An Adaptive Self Tuning Regulator with Genetic Controller Tuning

Adel A. Ghandakly

Department of Electrical and Computer Engineering, California State University, Chico, Chico CA 95929, USA
*Corresponding Author: aghandakly@csuchico.edu

Abstract This paper presents the development of a Genetic Algorithm Supervisory Loop (GASL) to enhance the performance of Adaptive Controllers in the Self Tuning Regulator (STR) framework. The GASL is designed to tune the identification model and control parameters to achieve an optimum performance of the process. The STR controller used to demonstrate the technique effectiveness is a real time Proportional plus Integral plus Derivative (PID) algorithm. The addition of the GASL to the PID STR provides superior performance by expanding the STR’s ability to maintain control of nonlinear systems under large and quick shifts in system modes and/or disturbances. In test cases, the basic STR shows performance deterioration, while the GASL enhanced STR maintained desirable performances. The GASL developed is applied to a nonlinear adaptive suspension system for investigation and assessment. Investigations results show the effectiveness of utilizing the GASL within the adaptive STR framework.

Keywords STR, Adaptive Control, Intelligent Control

1. Introduction

Systems with structural and performance complexities require sophisticated controller design techniques. Advances in Micro Controller Technology and Computer Architecture enhance the feasibility of those design techniques to operate in real time and provide some elements of learning and adaptation. Advances in technology led to highly complex processes that require control systems with tight specifications and high level of autonomy. Stationary (constant parameters) controllers are straightforward to design off line and perform well for systems that can be modelled adequately with stationary linear models. Those controllers lack adaptation and therefore they fail when such plant model becomes inadequate for systems with wide range of dynamics. Adaptive Controller (AC) techniques including those of the Gain Scheduling (GS), Self Tuning Regulator STR), Model Reference Adaptive Control (MRAC) can provide solutions to many such systems. Although AC can be effective for such systems, they in turn have limitations when changes in the system are drastic or multimodal in nature. Such limitations are mainly due to their required a priori and off line designer choice settings, and their lack of on line learning [1]. The need for expansion of current adaptive control techniques for complex processes has branched into the realm of intelligent control. Intelligent Adaptive Control (IAC) structures can present an effective approach to control complex and uncertain systems within stringent specifications. Such structures can range from utilizing simple fuzzy logic to complex neural networks [1-2]. Several intelligent control algorithms have been developed, including include Fuzzy Logic Control, Neural Networks, and Genetic Algorithms (GA). The design to be presented in this paper follows a GA based intelligent supervisory loop control method in an STR adaptive control framework with a PID controller structure [3]. An attractive quality of a GA is that it provides an optimization search without requiring a detailed system model for the mathematical derivatives that are inevitably required by other mathematical optimization algorithms [3-5]. The supervisory loop operates in parallel with the adaptive controller to expand its ability to maintain system control under large and quick shifts in system modes and/or disturbances. The developed technique is simulated for investigation and assessment by application to a nonlinear vehicle adaptive suspension system.

2. Design of the Adaptive Controller GA Supervisory Loop

The system structure is outlined in figure 1. The adaptive STR operates by a Recursive Least Squares Identification (RLSI) of the process by a set order difference equation model. The off line designer choice settings in this controller include the identification model order as well as the off line parameters of the controller structure used. The genetic algorithm tunes the identification model and control parameters to achieve an optimum performance of the
An Adaptive Self Tuning Regulator with Genetic Controller Tuning

Process as developed in the next section. The GA is developed as the intelligent supervisory loop for an adaptive STR with a real-time PID control structure as shown below in terms of the controller parameter polynomials T, R and S [6].

\[
u(k) = \frac{T(q^{-1})}{R(q^{-1})} y_r(k) - \frac{S(q^{-1})}{R(q^{-1})} y(k)
\]  

This expression will be satisfied if terms S, T and R are as follows:

\[
S(q^{-1}) = s_0 + s_1 q^{-1} + s_2 q^{-2}
\]

\[
T(q^{-1}) = -(s_0 + s_1 + s_2)
\]

\[
R(q^{-1}) = (1-q^{-1})(1+r q^{-1})
\]

Where

\[
s_0 = T_s K_f + \frac{K_d}{T_s} + K_p
\]

\[
s_1 = -\frac{2K_d}{T_s} - K_p + rK_p
\]

\[
s_2 = \frac{K_d}{T_s} - rK_p
\]

In the self-tuning PID algorithm, the process model is restricted to a 2nd order system given by

\[
y = \frac{B}{A} = \frac{b_1 q^{-1} + b_2 q^{-2}}{1 + a_1 q^{-1} + a_2 q^{-2}}
\]

Next, we can solve for the closed loop system.

\[
y = \frac{B T}{A R + B S}
\]

and

\[AR + BS = \left(1 + a_1 l q^{-1} + a_2 l^2 q^{-2}\right)(1+l q^{-1}) = 0\]

The latter polynomial identity is solved on line for the PID gains Kp, Ki, and Kd, and the GA then operates with a population set made up of several different values for \( l \). The following Recursive Least Square identification algorithm in terms of the model parameter vector \( \Theta \) and the recursion variables K and P, measurement set \( x(t) \), and the weighting factor \( \gamma \) is used [10] for developing the system model for both STR and GA computations.

\[
\Theta(t) = \Theta(t-1) + K(t)\left[\phi(t) - x'(t)\Theta(t-1)\right]
\]

\[
P(t) = \frac{\left[I - K(t)x'(t)\right]P(t-1)}{\gamma}
\]

\[
K(t) = \frac{P(t-1)x(t)}{\gamma + x'(t)P(t-1)x(t)}
\]

An iteration of the genetic process evaluates the PID gains for the specified \( l \). The genetic algorithm operates with a population set made up of several different values for \( l \). One iteration of the genetic process evaluates the PID gains for the specified \( l \). These gains are used to calculate the control signal for the next iteration. This control result is then evaluated in the process model that most accurately represents the system transfer function resulting from the
model matching genetic algorithm. Once this evaluation is complete, the predicted output is compared with the reference signal and the corresponding error represents the fitness value to be optimized or in this specific case minimized. The generated fitness values for each population point serve to select which population points produce the least amount of error between predicted output and reference. The population points are sorted according to their corresponding fitness values and the points providing better performance are selected as parents for the next generation. A new set of $\alpha$ is created through the genetic process and the algorithm repeats.

The PID gains are used to calculate the control signal for the next iteration. This control is then evaluated on the process model that most accurately represents the system transfer function resulting from the GA model matching. The predicted output is then compared with the reference signal and the corresponding error represents the fitness value to be minimized. The generated fitness values for each population point serve to select which population points produce the least amount of error between predicted output and reference. The population points are sorted according to their corresponding fitness values, and those with better performance are selected as parents for the next generation. A new set of $\alpha$ is created through the genetic process and the algorithm repeats.

Figure 2 shows the GA supervisory loop software flowchart with its components executed as follows:

1. Initialize loop parameters and initial populations, and measure the system output.
2. Conduct n-multiple order identifications and evaluate the identification error for the n model population.
3. The model with the least identification error is tagged as the best fit model. The exact model for fitness evaluation is obtained using the error between the multiple models and actual output as its fitness function. The exact model that the GA utilizes for the fitness evaluation is obtained through the use of a second genetic algorithm. The GA uses the error between the multiple identification models and actual output as its fitness function. The output of this GA is the order of identification to be used for the overall system performance.
4. Design and test the controller on the best model for all populations in the $\alpha$ parameter, and tag the parameter with the least output error.
5. Use best $\alpha$ to produce the system controller parameters.
6. For the purpose of feasibility of the genetic algorithm to be used online in an adaptive control scheme, the genetic supervisory loop is designed to utilize the digital system model obtained by the online identification rather than actual system model.

In the execution program of steps 2 and 4 above, the propagation of the best solution to the next generation is handled by limiting the reproduction to the best solutions and discarding the point that performed the worst. The population members are then sorted according to their respective ranks and only the best points are used for reproduction. Once crossover and mutation are conducted on these optimum points, the new population is created using the best solution along with the children created.

3. Application to an Adaptive Suspension System

To assess its effectiveness, the developed genetic STR PID has been applied to a nonlinear suspension system as described in figure 3. The system model is given by the state equations (15)-(17) given below [7], and the system parameters are shown in Table 1.

\[
\begin{bmatrix}
    Zs \\
    Zu \\
    \dot{Z}_s \\
    \dot{Z}_u
\end{bmatrix} =
\begin{bmatrix}
    0 & 0 & 1 & 0 \\
    0 & 0 & 0 & 0 \\
    -ks / ms & ks / ms & -bs / ms & bs / ms \\
    ks / mu & (ks + kr) / mu & bs / mu & -bs / mu
\end{bmatrix}
\begin{bmatrix}
    Zs \\
    Zu \\
    \dot{Z}_s \\
    \dot{Z}_u
\end{bmatrix}
\]

\[
\begin{bmatrix}
    0 \\
    0 \\
    1 / ms \\
    -1 / mu
\end{bmatrix} \cdot \begin{bmatrix}
    U
\end{bmatrix} +
\begin{bmatrix}
    0 & 0 & 0 & 0 \\
    -ks / ms & 0 & 0 & 0 \\
    ks / mu & kr / mu & 0 & 0
\end{bmatrix}
\begin{bmatrix}
    ns \\
    Zr
\end{bmatrix}
\]

\[
ns = (0.1ks * (Zs - Zu)^3 + 0.1bs * (Zs - Zu)^3 * sign(Zs - Zu)) / ks
\]

\[
U = ks * ns + u_{\text{adaptive}}
\]

And $u$ adaptive is the adaptive controller output, and $Zr$ is the disturbing reference.
Table 1. Suspension System Data

| ms   | μ | Ks   | Kr   | bs  |
|------|---|------|------|-----|
| 216.75 | 28.85 | 21700 | 184000 | 1200 |

The initial population required for the genetic algorithms for this application was taken as a set of six identification model orders between 2 and 7, and the initial population for the PID shifting parameter \( l \) was taken as another set of six values between 0.1 and 0.2. The suspension system was simulated under two disturbance and mode change patterns of disturbance/mode changes: in the first case, the disturbance in a Repeated Road Step Bump in \( Z_r \), plus a 30% unloading at 5 seconds (half the simulation time); and in the second case, the disturbance is a Two Road Bump disturbance in \( Z_r \), plus a -30% at 5 seconds.

The vehicle output movement \( Z_s \) has been investigated with three controllers: a) A stationary (fixed parameter) PID designed and tuned for the initial system dynamics, b) An online real time Self Tuning PID as described in the previous section, and c) The presented GASL STR, where the STR controller is also the same PID. The system response for each of the control systems and disturbance/mode change cases are given in Figures 4 and 5. In both cases, a consistent pattern of superiority of the GASL STR is demonstrated. All three controllers perform well and at the first simulation disturbance. As expected, the stationary PID performance deteriorates as the disturbances repeated, whereas the adaptive STR and the GASL STR were able to maintain a much better control as compared to their stationary non-adaptive counterpart. However, under repeated disturbances as well as a simulated modality change, the adaptive STR deteriorates significantly as shown by the large oscillations exhibited in its system response in Figures 4 and 5.

The system response with GASL STR maintained desirable performance under those disturbance and modality change.

4. Conclusions

Adaptive Controllers can be effective for time varying and nonlinear systems, but they have limitations when such systems operate with drastic and/or modality changes. A GASL has been developed and presented for the control of such systems and to enhance the performance of the adaptive controllers in the STR framework. The GASL is designed to tune the adaptive STR identification model and controller parameters to maintain desirable dynamic performance. The addition of the GASL to the PID STR expands the STR ability to maintain control of nonlinear systems under large and quick shifts in system modes and/or disturbances compared to the basic STR. The developed GASL has been applied to a nonlinear adaptive suspension system for assessment. Simulation investigation results have been presented to show the effectiveness of utilizing the developed supervisory loop technique within the adaptive STR framework as it maintained system desirable performance under repeated disturbance and modality change conditions.

REFERENCES

[1] “A Neural Network Parallel Adaptive Controller for Fighter Aircraft Pitch-Rate Tracking”, Sukumar Kamalasadan, Adel A. Ghandakly, IEEE Transactions of Instrumentation and Measurements, January 2011.

[2] “A Neural Network Parallel Adaptive Controller for dynamic Systems Control”, S. Kamalasadan, A. Ghandakly, In Press, IEEE Transactions of Instrumentation and Measurements, June 2007.

[3] “An Intelligent Adaptive Controller Design Based on a Genetic Algorithm”, Sam A. McGilvery, MS Thesis 2005, The University of Toledo, Toledo, Ohio.

[4] “Genetic Algorithms Optimisation of Decoupled Sliding Mode Controllers: Simulated and Real Results”, E Alfaro-Cid, E McGookin, D Smith, T Fossen, Control Engineering Practice, 2005, pp. 739-748.

[5] “PID Parameters Optimization Using Genetic Algorithm Technique for Electrohydraulic Servo Control System”, Ayman A. Aly, Intelligent Control and Automation, 2011, #2, pp. 69-76.

[6] “Design of Self-Tuning PID Power System Stabilizer ForMultimachine Power Systems”, Chi-hi Wu, Yuan-Yih Hsu, IEEE Transactions on Power Systems. Vol. 3, No. 3, August 1988, pp. 1059-1064.
[7] “Active Nonlinear Vehicle-Suspension Variable-Gain Control”, Karim A. Tahboub, Proceedings of the IEEE 13th Mediterranean Conference on Control and Automation, Limassol, Cyprus, June 27-29, 2005.