AIAA 99-4588

ADAPTIVE NEURAL CONTROL FOR MACE II

D. C. Hyland and D. P. Scharf
Department of Aerospace Engineering
The University of Michigan
Ann Arbor, MI

AIAA 1999 Space Technology
Conference and Exposition
28 – 30 September 1999
Albuquerque, NM

AIAA 99-4588

ADAPTIVE NEURAL CONTROL FOR MACE II

David C. Hyland[*] Daniel P. Scharf[†]

Aerospace Engineering Department
The University of Michigan, Ann Arbor, MI

Abstract

The Middeck Active Control Experiment (MACE) is being reflown as MACE II. Its express purpose is to gain on-orbit experience with a variety of adaptive controllers. Over a 9 month period the University of Michigan team has developed a number of neural network based adaptive approaches and taken these approaches from concept to flight software. The control schemes considered, and for which ground test results are presented, are: 1) a baseline classical design, 2) a fast, online, neural identification plus deadbeat control approach and 3) a neural, Hebbian Learning based 'classical optimizer' approach. The Hebbian algorithm also includes a Stability Margin Evaluator (SME) that ensures no closed loop unstable controllers are implemented. The classical designs compare well to more complicated, fixed gain designs from the original MACE flight. The online neural identification performs extremely well (ID to noise floor in 60 ms), but non-minimum phase zeros prevent deadbeat control implementation. Finally, the Hebbian Learning algorithm increased the performance of its initializing controller significantly in only 3 minutes without an instability by evaluating and selectively testing some 700 controller gain sets.

Introduction

The original Middeck Active Control Experiment (MACE) was developed by NASA Langley Research Center, the Massachusetts Institute of Technology and Payload Systems, Inc. The experiment was successfully flown on STS-67 in March 1995 and was designed to serve as a pathfinder for a qualification procedure for flexible, precision controlled spacecraft.[1] The technical focus of MACE was the investigation of robust, high-performance but fixed-gain controllers for instrument pointing control. More recently, the MACE hardware was refurbished for a second flight experiment designed to test the capabilities of adaptive controllers for precision pointing. This second flight test, called MACE II, is intended to test a variety of adaptive control approaches.[2] Reported in this paper is work performed in late 1998 and over the winter and spring of 1999 to design, develop, analyze and ground test a set of adaptive controllers, many based upon a neural network architecture, for inclusion into the planned MACE II flight experiment.

The impetus for the present work was the realization that contemplated future space missions will require a higher degree of autonomy in their control operations. This is because these systems will entail complex, high performance controls that will necessarily be sensitive to estimated system parameters and will be comprised of many components with a high probability of some component failures during the mission. Secondly, future concepts involve remote and long duration operations, during which the use of conventional, nonadaptive controls would necessitate cumbersome and costly ground station operations. In response to these factors, we believe it necessary to develop self-reliant control systems. In the context of this paper self-reliance is taken to be comprised of four basic features: (1) autonomous, multi-level mission

[*] Professor and Chair, dhiland@engin.umich.edu

[†] Ph.D. Candidate, dscharf@engin.umich.edu

HYLAND

planning and control, (2) comprehensive adaptive control capabilities, (3) the capability for identifying faults and working around problems to recover control effectiveness, and (4) the ability to effectively communicate critical events to human agents and to respond to human direction when such direction becomes necessary.

By addressing adaptive control capabilities, MACE II will take a modest step toward self-reliant systems. The specific goals of the University of Michigan study, performed under subcontract to Planning Systems, Inc., are to (1) demonstrate in a space flight environment the capabilities of adaptive neural control (ANC), (2) explore and resolve practical implementation issues for ANC on space systems, (3) experimentally test, via vibration control, a selection of ANC algorithms drawn from a variety of layers of the necessarily multi-layer hierarchy of intelligent control needed for self-reliant systems and (4) correlate experimental data and pre-flight predictions to increase the confidence with which ANC can be used in future missions.

The overall technical approach for MACE II emphasizes two distinct types of controls. The first category may be termed " N-Step ahead controls" (see for example, Reference 3). These are applicable to fast control loops for lower-level regulation and tracking tasks. They are generally based on the backpropagation learning mechanism and address supervised learning tasks (e.g. vibration control) wherein at every time, the desired response of the system is known and is provided to the learning algorithm for appropriate adaptation of the on-line control. A second and quite distinct type of algorithm may be termed "Hebbian" learning algorithms because they do not employ backpropagation but instead are based on the earlier and considerably simpler learning paradigm of Hebb.[4] Hebbian algorithms are applicable to more general situations than the N-Step ahead controls. In particular, its applicability includes infinite horizon optimization and unsupervised learning, in which the "right answer" or desired behavior are not known or provided to the system at every time step. Instead learning is guided by a regime of rewards and penalties imposed by the environment. In this work, both of the broad types of control are applied to active suppression of vibration-induced line-of-sight (LOS) pointing error within the context of the MACE II hardware and within the limitations of the flight processor.

MACE II Hardware Description

The MACE configuration was designed to represent an actively controlled, high-payload-mass-fraction spacecraft, such as earth-observing platforms, with multiple, independently pointing or scanning payloads. The test article consists of a flexible bus to which two payloads, a reaction wheel assembly and other actuators and sensors are mounted. Each payload is mounted to the structure using a two-axis gimbal that provides pointing capability. Instrumentation includes angle encoders on each gimbal axis, a three-axis rate gyro platform mounted under the reaction wheel assembly, and a two-axis rate gyro platform mounted in the primary payload. The bus is composed of circular cross-section struts connected by aluminum nodes. The structure is supported for ground tests by a pneumatic/electric suspension system.

Because the suspension cables, the bus and gravity vectors are all in the same plane, the structural dynamics decouple into vertical (about the Z-axis) and horizontal (about the X and Y axes) dynamics. The control is implemented using a real-time computer operating at a 500-Hz sampling rate. In referring to various control loops in the following, we resort to acronyms for the various control actuators and sensors. In particular, "PGZt" denotes the primary gimbal torque along the Z-axis and, likewise, "PGXt" refers to the device exerting torque along the X-axis. "PGRGZ" and "PGRGX" are the corresponding, colocated primary gimbal rate gyro outputs. Similarly, the nomenclature "SGZt" and “SGXt” refers to the torque actuators on the secondary gimbal. Further, "BRW1-3" refers to the bus reaction wheels (numbered 1 to 3) and "BRG1-3" refers to the colocated bus rate gyro outputs.

Development Process and Candidate Designs

The approach taken here to the design, development and testing of various neural net-based controllers consisted of several steps. First, we assimilated and explored the computational models of the MACE II hardware and, based on model results, formed a list of potentially useful candidate controllers. For each controller, we developed what we considered to be an implementable algorithm and then developed the corresponding deliverable software in accordance with MACE software design specifications. To troubleshoot practical issues, we tested a subset of the algorithms on the MHPE testbed [5] at the University of Michigan Space Structures Lab, then ground-tested the selected algorithms in a sequence of test sessions at the MACE ground test facility at the Air Force Research

HyLAND

Lab, Albuquerque, New Mexico. The finally validated algorithms were then delivered to Planning Systems Inc. the prime contractor responsible for experiment integration.

A set of common, underlying assumptions provided the basis for our design work. First, disturbances were provided by the secondary gimbals and these devices were not used as control actuators. We considered only controllers that closed loops around the primary gimbals (PGXt, PGZt) and primary rate gyros (PGRGX, PGRGZ) or the bus reaction wheels (BRW1-3) and bus rate gyros (BRG1-3). As in the first MACE flight, we defined the performance measure to be the primary gimbal LOS errors. These are not measured directly but are estimated by integrating up the PGRG outputs.

Based on these assumptions and preliminary analysis, a suite of six distinct control candidates was formed: (1) classical design, (2) ARMArkov Information Filter + N-Step Predictive Control, (3) Positive -Real + Adaptive notches, (4) Hebbian learning with classical framework, (5) Adaptive LQG design, and (6) combined design with colocated loops for the piezo actuators and strain gauges. We give a brief description of each of these designs in turn, as follows.

1. Classical design: Nonadaptive classical design using independent loops around colocated pairs of devices. This design gives a basic performance benchmark and yields design insight.

2. ARMArkov Information Filter + N-Step Predictive Control: Uses an ARMArkov identifier with Information Filter algorithm[6] with a standard deadbeat controller. We assume a 2-Input, 2-output design for primary gimbals and rate gyros.

3. Positive-Real + Adaptive Notches: Basic control is a classical-based positive-real control supplemented by adaptively adjusted notch filters to gain -stabilize modes near 180^0 phase crossover. Independent loops are assumed for PGXt + PGRGX and PGZt + PGRGZ. Here, the neural identifier has only to deal with the crossover modes.

4. Hebbian Learning with Classical framework: Here the control structure is fixed in accordance with the insights achieved in design #1 above, but the set of gain parameters is optimized via a Hebbian learning algorithm.

5. Adaptive LQG Design: Use Information Filter algorithm for ARMA model ID with a traditional LQG adaptive design that uses an ARMA model.[7]

6. Combined Design: One of the above plus independent, colocated loops for the piezo actuators and strain gauges. This would be used for a fault-tolerant, rapid controller recovery demo.

Based on the design model studies, it was concluded that controller #3 would probably not be appropriate since a number of anti-aliasing filters in the system introduces significant time delay, such that even the colocated loops are not approximately positive-real over a frequency band of any significant extent. Hence, work on controller #3 was suspended. In addition, while preparations on controller #5 continue for the present, it is likely that this algorithm is too computationally demanding to be accommodated by the MACE processor. Thus the main design thrusts, prior to the final design, #6, are controllers 1,2 and 4. The paper will detail the results obtained for these controllers during experimental tests carried out over the last several months. Further, as most initial work has concentrated on the colocated Z loop (i.e. PGZt to PGRGZ), the following discussion will restrict itself to the Z loop except when some X loop results have been obtained.

Classical Control

Shown in Figure 1 is the experimental open loop frequency response of the PGZt to PGRGZ loop (0.5 Hz to 250 Hz). The sample rate is 500 Hz. Note the two almost imperceptible modes near 40 Hz and 50 Hz with phases very near crossover (there is a small notch in the magnitude plot and two small spikes in the phase plot right before the phase jumps from –180° to +180°). These two modes pose significant design challenges since controllers dealing with the lower frequency modes tend to drive these two modes unstable. This observation lay behind control design #3, a positive real design for the other modes with these two higher modes being notched to stabilize them. Although, as already mentioned, this design was abandoned since the response is not positive real for any appreciable amount of the frequency range.

Also of interest is the small spike near 240 Hz (near the right edge of the magnitude plot in Figure 1) which at first appears to be nothing more than a numerical artifact from the identification routine. However, careful experimentation has shown that to ignore this spike (e.g. through truncation of the frequency response) is to invite instability.

HYLAND

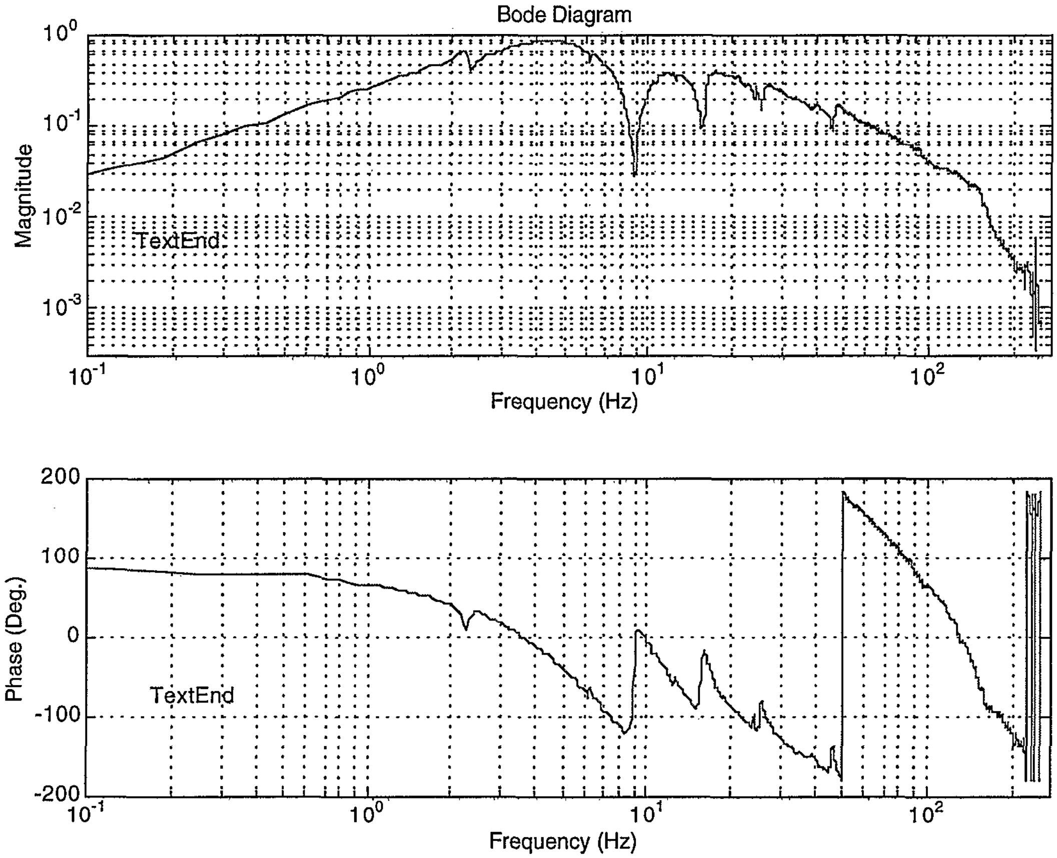

Figure 1. Open Loop Frequency Response: PGZt to PGRGZ

The classical design strategy finally decided upon was to design a loop separately for each mode based on a standard control template. First, however, the output of the plant is passed through a lead-lag filter to give the modes near crossover better phase. The standard templates are applied to the output of the lead-lag filter.

The template consists of a bandpass filter coupled with two notches; the two notches are collectively referred to as a double notch. The bandpass filter has four parameters to tune: overall gain, center frequency, damping ratio and a fourth parameter that effectively shapes the phase (it controls the location of a zero of the filter). The double notch has six parameters: two damping ratios and a center frequency for each notch.

A bandpass is centered on the first mode of the plant/lead-lag system to be attenuated and the parameters adjusted to achieve maximum closed loop performance. Then, turning to a Nyquist perspective, the double notch, its center frequencies being chosen in concert, is used to increase the gain margin by 'cutting out' a keyhole along the negative real axis. The bandpass is then readjusted to take advantage of the increased gain margin.

This process is repeated for subsequent modes until desired performance is achieved or stabilizing another loop proves unfeasible. Finally, a constant gain loop is added to attain as much attenuation as possible.

For MACEII, two template loops were used in the controller. This architecture requires 23 parameters to be chosen (10 for each template, 2 for the overall lead-lag filter and 1 for the constant gain loop). The spike at 240 Hz is responsible for the gain and phase margins of the final pre-ground test design: 2 dB and 9°. The experimental closed loop frequency response of this design is shown in Figure 2, as is the experimental open loop frequency response. Note that these frequency responses are from PGZt to Z LOS (line of sight). They were obtained by integrating the PGRGZ frequency responses.

HYLAND

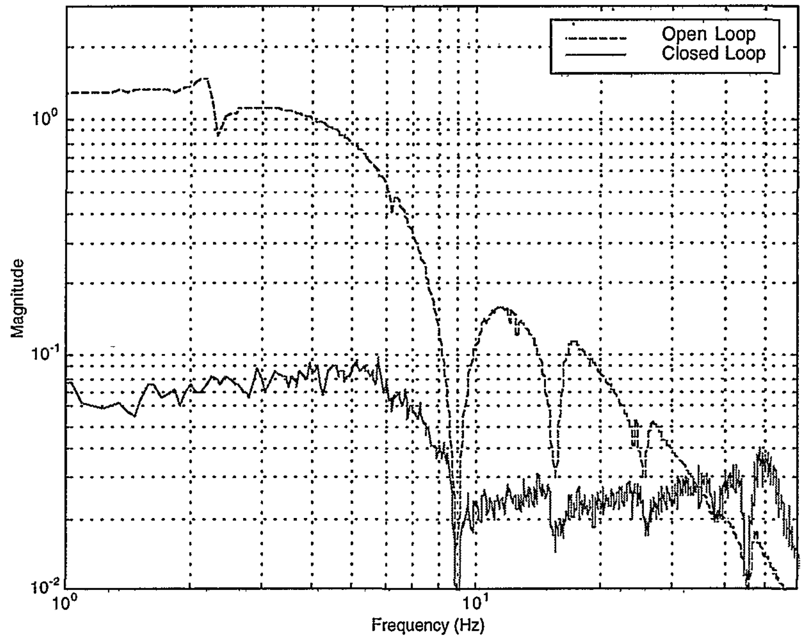

Figure 2. PGZt to Z LOS Open and Closed Loop Experimental Frequency Responses - Classical Design

Of particular interest is how well this controller compares to SISO controllers designed on the ground for the first MACE flight, in particular an LQG design and a Multi-Model (MM) design.[8] In general, the controller does better than the LQG design, only doing slightly worse below 1 Hz and above 40 Hz. The MM achieves better attenuation over the entire frequency range, but above 9 Hz the classical controller is almost as good as the MM design (MM closed loop magnitude is ~0.013).

The results for the x loop are shown below in Figure 3. The gain margin is 2 dB and the phase margin is 35°.

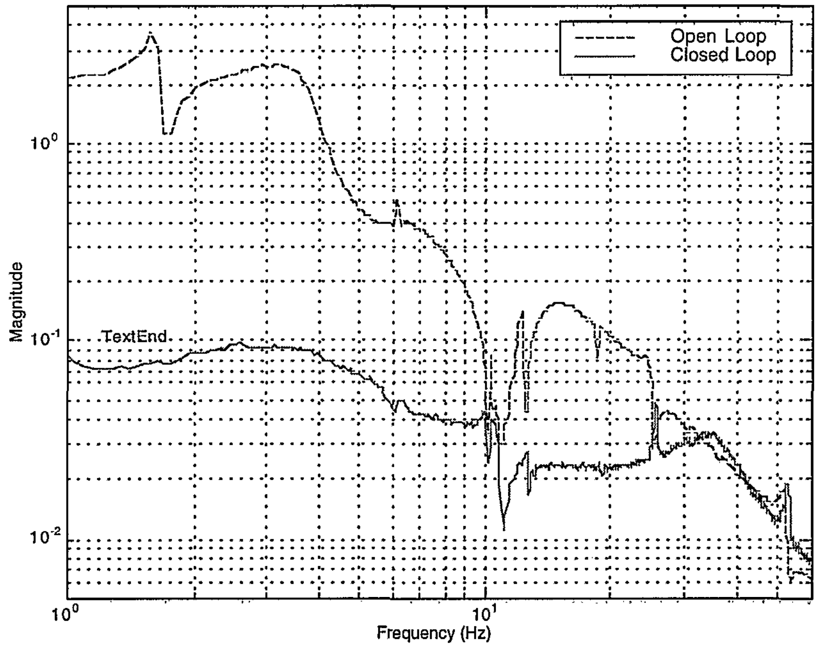

Figure 3. PGXt to X LOS Open and Closed Loop Experimental Frequency Responses - Classical Design

HYLAND

ARMArkov Information Filter + N-Step Predictive Control

In this design an ARMArkov or N-step ahead predictive model is identified using a neural net implementing the Information Filter Algorithm (IFA). IFA is a modified backpropagation method that involves the inverse correlation matrix of the neural net input. It can greatly increase the convergence speed of a neural network by normalizing the statistics of the input vector.

The neural net was remarkably noise insensitive, so a pure ARMA (i.e. one step ahead) model was used in the experimental runs. For preliminary work a SISO model from PGZt to PGRGZ was identified first. From simulation based on this model an ARMA order of 7 was found to work well in this case.

Figure 4 shows the absolute value of the neural net error as a function of time step for one of the ground tests. The neural net error is the difference between the actual measurement and the value predicted by the neural net. As can be seen from the figure, the error changes rapidly over the first 30 time steps, but then levels off at an RMS value of 0.0041. The remarkable fact here is that the RMS of the sensor noise is 0.0035. The neural net implementing IFA has identified the model to the noise floor. And further, it has done so in approximately 30 time steps = 60 ms.

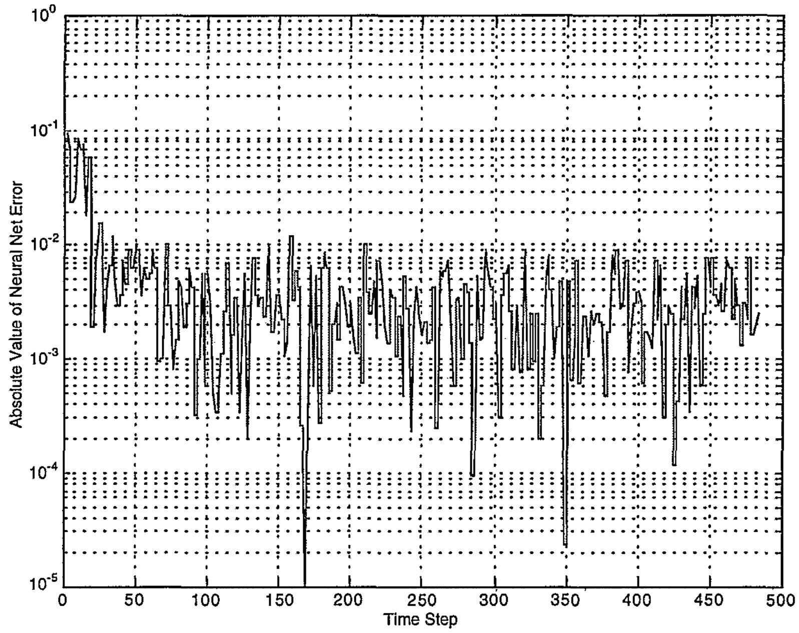

Figure 4. ARMA ID using IFA – Neural Network Error

Unfortunately, the ARMA model identified has non-minimum phase. In an attempt to compensate for a possible pure delay, ARMArkov models incorporating a delay were identified and they also proved to be non-minimum phase. Therefore, none of the models could be used in a deadbeat control scheme. As a result, this control approach had to be abandoned, even though the identification part of it worked beyond expectation.

Hebbian Learning with Classical Framework General Approach

In this approach, a neural net using a Hebbian learning rule is used to optimize the parameters of a fixed form classical controller. Essentially, the neural net generates controller parameter vectors and, based on a resulting cost imposed by the environment, generates new parameter vectors or subvectors thereof that aim to minimize the aforementioned cost. This approach can be considered an engineering implementation of a learning approach recently suggested by Vidyasagar.[9]

With one exception to be discussed momentarily, the neural net does not require training sets or, more generally, any supervised learning.

The fixed form classical controller used is the one that was already developed for the original classical design, namely an overall lead-lag filter, two bandpass with double notch loops and a straight gain loop.

HYLAND

However the lead-lag filter is fixed, leaving the neural net to optimize over the remaining 21 parameters.

As may be inferred from the previous comments, the neural net controller generation proceeds in stages, updating only a subvector of the control parameter vector each step. First the first bandpass is designed with all the other parameters remaining fixed. Then the first double notch is designed, followed by a redesign of the first bandpass to take advantage of the increased gain margin afforded by the double notch. The neural net repeats this procedure for the second bandpass/double notch loop and finally chooses a gain for the straight gain loop. Each of these individual designs is termed a design subcycle or more simply a subcycle. A design cycle is composed of, in this case, these 7 subcycles (bandpass 1, double notch 1, bandpass 1 again, bandpass 2, double notch 2, bandpass 2 again and overall gain). The design cycles continue until either the Z LOS error falls beneath a tolerance or a specified number of design cycles has occurred.

The cost, or performance metric, varies depending on which part of the template is being optimized. For bandpass filters and the straight gain, the cost is the sum squared of PGRGZ, which is the total Z LOS error, over a specified time window. For the double notches, the cost is the inverse gain margin (recall, the cost is to be minimized).

For the total Z LOS error, if at any time during summing process it exceeds a 'best yet' value for that design subcycle, the controller reverts to this 'best yet' controller and control optimization proceeds.

Returning to the exception noted earlier with regard to the unsupervised learning of the neural net, there is no guarantee that the parameter vector generated by the neural net will result in a closed loop stable system. Therefore, a Stability Margin Evaluator (SME) checks the candidate controller resulting from the just generated parameter vector and evaluates, online, a modified Nyquist stability criterion *before* the candidate controller is implemented. Closed loop unstable designs are simply discarded. The SME acts as a supervisor of the neural net.

The SME has two parts: an online open loop identification module and a gain margin calculation/stability check module. The identification module uses a segmented cross-spectral power method. A sum of sinusoids with frequencies at a subset of the bin frequencies of the soon to be done FFT is fed into the plant. After enough data has been collected, the coherence is calculated as a check that the resulting frequency response will be accurate. If the coherence meets a tolerance, the module cycles to the next 'comb' of frequencies and repeats the process until the frequency response has been identified at all the bin frequencies. If the coherence does not meet the tolerance, the sum of sinusoids is increased in magnitude and the process repeated until either the coherence is acceptable or the measurement or control saturates. In the case of saturation, the sum of sinusoids is reduced in magnitude until they no longer saturate, the frequency response is calculated and the SME steps to the next comb. Finally, the SME also estimates the uncertainty bounds in the identified frequency response and these high and low estimates are used in the gain margin/stability check module.

The gain margin/stability check module performs the following tasks. It first checks to see if any points of the loop frequency response fall within a 'lockout region' about the –1 point. This conservative estimate provides a quick and easy way of evaluating closed loop stability. Further, the module estimates any negative real axis crossovers that the lockout region check may have missed due to bin frequency spacing. If the crossovers occur outside the lockout region or occur not at all, the controller is stable and the gain margin is readily calculable. If the crossovers occur in the lockout region, the controller is discarded. This series of checks is done for the two loop frequency responses resulting from using the high and low estimates of the open loop frequency response. Both must pass the test. The gain margin is taken as the smaller of the two estimates.

Experimental Results

To-date, issues relating to the appropriate algorithmic logic to secure satisfactory coherence when using the open loop ID portion of the SME are still being investigated. Therefore, in the following we discuss detailed results for the gain margin evaluation portion of the SME.

To test the gain margin/stability check part of the SME and the Hebbian control optimization while allowing for further development of the online ID software, an open loop model of the plant (generated from an offline ID) was loaded into the algorithm.

For the results to be presented, each design subcycle (i.e. the design of a particular control parameter subvector) consisted of 10 iterations with the neural net allowed to vary the parameters of an initializing controller by approximately +/-20%. An initializing controller is not necessary, but it is necessary to specify a range over which the neural net may vary the control parameter vector. Favorable results may be obtained with quite wide ranges, although a greater number of design cycles is usually necessary. In this case the total number of overall design cycles was set to 10 also. For the Z LOS error

HYLAND

cost, the PGRGZ measurement was squared and summed for up to 200 time steps.

With these parameters, the control optimization took just under 3 minutes to run to completion. First the stability results will be discussed and then optimization results.

In short, during the run no instabilities were observed (during algorithm development a few unstable controllers were tried and so the authors had some idea of how a gross instability looked). To put this on a more quantitative basis, consider that an unstable closed loop will quickly drive the Z LOS error high, causing the control to revert the 'best yet' controller. By looking at when this reversion occurred for all the control parameter vectors implemented (i.e. all those deemed closed loop stable by the SME), an estimate of the number of possible wrong decisions by the SME may be made. The only problem with this reasoning is that as a software artifact, those control parameter vectors with the gain margin as their cost were assigned a revision time of 1. This assignment masks any Z LOS cost based revision times of 1 since a couple hundred gain margin based control parameter vectors were generated. However, those control parameter vectors with revision times of between 2 and approximately 10-15 remain suspect.

Shown in Figure 5 is a histogram detailing the number of summing steps actually taken for each implemented control parameter vector with a Z LOS error based cost. A value less than 200 indicates that the Z LOS error for that control parameter vector exceeded a previous 'best yet' Z LOS error for its design subcycle. A value of 200 indicates that the control parameter vector was either the first closed loop stable one generated in its design subcycle or it did better than all preceding control parameter vectors in that design subcycle.

For an unstable design, the Z LOS error would grow precipitously with a correspondingly short summing time. The value of 1 time step is excluded for the reasons detailed in the preceding paragraphs.

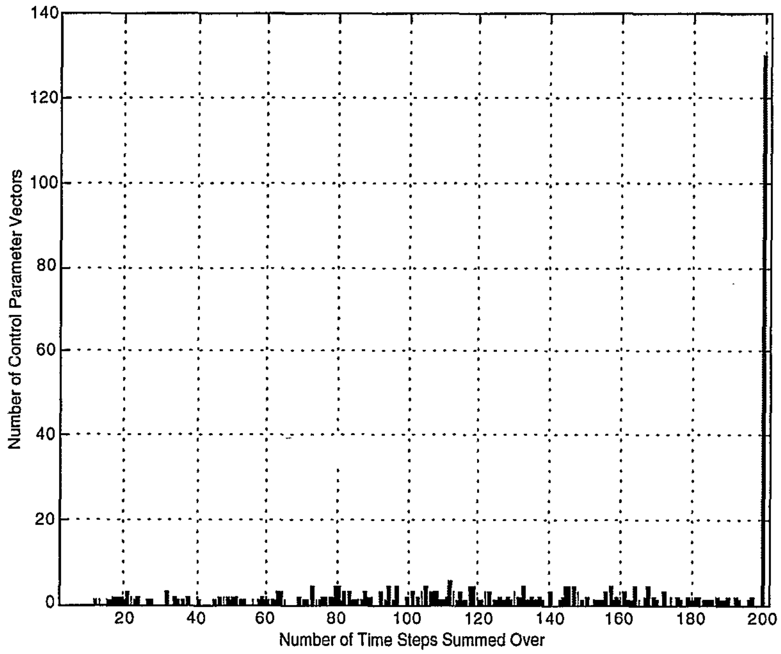

Figure 5. Summing Times > 1 for Z LOS Error Based Control Parameter Vectors

As can be seen from Figure 5, there are only 2 control parameter vectors with summing times less than 15 time steps. From subsequent runs, the average number of control parameter vectors with less than 15 summing time steps is 4.25. Considering that in the same runs there were on average 392 Z LOS error based control parameter vectors implemented during the 10 design cycles, this is only an approximate maximum possible false stable decision rate of 1%. Further, the SME only required a 1dB gain margin and 5 degree phase margin. By increasing these values, this maximum possible rate can be further reduced.

This approximate analysis substantiates, but of course does not prove, the claim than no destabilizing controllers were implemented.

Assured that the gain margin/stability check module of the SME functions properly, now consider the performance of the Hebbian optimization. Shown in

HYLAND

Figure 6 is a comparison of the open loop and the initial closed loop power spectral densities (PSDs) of the PGRGZ measurement with a uniform white noise disturbance. The initial controller was used to initialize the Hebbian optimization. The initial controller attenuates well at low frequencies at the cost of increasing the signal at higher frequencies. The design in not meant to be spectacular, though in fact, it does reduce the power in PGRGZ by a little under a half. If considerable more design time were to be spent, perhaps a better tradeoff between attenuating low frequency and exciting high frequency modes could be found.

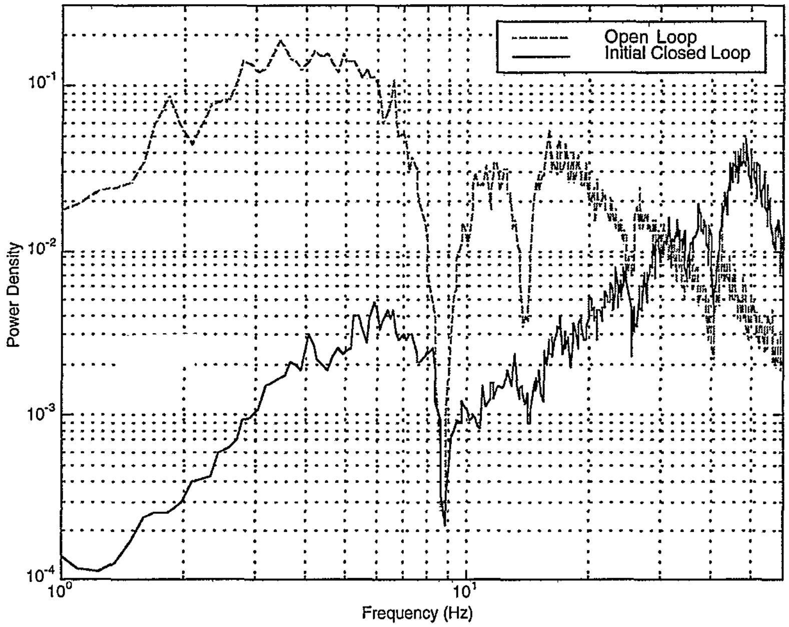

Figure 6. Open and Initial Closed Loop PSDs

Figure 7 compares the initial closed loop PSD and the closed loop PSD resulting from the controller after 10 Hebbian design cycles. From 1 Hz out to approximately 43 Hz, the final closed loop PSD is equal or greater than the initial controller. However, the reduction of the peak at 48 Hz makes all the difference. Note that this peak is almost an order of magnitude greater than the rest of the PSD.

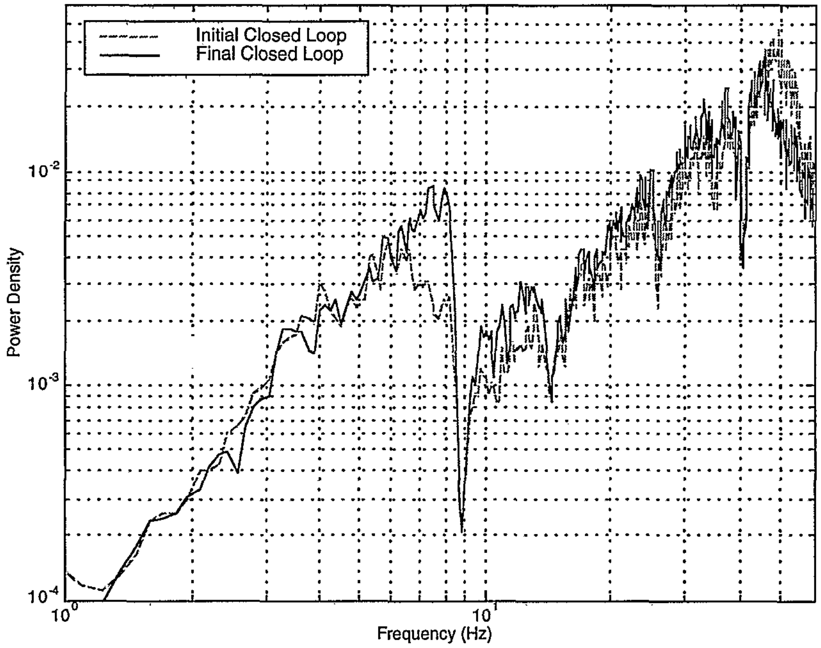

Figure 7. Initial and Final Closed Loop PSDs

HYLAND

To better illustrate the improvement that the Hebbian algorithm achieved, consider Table 1 in which the power in each signal (the area under the curves) is summarized.

Table 1. Signal Power Comparisons

	Power 1-60 Hz	% Improvement over Previous
Open Loop	10.11	-
Initial Closed	5.39	46.7
Final Closed	4.86	9.8

The final Hebbian controller has improved the initial control design by almost 10% in only 3 minutes. The Hebbian optimizer has found a better trade-off between low and high frequency attenuation.

Conclusion

A number of candidate controllers have been presented for the MACE II flight experiment as well as the results so far gathered from ground tests.

The classical controllers provide a baseline from which to evaluate the adaptive controllers. In addition, their ground performance compares favorably to controllers designed for MACE I while having the advantage of a smaller order.

Unfortunately, while the neural identification using the Information Filter Algorithm performed splendidly, identifying a 7th order ARMA model to the noise floor in 60 ms, the resulting model was not usable in a deadbeat control scheme. As a result, this approach had to be abandoned.

The Hebbian Learning algorithm worked extremely well, improving its initializing controller by 10% in 3 minutes. While work continues on the online identification feature of the Stability Margin Evaluator, the simplified Nyquist Stability Criterion and the gain margin calculation aspects of the SME function as desired. An approximate maximum possible false stable decision rate of 1% was derived empirically. This rate is too high for actual practice, but it is believed to be a very conservative estimate.

Of particular note is that the Hebbian Learning results demonstrate an algorithmically simple yet highly effective and reliable approach for self-reliant optimization of complex control systems. A key element ensuring reliability is that classical stability margins are treated as on-line performance evaluation criteria, on an equal footing with the customary mean square performance measures. Work continues to refine these approaches and to extend the basic concepts to nonlinear systems.

References

1. Miller, D.W., Crawley, E.F., How, J.P., Liu, K., Campbell, M.E., Grocott, S.C.O., Glaese, R.M. and Tuttle,T.D., "The Middeck Active Control Experiment (MACE): Summary Report", *MIT Space Engineering Research Center Report,* SERC #7-96, JUNE 1996.

2. Denoyer, K.K., Hyland, D.C., Davis, L.D., and Miller, D.W., "MACE II: A Space Shuttle Experiment for Investigating Adaptive Control of Flexible Spacecraft", AIAA-98-4319, *AIAA Guidance, Navigation, and Control Conference,* Boston, MA, August 10-12, 1998.

3. Narendra, K.S., and Parthasarathy, K. "Identification and control of dynamical systems using neural networks", *IEEE Transactions on Neural Networks*, 1, pp.4-27., 1990.

4. Haykin, S., *Neural Networks: A Comprehensive Foundation*, Prentice-Hall, New Jersey, 1994.

5. Phillips, D.J., Hyland, D. C., Collins, E.G., and King, J.A., "The Multi-Hex Prototype Experiment", *Proc. IEEE Conf. Dec. Contr.* , Brighton, U.K., pp. 2024-2029, December 1991.

6. Hyland, D.C., and Denoyer, K.K., "An Information Filter Approach to Rapid System Identification: Convergence Speed and Noise Sensitivity", *Proceedings of the AIAA Guidance, Navigation and Control Conference* , Boston, MA, August 10-12, 1998.

7. Astrom, K.J., and Wittenmark, B., *Adaptive Control*, Addison-Wesley, 1995, pp. 164-168.

8. Grocott, S., How, J., Miller, D., MacMartin, D.,Liu, K.,"Robust Control Design and Implementation on the Middeck Active Control Experiment," *Journal of Guidance, Control, and Dynamics* , Vol. 17, No.6, pp,1163-1170, 1994.

9. Vidyasagar, M., *A Theory of Learning and Generalization,* Springer-Verlag, London, 1997

HYLAND