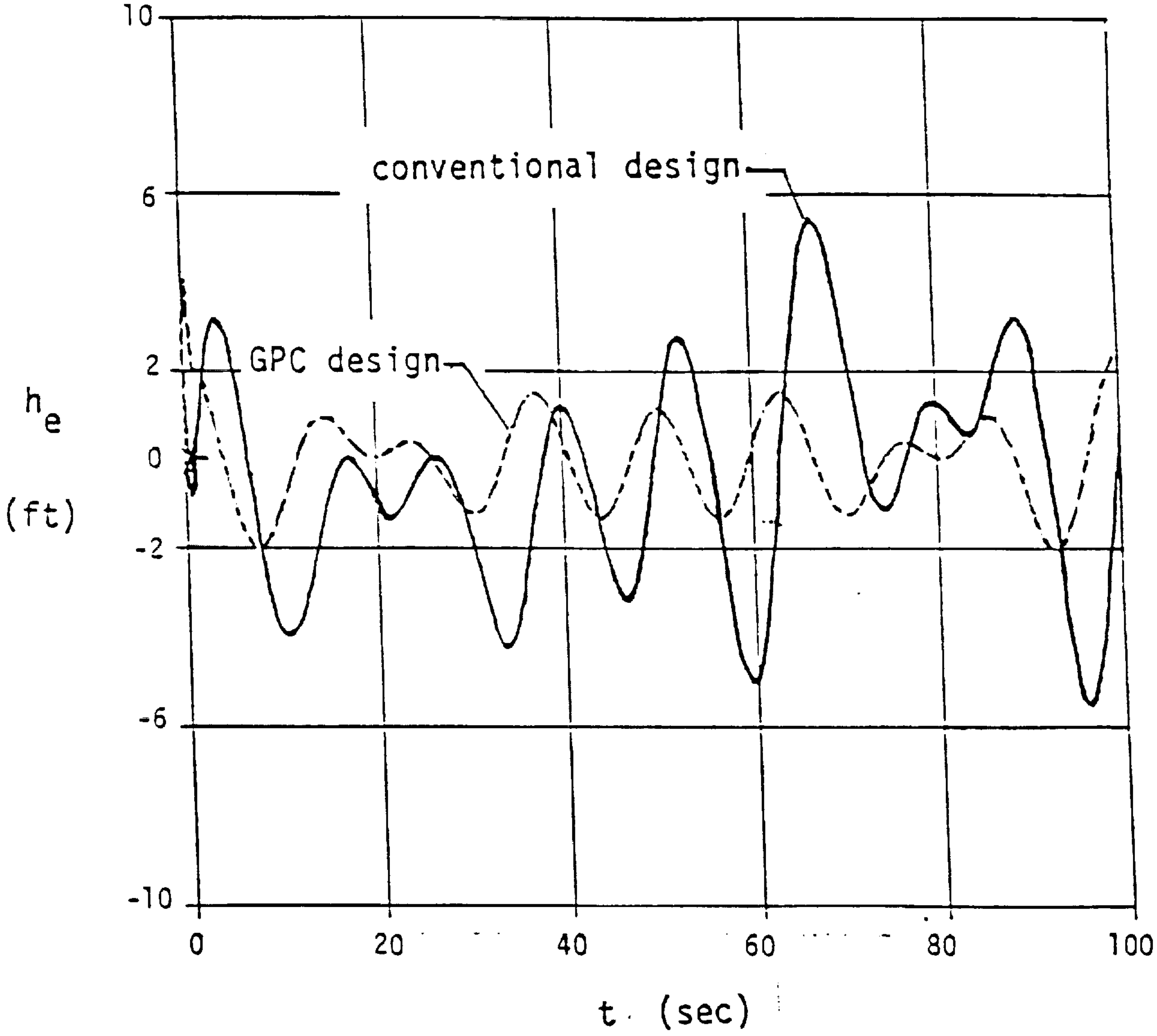
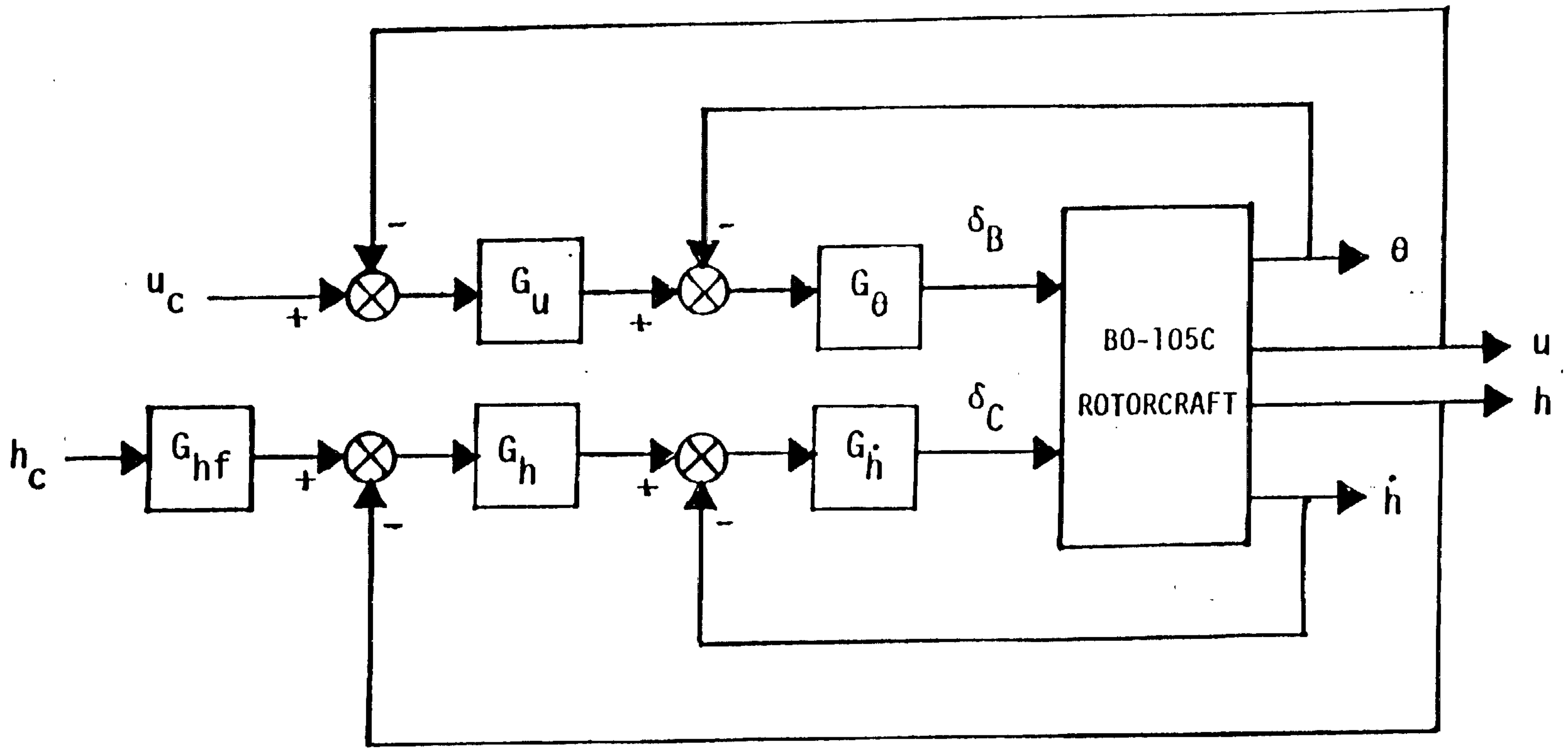


Fig. 14 A comparison of altitude tracking errors for systems of Fig. 3 and 15.



$$G_h = \frac{0.819(s/1.72+1)}{(s/20+1)} \quad 1/\mathrm{sec} \qquad G_{hf} = \frac{(s/.786+1)}{(s/10+1)}$$

Fig. 15 A conventional control system design.