# ROBUST COMMAND AUGMENTATION SYSTEM DESIGN USING GENETIC METHODS

G. D. Sweriduk[*] and P. K. Menon[†]
*Optimal Synthesis*
450 San Antonio Road, Suite 46
Palo Alto, CA  94306

M. L. Steinberg[‡]
Aeromechanics Division
Naval Air Warfare Center
Patuxent River, MD  20670

## Abstract

This paper describes the use of a genetic search method in the design of a command augmentation system for a high-performance aircraft. A genetic algorithm is used to design H-infinity controllers for the longitudinal and lateral-directional channels. The integral of absolute value of error between actual response and that of an ideal model is used as the fitness criterion. Starting from an initial population of weighting functions, the algorithm generates new functions with the goal of improving the fitness.

## I. Introduction

Current practice in flight control design is to design controllers for a matrix of flight conditions. Each design can be a tedious process, and there is a significant amount of trial-and-error involved. The parameters are selected and then the closed-loop system is simulated. What is more, modern requirements for a fighter aircraft[Enns, et al.] go well beyond natural frequency and damping requirements. There are a plethora of desirable qualities, making the design process even more difficult. The purpose of this paper is to present preliminary results of ongoing research into the use of genetic search methods to help automate and accelerate the design process.

A genetic search, in comparison with other optimization techniques, is desirable for two reasons. The first reason is that all sorts of criteria may be used for minimization or maximization. There is no problem with fitness measures (i.e. cost functions) which are discontinuous or non-smooth. The fitness measure may not look anything like the forms used in other techniques. This allows the designer a great deal of freedom, as well as the ability to choose criteria which more closely represent the actual goals, rather than having to interpret those goals into some mathematical function which adds a layer of abstraction. The second advantage of a genetic search is that it can be used to design control *laws*, rather than just parameters. In other words, the structure of the controller does not have to be specified in advance, with only some numerical constants to be optimized. Properly set up, a genetic search can piece together mathematical functions to form control laws.

This study will use a genetic search to design a command augmentation system (CAS) with H-infinity control based on linearized models over the flight envelope of an F/A-18 aircraft. These controllers will then be evaluated on a 6 degree-of-freedom nonlinear model of the aircraft.

## II. Genetic Algorithms and Genetic Programming

Only a brief description of genetic algorithms and genetic programming will be given in this paper. The interested reader is referred to [Menon, et al.] and the references below for more information.

The basic concepts of genetic algorithms were set forth by [Holland,75], which showed how the evolutionary process could be applied to artificial systems[Koza,92]. The genetic algorithm is a mathematical algorithm which transforms a set (population) of mathematical objects into a new set using operations similar to reproduction and survival of the fittest, as Charles Darwin described in natural populations[Darwin,1859]. The operations are called reproduction, crossover, and mutation. Each new population is called a generation. The fitness of each member of

---

[*] Research Scientist, Senior Member, AIAA

[†] President, Associate Fellow, AIAA

[‡] Flight Control Engineer, Member, AIAA

the current generation is evaluated according to some specified function. The members with the best fitness are more likely to be selected to be carried over to the next generation (reproduction) or used to create offspring (crossover) which will be included in the next generation. Members with poor fitness are more likely to be eliminated from the population. Members can also be selected at random and altered (mutation).

Throughout the 1980s extensions to the standard genetic algorithm were proposed [Holland, De Jong] (note - see Koza p.64). In the standard genetic algorithm the members are usually fixed-length strings. The strings are made up of binary numbers which can represent real numbers or just actions (i.e. fast vs. slow, high vs. low, etc.). A string therefore represents a set of numbers or a sequence of actions. By breeding and mutating these strings, new combinations are formed, and the new strings are evaluated for fitness. However, the length of the string, and the sturcture of the solution, is always fixed in genetic algorithms. In genetic programming the complexity of the members undergoing adaptation is much greater. The members may be rules such as logical operators. In this way a genetic algorithm can be used to create computer programs to solve a specific problem. According to Koza, "...the structures undergoing adaptation in genetic programming are active. They are not passive encodings of the solution to the problem. Instead, given a computer on which to run, the structures in genetic programming are active structures that are capable of being executed in their current form." [Koza,92,p.76]. It is this process of creating programs that leads to the term "genetic programming." There are variations of the classical genetic algorithm and genetic programming, but they all share the basic concepts, so they are referred to collectively as genetic methods.

## III. Command Augmentation System Synthesis

Linear models were obtained from a nonlinear, 6 degree-of-freedom model of an F/A-18 aircraft. From the full-order linearized model, a second-order longitudinal model and a fourth-order lateral-directional model were extracted. The states of the longitudinal model are q (pitch rate) and w (body z-axis velocity), the measurements are q and $n_z$ (normal acceleration), and the input is the stabilator. Normal acceleration will be the commanded quantity. The states of the lateral-directional model are v (body y-axis velocity), $\phi$ (roll angle), p (body axis roll rate), and r (body axis yaw rate). The measurements are $p_s$ (stability axis roll rate), r, and $n_y$ (lateral acceleration), and the inputs are effective aileron and rudder. Roll control is achieved with a combination of ailerons and differential stabilator, which is referred to here as effective aileron. The quantities to be commanded are $p_s$ and $n_y$.

Two separate controllers will be designed, one for the longitudinal dynamics and one for the lateral-directional dynamics. Figures (1) and (2) show the generalized plants which are to be used for the designs. There are poles, zeros, and gains of the weighting functions, which the engineer would select based on certain guidelines and also trial-and-error. The weights will be expressed in terms of free parameters. The performance variables are the weighted control signals and the errors between actual outputs and the outputs of ideal models. The ideal models represent the dynamics which are classified as Level 1 handling qualities in [DOD,MIL-STD-1797]. Uncertainties are included at the plant input in the longitudinal case and at the output in the lateral-directional case, and disturbances are included in all measurements.

The weights are chosen by the genetic algorithm and the controller is designed with commercially-available H-infinity design software (MATLAB with μ-Tools Toolbox). In the longitudinal case, the free parameters determine the weights as follows:

$$W_{s1} = \frac{K5\left(s + (K6 + K7)\right)}{(s + K7)}, \quad W_{s2} = K4$$

$$W_{\Delta} = \frac{1/(1 + K1)(s + (K2 + K3))}{(s + K3)}$$

$$W_u = K8$$

$$W_{d1} = W_{d2} = 0.01$$

where K1 - K8 are the free parameters and represent real numbers. Both weights $W_{s1}$ and $W_{s2}$ are on error signals between the actual output and the output of an ideal model. The penalty is higher at low frequencies and lower at high frequencies, so the zero should be faster than the pole. Since the emphasis is on normal acceleration, less weight is placed on pitch rate, and hence $W_{s2}$ is expected to be a small, constant value. For $W_{\Delta}$, the uncertainty is expected at low frequency, and low gain above that, so the zero is faster than the pole, and the gain is constructed to ensure the gain is less than 1 at high frequencies. $W_u$ is the weight on control and is a constant value over all frequencies. A frequency-dependent weight could have been used but this approach seems to work and keeps the order of the controller, and the number of free parameters, smaller. $W_{d1}$ and $W_{d2}$ are noise levels and are held fixed.

The free parameters in the lateral-directional case K9 - K17 determine the weights as follows:

$$W_{s1} = \frac{1/(1+10*K9)\left(s+10^{-4}\right)\left(s+(K10+K11+K12)\right)}{\left(s+10^{-4}+K11\right)\left(s+10^{-4}+K12\right)}$$

$$W_{s2} = \frac{1/(1+10*K13)(s+K14+K15)}{s+K15}$$

$$W_{t1} = W_{t2} = W_{t3} = \frac{0.01(s+1)}{s+10}$$

$$W_{u1} = K16\,,\quad W_{u2} = K17$$

$$W_{d1} = W_{d2} = W_{d3} = 10^{-3}$$

Both weights $W_{s1}$ and $W_{s2}$ are on error signals between the actual output and the output of ideal models. The weight for roll rate, $W_{s1}$, is largest in the middle frequencies. A zero at very low frequency is inserted to reduce gain at low frequency. There are two poles faster than this zero, and another zero above the poles. The gain is constructed to ensure that the magnitude is less than 1 at high frequencies. The weight on lateral acceleration, $W_{s2}$, is constructed to have high gain at low frequency and gain less than one at high frequency. The weights on output uncertainty, $W_{t1}$, $W_{t2}$, and $W_{t3}$, are fixed for all designs and reflect increased uncertainty at higher frequencies. The weights on controls, $W_{u1}$ and $W_{u2}$, are constants. The weights on noise, $W_{d1}$, $W_{d2}$, and $W_{d3}$, are held fixed at a small value.

The values for the weights have the following parametrization:

$u_i = 1,\ldots, 9, 10^{0}, 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{1}, 10^{2} \qquad i = 1,\ldots, 16$

$K(n) = u_i * u_j$

Each of the free parameters K(n) will have its own separate population. A member, then, for this work, means a set of members, each drawn from one of the K(n) populations. This set will be evaluated and assigned a value of fitness, so that each element of the set has the same fitness associated with it.

In this work a set of parameters is sought for designing H-infinity controllers. While the problem could be set up using binary strings, as in the classical genetic algorithm, an alphanumeric parametrization is used instead, which adds flexibility. The standard genetic algorithm represents choices with binary strings of fixed length. Such an arrangement describes discrete values. Using the above parametriztion, and assuming fixed strings with only one multiplication, there would be 16 x 16 = 256 possible values for each free parameter. For H-infinity design, the poles, zeros, and gains of the weighting functions are seldom required to have more than one or two significant digits, so that the expressions can be kept fairly simple. Any positive, real number could be represented by multiplication and addition of the parameters $u_i$ if expressions were allowed to grow longer with more operations. However, although expressions are manipulated, the end result is still just the formation of real numbers without affecting the structure of the control law. Hence, this is really a genetic algorithm without using binary strings.

Each member is evaluated by substituting the values into the block diagrams, forming the generalized plant, obtaining the H-infinity controllers, and then simulating the closed-loop system. Not all members will produce a controller. The controller synthesis function requires an initial guess for the upper and lower bounds of the closed-loop H-infinity norm. The upper limit is set fairly high, and any member which does not saitisfy the upper bound is rejected. Given a controller, fitness is determined by evaluating the responses of the closed-loop system with various inputs. The inputs are step functions to normal acceleration in the longitudinal case and roll rate and lateral acceleration in the lateral-directional case. An example of how the fitness is determined is shown in Fig. 3. The idea is to try to match the behavior of the ideal model, or minimize the error between the ideal model and the actual system. The goal, then, is to obtain the lowest possible fitness. Additional terms were added to penalize cross-coupling between $p_s$ and $n_y$, and non-minimum phase behavior, and high values of infinity norm. Also, for practical reasons, controllers with right-half-plane poles or very fast poles are rejected. A penalty was placed on the closed-loop infinity-norm bound, $\gamma$, as well.

The initial population is generated randomly. After evaluation of the initial population, the process of crossover is carrried out to generate new members. A maximum population size is maintained, and members with high fitness are deleted from the populations. This process is continued until a desirable response is achieved. In evolutionary processes it is difficult to say that any particular member is optimal, so deciding when to stop the genetic algorithm is somewhat subjective. In theory, running the algorithm longer will continue to produce better results. Several runs are usually conducted.

The genetic search method used in this work differs from the classical algorithm in a few ways. Firstly, only one type of operation, crossover, is used. Secondly, selection of the members for operations is done randomly rather than on the basis of fitness. Thirdly, in the traditional meaning of genetic methods, a generation is the population after all members have been acted upon. However, in this work, a generation refers to the population after one pair of members has undergone crossover. In the traditional sense, all of the members have undergone reproduction (replication) except the two which have undergone crossover.

## IV. Results

Results for the case sea level, Mach 0.6, are presented. For the longitudinal case, three runs were made, each starting with a population of 500 members, a maximum population size of 500, and lasting 1000 generations. The third run produced the best result. The best fitness of the initial population was 0.0331, and the best fitness after 1000 generations was 0.0187. While the changes may appear small, the differences are significant. The controller with the lowest fitness produced a closed-loop infinty norm of 0.8805, and the simulation showed good performance. In the lateral case, two runs of 600 generations were made (less generations were used in the lateral case due to the slower run time). The second run produced the better result, with a fitness of 0.0240 and a closed-loop infinity norm of 5.8588, and good performance in the simulation.

The free parameters obtained were:
longitudinal
6.0000e+00
5.0000e+00
1.0000e+00
1.0000e-05
4.0000e-03
6.0000e+00
3.0000e-02
1.0000e-02

lateral
1.0000e+00
7.0000e+02
4.0000e-01
6.0000e-02
9.0000e-02
5.4000e+01
1.0000e-02
6.0000e+00
1.0000e+00

The controller poles were:

longitudinal
-1.8103e+01
-9.0605e+00
-1.2258e+00
-2.9929e-02
-2.1010e+00+ 2.1425e+00i
-2.1010e+00 - 2.1425e+00i
-2.1000e+00+ 2.1424e+00i
-2.1000e+00 - 2.1424e+00i

lateral
-2.2297e+01
-1.2002e+01+ 7.3827e+00i
-1.2002e+01 - 7.3827e+00i
-3.3034e+00
-2.3509e+00+ 7.0236e-01i
-2.3509e+00 - 7.0236e-01i
-1.0574e-01
-4.2186e-02+ 6.9832e-02i
-4.2186e-02 - 6.9832e-02i
-7.5030e-04
-1.0000e+01
-1.0000e+01
-1.0000e+01

The closed-loop responses to various inputs are shown in Figs. (4), (5), and (6). The first is the response to a 0.1 g pulse command to normal acceleration, the second response is with a 0.1 rad/sec doublet command to stability axis roll rate, and the third is a 0.1 g pulse to lateral acceleration. The closed-loop system responds well to all three inputs. There is some unavoidable nonminimum phase behavior.

## V. Conclusions

With the parametrization used in this study, the genetic algorithm has produced good results. Admittedly, the fitness criterion is rather simple. It should be possible to add further criteria. Fitness could be defined by directly computing handling qualities metrics, similar to the NASA/Army code CONDUIT[ ]. Also, the use of nonlinear simulations with the complete aircraft model would give a more accurate evaluation and thus would be preferable. However, this was impractical with current computer resources, given the computation time required for nonlinear simulations.

The code which implements the genetic search is being expanded to perform all of the standard operations and to include a variety of selection methods. More study of the effects of initial and maximum population size is needed.

## VI. References

[ ] Enns, D., et al., "Application of Multivariable Control Theory to Aircraft Control Laws; Interim Report: Definition of Control Problems and Requirements," Air Force Report WL-TR-96-3071, January, 1996.

[ ] Menon, P.K., Yousefpor, M., Lam, T., and Steinberg, M.L., "Nonlinear Flight Control System Synthesis Using Genetic Programming," *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, New Orleans, LA, 1996, pp. ??-??

[ ] Holland, J.H., *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, MI, 1975.

[ ] Darwin, C.A., *On the Origin of Species by Means of Natural Selection*, John Murray, London, 1859.

[ ] DeJong, K.A., "On Using Genetic Algorithms to Search Program Spaces," *Genetic Algorithms and Their Applications: Proceedings of the Second International Conference on Genetic Algorithms*, J. J. Grefenstette, editor, Erlbaum Press, 1987.

[ ] Koza, J.R., *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, MIT Press, Cambridge, MA, 1992.

[ ] United States Department of Defense, “Flying Qualities of Piloted Vehicles, MIL-STD-1797”, March, 1987.

[ ] Tischler, M.B., et. al, “CONDUIT - A New Multidisciplinary Integration Environment for Flight Control Development,” NASA Technical Memorandum 112203, June 1997.

## Acknowledgements

This work is being supported by the United States Navy under the Department of Defense SBIR (Small Business Innovations Research) program, contract number N62269-5361-1291. The authors would like to acknowledge the contribution of The Boeing Company to this work. In particular, the first author would like to express his appreciation to Kevin Saari and Jeff Zimmerman for their assistance in validating the model and Kevin Wise and Joe Brinker for lots of helpful advice.

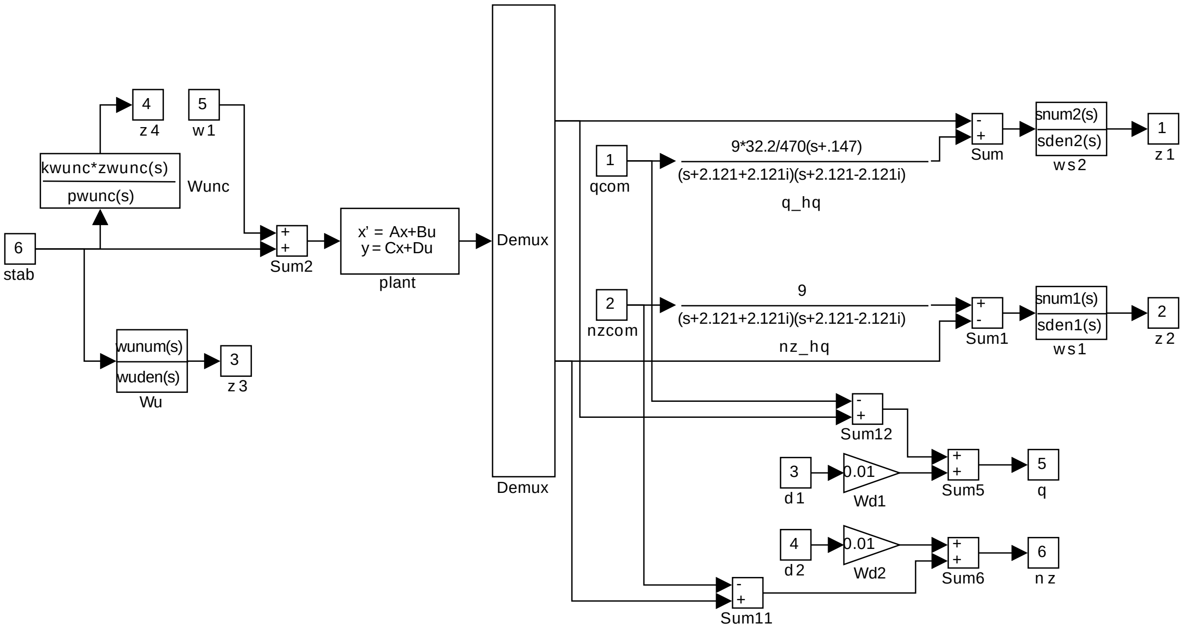

**Figure 1: Longitudinal Block Diagram**

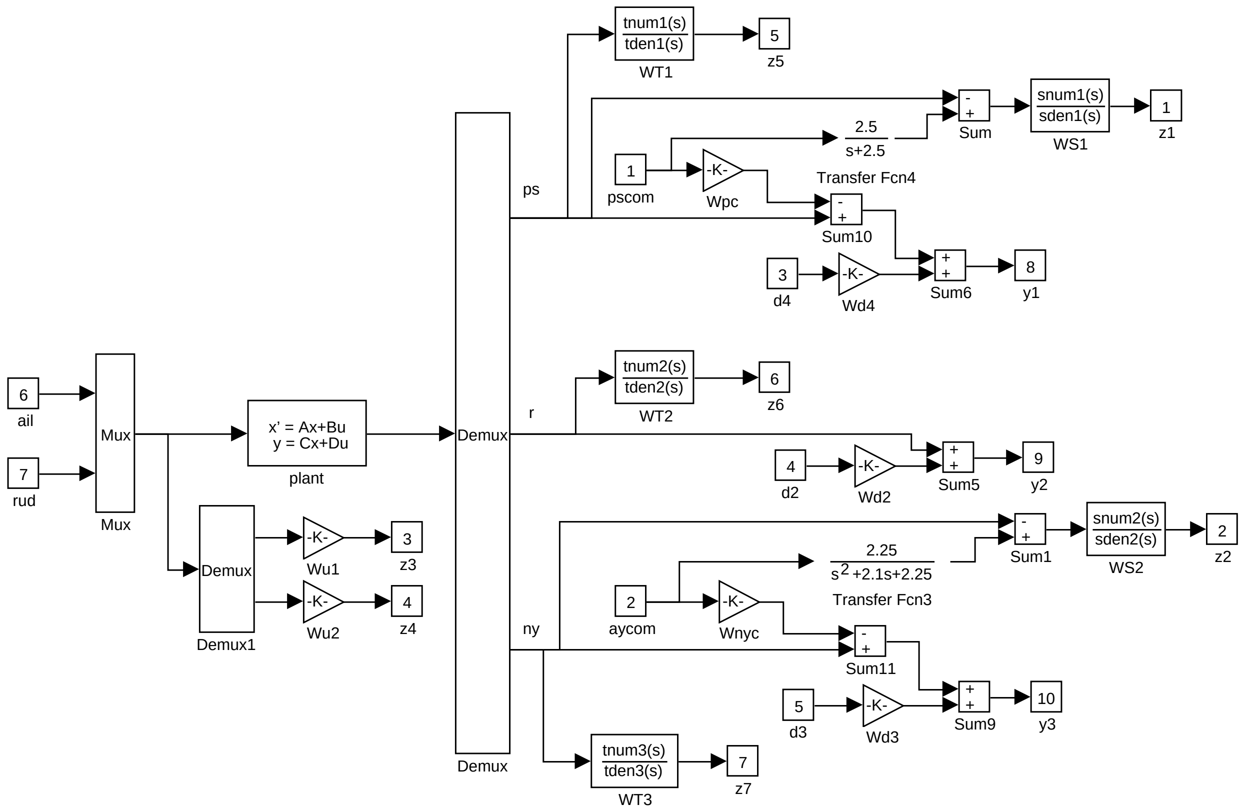

**Figure 2: Lateral-Directional Block Diagram**

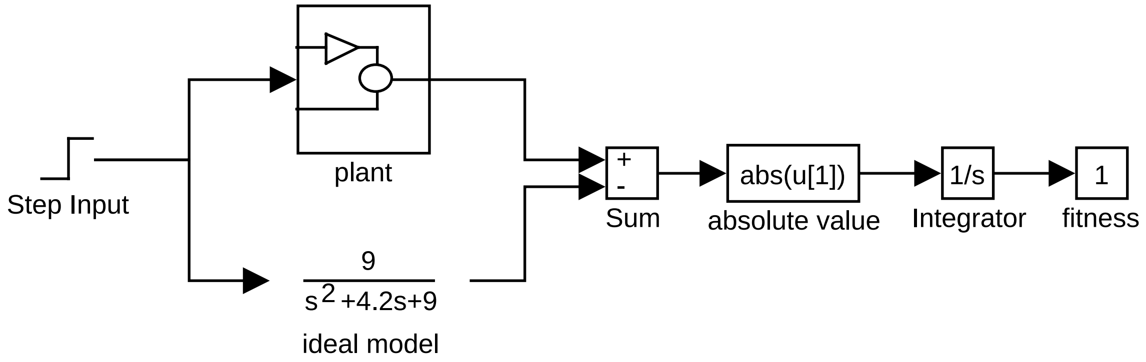

**Figure 3: Fitness Calculation**

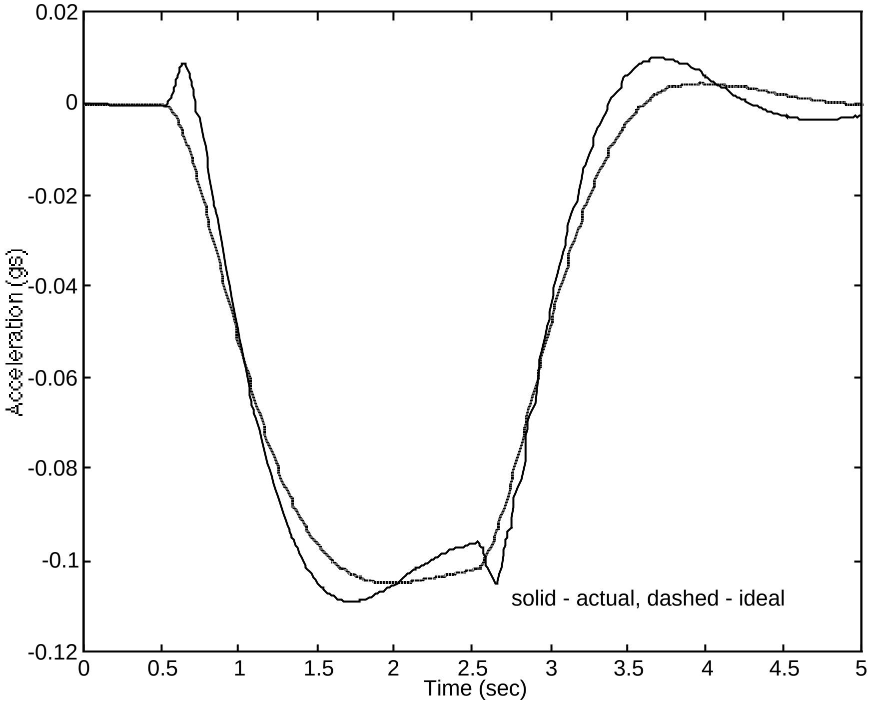

Figure 4: nz response

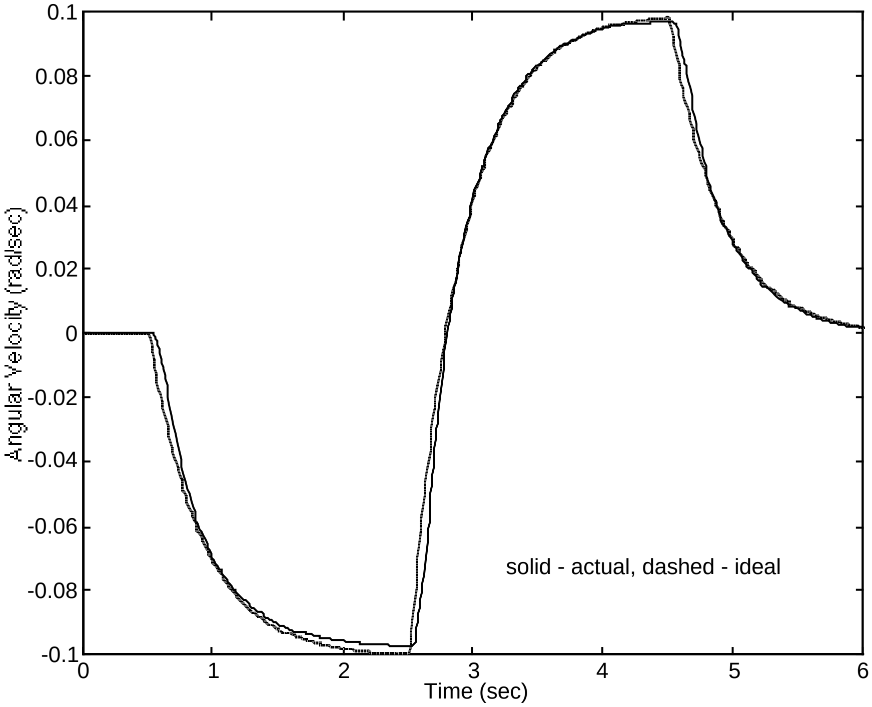

Figure 5: ps response

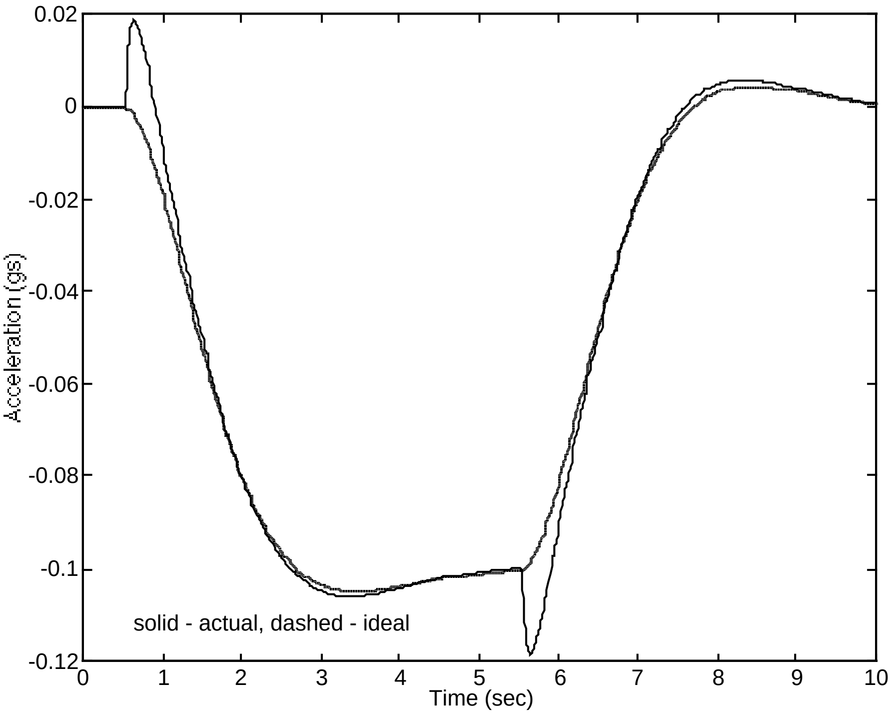

**Figure 6: ny response**