Proceeding of the 2004 American Control Conference
Boston, Massachusetts June 30 - July 2, 2004

Numerical Solution for Multivariable Idle Speed Control of a Lean Burn Natural Gas Engine

Arvind Sivasubramanian, Peter H. Meckl

***Abstract*— This paper proposes a numerical solution to the idle speed and air-fuel ratio control of a lean-burn natural gas engine. Since more than one variable is involved (speed and air-fuel ratio), multivariable control techniques are used. A new gain tuning method is devised for the control gain tuning of a multivariable non-decoupling PI controller. For a disturbance rejection problem, this type of controller improves the performance of one variable by compromising the other. The proposed tuning method also makes it possible for the designer to choose between the amount and type of trade offs in performance between variables. The proposed method uses an expert systems approach to numerically find the optimum control gains needed for the closed loop system to meet the specified performance level.**

I. Introduction

THE automobile industry is a fast growing industry with many resources for research and development. Automobile engines form a major area of research due to their complexity and possibilities for further improvement. Gasoline and diesel engines have been used traditionally for many years. High costs of these fuels and pollution concerns led to the development of engines that use unconventional fuels like natural gas and electricity (Hybrid engines). Any engine has variables such as speed, air-fuel ratio, torque, etc that should be controlled so that they are always within desired levels. Control techniques can be used to achieve set-point tracking or disturbance rejection. Cruise control is an example of set-point tracking where the engine speed should stay close to the set value. Reducing speed undershoots during torque loads at idle is an example of disturbance rejection.

The variable being controlled in both of the above examples is speed. But in reality there is more than one variable to control. For example, designing a controller solely based on speed might affect the air-fuel ratio which in turn affects the emissions level. On the other hand, a controller meant to keep the emissions level in check will not guarantee good speed tracking (or speed recovery in case of a disturbance rejection problem). This calls for control techniques that control both speed and air-fuel ratio. Multivariable control techniques solve this problem since they allow control over more than one variable. Two types of multivariable controllers are decoupling and non-decoupling controllers. Decoupling controllers are ideal for reference tracking problems where a change in one input should not affect the other output. Non-decoupling controllers are ideal for disturbance rejection problems where there are no reference inputs.

Tuning of multivariable PI controllers has been studied by many researchers in the past. The motivation behind this work was to develop a Ziegler-Nichols type tuning rule for the multivariable case. Davison [1] was the first to try tuning of a multivariable PI controller. The controller was designed for a servo mechanism. Later his techniques were extended to tune a multivariable PI controller for a continuous plant by Penttinen and Koivo [2] where the controller decouples the plant at high frequencies. Peltomaa and Koivo [3] then tuned a multivariable PI controller for a discrete plant. In the controller proposed by [3], the proportional and integral gain matrices were designed such that they decouple the system at high and low frequencies, respectively. Porter [4] developed a complete decoupling controller for a set-point tracking problem. A better control is obtained by a decoupling controller because in a set-point tracking problem, it is not desirable to have changes in one channel affect the other. Maciejowski [5] extended the work by Penttinen and Koivo [2] to decouple the plant at the desired bandwidth instead of at high frequencies. Tanttu et al. [6] experimentally compared some of the tuning techniques mentioned above. Menani and Koivo [7] combined multivariable tuning with adaptive relay feedback to identify the system model online at any desired frequency. Gangopadhyay and Meckl [8] proposed a multivariable PI controller that does not decouple the plant at high frequencies. Instead, the loop interactions were used to reject disturbances more effectively. Thus a non-decoupling controller would be more effective in rejecting disturbances than a decoupling controller. They also proposed stability ranges for the scalar gains μ and δ, which would decide K_p and K_I. This paper is based on the work done in [8] and extends that work to suggest some

Manuscript received March 12, 2004.

Arvind Sivasubramanian is a graduate student in Mechanical Engineering, Purdue University, West Lafayette, IN 47907 USA (e-mail: sivasuba@purdue.edu)

Peter H. Meckl is an associate professor in Mechanical Engineering, Purdue University, West Lafayette, IN 47907 USA (phone: 765-494-5686; e-mail: meckl@ecn.purdue.edu).

guidelines for choosing μ and δ so that the designer has reasonable control over the amount and type of performance in each channel.

To achieve this goal, the loop interactions have to be understood and an analytical tool has to be identified that relates the performance in each channel to the control gains. Postlethwaite and MacFarlane [9] suggested a root locus for multivariable systems using a complex variable approach. It was identified that an $n \times n$ system could have a root locus with up to n branches at a given point on the real axis, overlapping each other. This is unlike the Single Input Single Output (SISO) case where there can be only one branch on the real axis at a given point. Yagle and Levy [10] came up with an easy method to identify the number and location of the different branches on the real axis for a multivariable system. However, both of these works fail to distinguish between channels, although they give the location of all the poles for all possible gains.

On the frequency domain side, Rosenbrock [11] suggested using Gershgorin circles to analyze the loop interactions in multivariable systems. He also suggested a modified Nyquist criterion, called the Nyquist Array, for multivariable systems that was based on Gershgorin circles. Recently, Chen and Seborg [12] tuned a multivariable PI controller based on Gershgorin bands. But since Gershgorin bands and Nyquist Array can be used only for a decoupled multivariable system, those approaches are not helpful in achieving the objective of this research. The plant cannot be decoupled to allow these methods to be used because that defeats the purpose of having a non-decoupling controller. Bryant and Yeung [13] used Nyquist Arrays to design controllers for multivariable systems by sequentially closing the loops. Yueng [14] proposed a sequential design procedure using root loci for multivariable systems. However, in both cases, the Nyquist Array or root locus obtained in one stage becomes invalid after the next loop is closed. Since all the analytical methods failed to help achieve the objective, a new method based on computer aided design and expert systems is proposed in this paper.

II. ENGINE MODEL

The system dealt with in this paper is a lean-burn natural gas engine [15]. The engine model was experimentally obtained for a natural gas engine installed on a school bus. The linearized model of a lean-burn natural gas engine around the idle operating point is used to design the new controller. The system is 2×2 with perturbed speed (δn) and air-fuel ratio (δL) as outputs and perturbed throttle ($\delta\alpha$) and perturbed fuel ($\delta \dot{m}_f$) as inputs. Linearized and discretized transfer functions between the throttle and engine speed and fuel and engine speed are given below. The individual transfer functions G_{ii} for a sampling time of 0.1 second are given by

$$G_{11}(z) = \frac{\delta n}{\delta\alpha} = \frac{4.07z^{-1} + 3.34z^{-2}}{1 - 1.494z^{-1} + 0.552z^{-2}} \tag{1}$$

$$G_{12}(z) = \frac{\delta n}{\delta \dot{m}_f} = \frac{-56.2z^{-1} - 30.18z^{-2} + 11.41z^{-3}}{1 - 1.43z^{-1} + 0.48z^{-2}} \tag{2}$$

$$G_{21}(z) = \frac{\delta L}{\delta\alpha} = \frac{0.917}{z - 0.545} \tag{3}$$

$$G_{22}(z) = \frac{\delta L}{\delta \dot{m}_f} = \frac{-14.28}{z - 0.537} \tag{4}$$

The transfer function between the external disturbance torque and speed is given by,

$$G_d(z) = \frac{0.489z^{-1} - 0.245z^{-2}}{1 - 1.494z^{-1} + 0.552z^{-2}} \tag{5}$$

The main control objective is to reduce the transient speed and air fuel ratio (a/f) fluctuations during the torque load disturbance. The complex nature of the cross coupling between the two variables, speed and air-fuel ratio, calls for the use of multivariable control techniques. A multivariable PI controller was used to achieve this. The objective of this paper is to obtain design rules for choosing the control gains so that the resulting closed loop system is stable and conforms to the required performance specifications.

III. CONTROLLER BACKGROUND

Consider the following discrete-time linear time-invariant open-loop stable plant:

$$x(k+1) = Ax(k) + Bu(k) + Wz(k) \tag{6}$$

$$y(k) = Cx(k) \tag{7}$$

$$e(k) = y_r - Cx(k) \tag{8}$$

where $x(k) \in R^n$ is the state vector; $u(k) \in R^m$ is the control input; $y(k) \in R^m$ is the measured output vector; $e(k) \in R^m$ is the error between the constant reference input vector y_r and the measurable output vector y; and $z \in R^q$ is a constant disturbance vector. The matrices A, B, C, and W are constant and have appropriate dimensions.

The control design for the problem uses three scalar gains, namely μ, δ and γ. These three gains, along with matrices A, B and C, determine the K_p and K_I control gain matrices. The dependence of K_p and K_I on μ and δ can be given by the following relations:

$$K_P = \mu K_P^* \tag{9}$$

$$K_p^* = B^+(\gamma I + A)C^+ \tag{10}$$

$$A_0 = A - BK_pC \tag{11}$$

$$Q(z) = C(zI - A_0)^{-1}B \tag{12}$$

$$K_I = \delta Q^{-1}(1) \tag{13}$$

where the superscript $(+)$ indicates a pseudo-inverse of a matrix in the sense of Moore-Penrose and A_0 is the closed loop system matrix.

The guidelines for choosing μ and δ are as follows,

$$0 < \mu < \frac{2}{\gamma + 1 + \|E\|} \approx 1 \tag{14}$$

$$0 < \delta < \frac{1}{5T} \tag{15}$$

where T is the sampling time and $E = A - BK_p^* C + \gamma I$. For perfect disturbance rejection, $\left(A - BK_P^* C\right)$ should be equal to $-I$. This is not possible. However, the effect of disturbance can be minimized by choosing the proportional gain matrix K_p^* such that the norm of E is minimized. Hence for better disturbance rejection,

$$\gamma \approx 0.9 \tag{16}$$

Choosing the gains according to the above guidelines just ensures the stability of the system. It does not say anything about the performance of the system. This paper will provide guidelines for selecting these three gains based on settling time and percentage overshoot. The three guidelines above, along with the new guidelines, will ensure that the system will not only be stable, but also meet the required performance criteria.

The approach of this paper is as follows: γ should be as close to unity as possible for better disturbance rejection and hence is fixed at 0.9. K_p and K_I matrices depend on μ and δ. Choosing two scalar gains, μ and δ, to ensure stability and performance simplifies the design of a multivariable PI controller. This is very similar to a SISO problem where the control gains are scalars instead of matrices. A PI controller for a highly coupled multivariable system can thus be designed by choosing just two scalar gains as in a SISO case. Since the controller for which μ and δ are tuned is a non-decoupling controller, it is possible to improve the performance in one channel with the help of a compromise of performance in the other. Although extensive work has been done in multivariable PI tuning, none of it concentrated on creating a provision by which the designer can trade off between the amounts of performance gain in one channel to the performance loss in the other. This will be attempted in this paper by using relative weights to choose between channels and between types of performance (settling time and overshoot).

IV. NEW TUNING METHOD

For the particular system under study, the gain ranges for stability were determined by [8] and found to be

$$\gamma \approx 0.9 \tag{17}$$

$$0 < \delta < 2 \tag{18}$$

$$0 < \mu < 1 \tag{19}$$

One particular combination of μ and δ will satisfy the required performance criteria and the goal would be to find that combination.

A. Performance Criteria

The performance criteria, settling time and percentage overshoot, are difficult to define for a disturbance rejection problem where the steady state value is zero. So, two other measures called the Integral Time Absolute Error (ITAE) and the Integral Square Error (ISE) were chosen:

$$ITAE = \int_0^T t\left|e(t)\right| dt \tag{20}$$

$$ISE = \int_0^T e(t)^2\, dt \tag{21}$$

Here, $e(t)$ is the error signal between the desired and the actual output, and T is the total simulation time in seconds. $ITAE$ penalizes large settling times and will be used as a measure of settling time. Although ISE penalizes all overshoots, not just the maximum overshoot, it is still a good measure of the transient behavior of the system.

B. Expert Systems Approach

Since the model of the plant is available, simulations can be done for all possible gains to find the combination for which $ITAE$ and ISE for both channels are small. But running simulations for all possible gains would take a lot of computation time. An expert system-like approach can be used instead to arrive at the optimum value with much less computation time. This program starts at an arbitrary combination of the control gains and arrives at the combination that yields the best performance with the shortest path through the μ - δ grid.

C. Relative Weights

Through extensive simulations it was found that the best $ITAE$ for speed, the best $ITAE$ for air-fuel ratio, the best ISE for speed and the best ISE for air-fuel ratio all occur at different gain combinations. But only one set of gains can be chosen at the end. So relative importance factors, or weights, will be used to choose the relative importance between the two channels and between the two parameters. The user would be able to choose the importance factor between speed and air-fuel ratio channels, and between $ITAE$ and ISE. Both weights would be from 0 to 1. The program would then return a single combination of μ and δ for which the weighted performance criteria are met on both channels based on their respective importance factors.

The final gain combination is achieved by using user defined weights in two stages. Since all the four values are of different magnitudes, they are normalized by dividing each of them by their respective maximum values. In the first stage, the user chooses a relative weight for the first channel as given below:

$$itae = wt1 \times itae1 + (1 - wt1) \times itae2 \quad (22)$$

$$ise = wt1 \times ise1 + (1 - wt1) \times ise2 \quad (23)$$

Here, $wt1$ is the user defined relative weight for the first channel (speed). $itae1$ and $ise1$ are the $ITAE$ and ISE values of speed, respectively. $itae2$ and $ise2$ are the $ITAE$ and ISE values of air-fuel ratio, respectively. $itae$ and ise are the weighted values of $ITAE$ and ISE, respectively, the weighting being based on the relative importance of channels ($wt1$). For example, if speed is four times more important than air-fuel ratio, then $wt1 = 0.8$. Then $itae$ will be the value of $ITAE$ on the μ - δ grid such that $ITAE$ of speed is given 80% importance. If $wt1 = 1$, then $itae$ and $itae1$ will occupy the same point on the μ - δ grid.

In the second stage of weighting, the user chooses a weight $wt2$, which is the relative importance of $ITAE$ over ISE. This is given by:

$$optimalvalue = wt2 \times itae + (1 - wt2) \times ise \quad (24)$$

For example, if the user chooses weights $wt1 = 1$ and $wt2 = 1$, then the optimal value and $itae1$ occupy the same point on the μ - δ grid. This two stage weighting presents a lot of flexibility to the designer. Since the performance of one channel affects the other, the effect of an undesirable channel over a desirable one can be minimized by choosing appropriate weights.

D. The Code

MATLAB code was written to implement the new tuning method. When the code is invoked, the user will be prompted to enter the system transfer function matrices with the option of using the default plant in [8]. The program then automatically calculates the ranges of gains for which the system is stable. γ is assumed to be 0.9 in all cases. Once the ranges of gains for μ and δ are known, the program forms a μ - δ grid with a value of zero for all the grid points initially. The number of grid points is determined by the resolution for μ and δ, which is determined by the user based on the speed of the processor. The center point of the μ - δ grid is then calculated and assumed to be (i, j). The values of $ITAE$ and ISE are calculated at this point on the grid. These values are then weighted according to the weights specified by the user and the resulting optimum value replaces zero at the (i, j) point, also called the centroid. The centroid, along with the eight points around it, forms a square and is called a template. The optimum values at all the eight points, excluding the centroid, on the template are calculated and compared with the optimum value at the centroid. The location of the smallest number is found and the centroid is moved to that point. This forms the centroid of a new template. The optimum values are calculated at the eight points around the new centroid. Of the eight points on the new template, only the new points are calculated. Any points on the new template that were on the old one are just taken from the knowledge base to reduce computation time. The minimum point on the new template is identified and the centroid is redefined at this point. This process goes on until the minimum point on the entire μ - δ grid is determined. The corresponding values of μ and δ would be the optimum gain combination to achieve the required performance specification.

E. Extension to Other Systems

The code is designed such that it can be extended to other 2 input – 2 output systems apart from the default plant in [8]. With slight modification of the code, it can be extended to higher order Multi Input – Multi Output (MIMO) systems and more easily to SISO systems. This adaptability to other 2 input – 2 output systems is made possible due to the fact that a MIMO PI controller for any such system can be designed by the choice of just two values, namely μ and δ. The only change from the default system would be the actual range of μ and δ within which the system is stable. All the computations are to be made only in the stable region. These ranges are given by (14) and (15). It is clear that the value of μ for any system would be between 0 and 1, and that of δ would be a function of the sampling time. Thus for any system, the ranges of μ and δ in which the system is stable can be determined. Once the ranges are known, then the logic to find the optimum values of μ and δ is the same as in the default case.

V. SIMULATION RESULTS

This section contains the simulation results for the natural gas engine model and also highlights the advantages of this new approach to multivariable PI gain tuning. The engine model is given by (1), (2), (3) and (4). The sampling time used was 0.1 seconds. All simulations were done for a disturbance rejection problem with a torque disturbance acting on the engine. The disturbance transfer function is given by (5). A disturbance step of 5 Nm (3.7 ft-lbf) was applied. This is equivalent to applying an external torque of 5 Nm to the engine.

First, simulations were run for weights of 99% and 1% for engine speed and the responses of both variables were compared. The weights between $ITAE$ and ISE were maintained at 50% each for easier comparison. As shown in Figure 1, when speed had a preference for 99%, the speed under-shoot was around 3.1 rpm and settling time was almost 3 seconds. When the preference for speed was reduced to 1%, the speed under-shoot increased from 3.1 rpm to 3.8 rpm and the settling time remained the same.

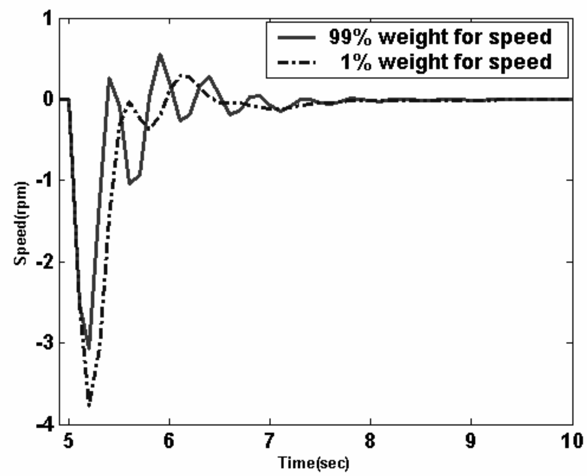

Fig. 1. Simulation results for speed with different $wt1$ (solid line – 99% weight for speed, dashed line- 1% weight for air-fuel ratio).

In these examples, 1% preference for speed means 99% preference for air-fuel ratio. All settling times were taken as the time at which the response is within 0.02 units of the steady state value (± 0.02 rpm for speed and ± 0.02 for air-fuel ratio). So with a loss of performance in the speed (with 1% preference), it can be expected that there will be better performance for the air-fuel ratio. This is verified in Figure 2.

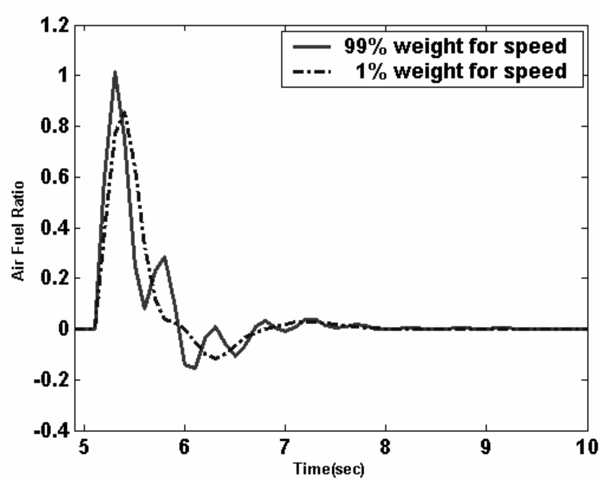

Fig. 2. Simulation results for air-fuel ratio with different $wt1$ (solid line – 99% weight for speed, dashed line- 1% weight for air-fuel ratio).

Next, simulations were done for different weights for $ITAE$ and ISE. The weights between the variables were held at 50% each for better comparison. Simulations were first done for 99% preference for $ITAE$ and then with 1% for the same. This is shown in Figure 3.

For an $ITAE$ preference of 99%, speed has a settling time of 2.5 seconds and an under-shoot of 3.2 rpm. When the weight was reduced to 1%, the settling time increased to 3.5 seconds. This increase in settling time can be

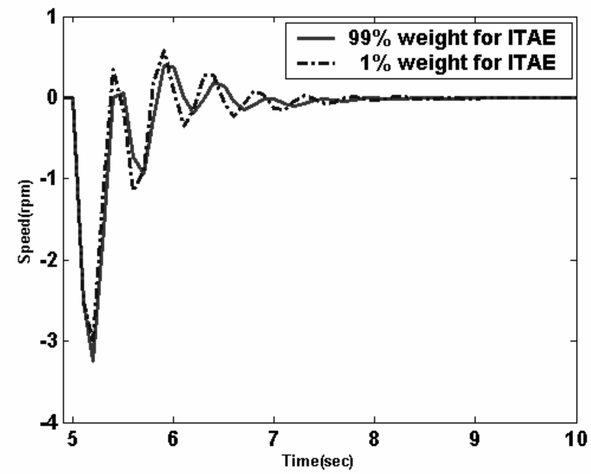

Fig. 3. Simulation results for speed with different $wt2$ (solid line – 99% weight for $ITAE$, dashed line- 1% weight for $ITAE$).

expected to cause an improvement in under-shoot because when the $ITAE$ preference goes from 99% to 1%, the ISE preference increases from 1% to 99%. This is indeed true, as shown in Figure 3, where the under-shoot improved to 3 rpm from 3.2 rpm. In other words, the improvement in under-shoot is at the expense of settling time.

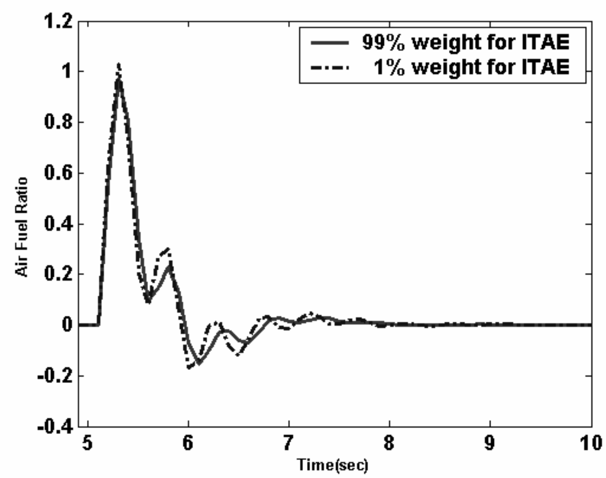

Fig. 4. Simulation results for air-fuel ratio with different $wt2$ (solid line – 99% weight for $ITAE$, dashed line- 1% weight for $ITAE$).

Figure 4 shows the changes in air-fuel ratio response for different weights between $ITAE$ and ISE. Similar trends as compared to Figure 3 can be seen here. As the $ITAE$ preference goes from 99% to 1%, the settling time increases from 2.4 seconds to 2.6 seconds. At the same time however, the overshoot also increases from 0.98 to 1.02. This is because the controller was designed such that better speed recovery is achieved by a higher air-fuel ratio overshoot. It was seen in Figure 1 that when the weight for the speed variable was increased, the speed under-shoot improved and the settling time remained the same. However, if there is need for an improvement in performance through decrease in settling time, then it is still possible with a higher weight for *ITAE* . Figure 5 supports the above argument. When the weight for *ITAE* was increased from 50% to 99%, the settling time decreased from 3.1 seconds to 2.8 seconds. The under-shoot increased to 3.25 rpm from 3.05 rpm. Hence an enhanced performance was achieved through better settling time, rather than better under-shoot as in Figure 1.

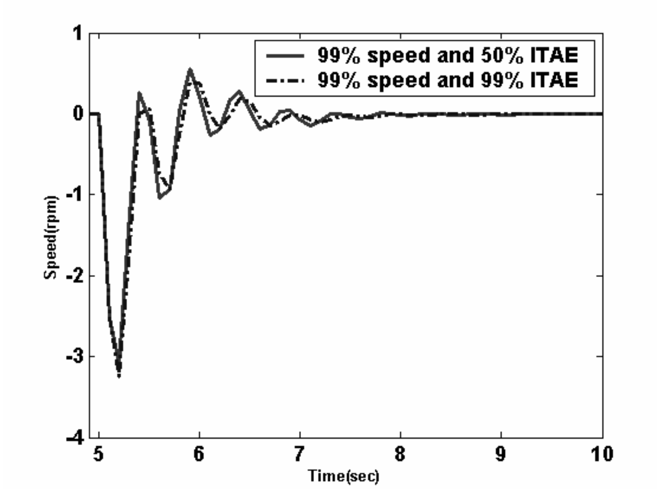

Fig. 5. Simulation results for speed with changes in both the weights (solid line- 99% speed and 50% *ITAE* , dashed line- 99% speed and 99% *ITAE*).

VI. CONCLUSION

This research develops a new multivariable PI controller for the idle speed control of a lean burn natural gas engine. The controller gains were chosen by a new gain tuning method that arrives at an optimum gain pair numerically using an expert system like approach. This tuning method gives flexibility to the user to choose the amount and type of performance that is desired for each engine variable (speed and air-fuel ratio). Simulation results show that the amount and type of performance of each variable changes for different weights. Since the performance of one variable affects the other, the effect of an undesirable variable over a desirable one can be minimized by choosing appropriate weights.

REFERENCES

[1] Davison, E.J., 1976, "Multivariable tuning regulators: the feedforward and robust control of a general servomechanism problem," *IEEE Transactions on Automatic Control*, AC-21, pp. 35-47.

[2] Penttinen, J., and Koivo, H. N., 1980, "Multivariable tuning regulators for unknown systems," *Automatica*, Vol. 16, pp. 393-398.

[3] Peltomaa, A., and Koivo, H. N., 1983, "Tuning of a multivariable discrete time PI controller for unknown systems," *International Journal of Control*, Vol. 38, pp. 735-745.

[4] Porter, B., 1982, "Design of tunable set-point tracking controllers for linear multivariable plants," *International Journal of Control*, Vol. 35, pp. 1107-1115.

[5] Maciejowski, J. M., 1989, *Multivariable Feedback Design*, Wokingham, Addison-Wesley, England.

[6] Tanttu, J.T., Cameron, F., and Lisitzin, H., 1991, "Experimental comparison of some multivariable PI controller tuning methods," *IECON Proceedings*, Vol. 3, pp. 1818-1823.

[7] Menani, A., and Koivo, H. N., 1996, "Automatic tuning of multivariable controllers with adaptive relay feedback," *Proceedings of the 35th IEEE Conference on Decision and Control*, Kobe, Japan, pp. 4695-4700.

[8] Gangopadhyay, A., and Meckl, P., 2001, "Multivariable PI tuning for disturbance rejection and application to engine idle speed control simulation," *International Journal of Control*, Vol. 74, No. 10, pp. 1033-1041.

[9] Postlethwaite, I., and MacFarlane, A. G. J., 1979, *A Complex Variable Approach to the Analysis of Linear Multivariable Feedback Systems*, Springer-Verlag.

[10] Yagle, A. E., and Levy, B.C.,1981, *Multivariable Root Loci on the Real Axis*, Laboratory for Information and Decision Systems and the Department of Electrical Engineering and Computer Science, MIT., Cambridge, MA 02139.

[11] Rosenbrock, H. H., 1974, *Computer-aided Control System Design*, Academic press.

[12] Chen, D., and Seborg, D. E., 2002, "Multiloop PI/PID controller design based on Gershgorin bands," *IEE Proc. Control Theory Appl.*," Vol. 149, No. 1, pp. 68- 73.

[13] Bryant, G. F., and Yeung, L. F., 1994, "Sequential design procedures for multivariable control systems: a direct method based on Gauss-Jordon factorisation," ," *Proceedings of the 33rd IEEE Conference on Decision and Control* , Lake Buena Vista, FL, pp. 3630- 3631.

[14] Yeung, L. F., 1995, "Sequential design procedure for multivariable systems: The root loci approach," *Proceedings of the 34th IEEE Conference on Decision and Control*, New Orleans, LA, pp. 4114-4119.

[15] Gangopadhyay, A., and Peter H. Meckl., 2001, "Modelling and Validation of a Lean Burn Natural Gas Engine," *ASME Journal of Dynamic Systems, Measurement, and Control*, Vol. 123, No. 3, pp. 425-430.