2008 American Control Conference
Westin Seattle Hotel, Seattle, Washington, USA
June 11-13, 2008

A Model Free Automatic Tuning Method for a Restricted Structured Controller by Using Simultaneous Perturbation Stochastic Approximation (SPSA)

QingHui Yuan

Abstract— **A model free auto tuning algorithm is developed by using Simultaneous Perturbation Stochastic Approximation (SPSA). For such a method, plant models are not required. A set of closed loop experiments are conducted to generate data for an online optimization procedure. The optimum of the parameters of the restricted structured controllers will be found via SPSA algorithm. Compared to the conventional gradient approximation methods, SPSA only needs the small number of measurement of the cost function. It will be beneficial to application with high dimensional parameters. In the paper, a cost function is formulated to directly reflect the control performances widely used in industry, like overshoot, settling time and integral of absolute error. Therefore, the proposed auto tuning method will naturally lead to the desired closed loop performance. A case study of auto tuning of spool position control in a twin spool two stage valve is conducted. Both simulation and experimental study in TI C2000 target demonstrate effectiveness of the algorithm.**

I. Introduction

Automatic tuning of PID controllers have been one of the active research areas for decades. Automatic tuning makes it possible to automatically generate gain schedules. The design methods of automatic tuning can be classified into three categories (1) feature based techniques; (2) analytical methods; and (3) optimization [1]. The feature based automatic tuning includes Ziegler-Nichols methods, and its families with further modification [2]. In time domain, Ziegler-Nichols methods can directly provide the gains based on the dominant dead time and the dominant time constant of the step response. In the frequency domain, the ultimate gain and ultimate frequency will directly offer the PID gains through the tuning formula [2]. However, it is well known that Ziegler-Nichols design criterion gives an underdamped system performance, which is undesirable for many cases. When it comes to analytical methods, we have pole-placement analysis, Internal Model Controller (IMC), and so on. Analytical method can provide the predictable performance if the dynamics of open loop system are known. For optimization based methods, Modulus Optimum (BO) and Symmetrical Optimum (SO) are the most often used methods in frequency domain. They attempt to obtain the optimal loop transfer function, from which the desired controller is constructed.

As aforementioned, most of available auto-tuning methods need plant models. Nevertheless, plant models are sometimes difficult/expensive to obtain. A first principle based model usually needs a lot of effort to develop and to validate; while a detailed empirical model requires a large amount of data from the process. In addition, variation of dynamics from one component to another due to manufacturing tolerance makes it difficult to apply an identical plant model to different instances. Therefore, an online model free tuning methods will be of significant interest in industry. For such a method, plant models will not be required. A set of specific closed loop experiments, involving both the actual unknown plant and the proposed controller, are conducted to generate data for use in an online optimization procedure. A cost function will be defined in association with the closed loop performance. The online optimization will locate the optimal values of the corresponding parameters of the controller. The model free auto tuning approach is suitable for restricted structure controller whose structure and parameters have been predefined. Such a controller could be linear, like phase-lead, phase-lag or PID controllers [3, Chapter 3], or nonlinear.

For model free optimization based auto-tuning algorithm, the critical part is the recursive search algorithm. In industry, computation capability of embedded electronics is usually limited due to cost constraints. However, the fast convergence of the algorithm is preferred in order to improve productivity and reduce downtime. Some typical optimization algorithms, like Newton-Raphson and Steepest Descent, will be effective if good gradient information is available. In many cases, the gradient is not given and can only be approximated. For those methods like finite difference equations, the larger number of gain perturbations and system response generation are needed. There have been some efforts of generating gradient with less computation. For example, Controller Parameter Cycling Tuning [Page. 128] [3] uses trigonometric function orthogonality to extract gradient and Hessian information. The gradient can be obtained via sine gain perturbation - sine extraction, while the Hessian matrix can be calculated via sine gain perturbation - cosine extraction. However, the problem is that the number of measurement of the cost function will dramatically increases as the dimension of parameters expands.

In this paper, we will adopt Simultaneous Perturbation Stochastic Approximation (SPSA) into auto-tuning algorithm. SPSA is a unique optimization method that can efficiently approximate gradient via the cost function measurement. The benefit of SPSA is that the number of measurement of the cost function is independent of the dimension of the parameters. In other words, this method is especially efficient in high dimensional problems. A good solution can be offered by for a small number of measurement of the cost function.

The rest of the paper will be organized as follows: In section II, the diagram of auto-tuning algorithm is introduced. The details of cost function formulation and the

Q. Yuan is with Eaton Corp. Innovation Center, Eden Prairie, MN 55344, USA (email: QinghuiYuan@eaton.com).

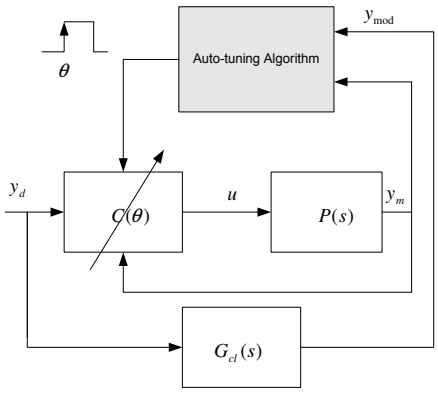

Fig. 1. The diagram of the auto-tuning algorithm, in which $G_{cl}(s)$ is the desired closed loop system transfer function, $P(s)$ is the actual (nonlinear) plant, $C(\theta)$ is the parametric controller where θ is the control parameters, y_m is the measured output, and y_{mod} is the output from the reference model $G_{cl}(s)$.

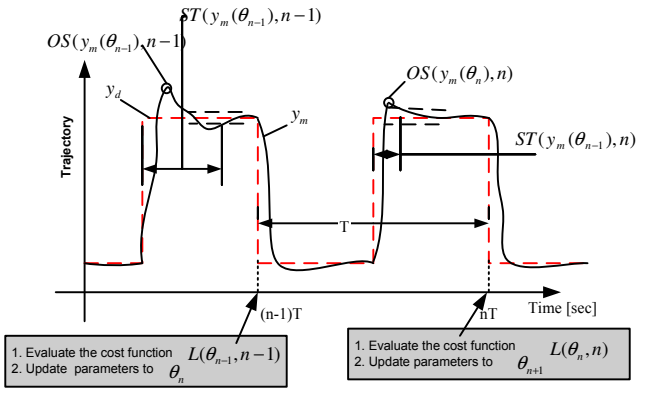

Fig. 2. Step response profile and Discrete Distributed (DD) cost function.

SPSA algorithm is presented, and auto tuning procedure is developed. In Section III, the case study of automatic tuning of spool position controller of an Electro-Hydraulic valve is conducted, both in simulation and experimentally. The concluding remarks are included in Section IV.

II. Automatic Tuning Algorithm for a Restricted Structured Controller

The diagram of the auto-tuning algorithm is illustrated in Fig. 1, where $P(s)$ is the plant, $C(\theta)$ is a restricted structured controller that consists of the time invariant parameters $\theta \in R^p$ where p is the dimension of the parameters, and $G_{cl}(s)$ is a reference model $G_{cl}(s)$ representing a desired transfer function from the set point to the output. For simplicity, it will usually a low order approximation.

Auto-Tuning Algorithm is located in the upper part of the diagram. It continuously takes the outputs from the actual plants and the reference models. The search algorithm is used to recursively update the parameters in order to reduce the cost function. The cost function can be selected to reflect the error between the reference model output and the actual output.

The time domain based method is adopted due to its easier use and less complexity compared to the frequency domain based method [4]. Among available time based methods, step response is very effective. Note that the *closed loop* step response method described here is different from the conventional ones. In [1], for example, step response of the plant is introduced, system ID is used to estimate the model, and then the gains of the controller can be set based on the obtained plant model. By contrast, what to be addressed is the closed loop step response. In Fig. 1, it is the set point y_d, instead of control command u, that will be step trajectory. The ultimate goal is to find the optimal gain θ such that y_m approaches to y_{mod}. Such a method may even be applied to the plant that is open loop unstable.

In this section, the cost function of the auto tuning problem is formulated to directly reflect the control performance specs widely used in industry. Then, the conventional SPSA, as well as its modified version in order to enhance convergence, is presented. Finally, SPSA based automatic tuning procedure is introduced.

A. Discrete Distributed (DD) Cost Function

In any control system design, the performances like settling time, overshoot, and integral of the absolute error, are of prime importance. It would be very useful if auto tuning process can directly help to achieve the performance goals. A cost function needs to be formulated to capture the associated performance.

These specs are related to the full cycle of the step profile. Therefore, should a cost function be constructed to reflect these specs, it can only be evaluated after all the information for a period have been collected. In other words, for a periodic step response, cost functions will be evaluated in the end of each period, as illustrated in Fig. 2. A cost function is defined as

$$L(\theta,n) = w_1 \int_{nT}^{(n+1)T} \|y_m(\theta) - y_{mod}\| dt + w_2 \|OS(y_m(\theta),n) - OS_d\| + w_3 \|ST(y_m(\theta),n) - ST_d\| \quad (1)$$

in which $L(\theta,n)$ is a cost function for a given parameter θ in the time span $t \in [nT,(n+1)T)$ where $n = 1,2,\cdots$ and T is the period of the step profile. y_m is the measurement, y_{mod} is the output of the reference model, $OS(x,n) : \{R^z, R\} \to R$ is the mapping of the trajectory $x \in R^z$ to the overshoot in the time span $t \in [nT,(n+1)T)$, $OS_d \in R$ is the desired overshoot, $ST(x,n) : \{R^z, R\} \to R$ is the mapping of the trajectory $x \in R^z$ to the settling time in the time span $t \in [nT,(n+1)T)$, and $ST_d \in R$ is the desired settling time, $w_i > 0$ for $i = 1,2,3$ are the weighting functions for integral of the absolute error, overshoot and setting time.

The cost function in Eq. (1) is unique due to its discreteness and distributiveness. Firstly, the cost function is evaluated in the end of each step response cycle. Such a process is a *discrete* event with the frequency of $1/T$. Secondly, given a set of parameters, one cycle is needed to evaluate a cost function. For more than one cost function, they have to be *distributed* and evaluated in multiple cycles. In Fig. 2, for example, two cost functions, $L(\theta_{n-1},\cdot)$ and $L(\theta_n,\cdot)$, are evaluated at $(n-1)T$ and nT, respectively.

The tuning process can be formulated as an optimization problem:

$$\theta^* = \arg\min_{\theta} L(\theta, n) \tag{2}$$

where θ^* is the optimal parameters that minimize the cost function for all n.

B. Conventional Simultaneous Perturbation Stochastic Approximation (SPSA)

For such an optimization problem, the search algorithm is key. Note that many popular algorithms, like Newton-Raphson and Steepest Descent, cannot be used directly since the gradient information is unavailable. We have to resort on the gradient approximation. In [5], the comparative study has been conducted for various gradient approximation methods. It has been found that Simultaneous Perturbation Stochastic Approximation (SPSA) is the preferable algorithm than the standard finite difference SA (FDSA) and the random direction SA (RDSA).

The generic recursive SA procedure is [6]

$$\hat{\theta}_{k+1} = \hat{\theta}_k - a_k \hat{g}(\hat{\theta}_k) \tag{3}$$

where $\hat{\theta}_k \in R^p$ is an approximate of the solution θ^* at kth step of recursion, $\{a_k\}$ is a sequence of positive scalars tat approaches zero gradually, $\hat{g}(\cdot) \in R^p$ is an approximation of the gradient $g(\cdot)$, and $k = 1, 2, 3, \cdots$ counts the iterations of the algorithm.

In general, the SPSA gradient approximation for $g(\hat{\theta}_k)$ is

$$\hat{g}(\hat{\theta}_k) = \begin{bmatrix} \frac{L(\hat{\theta}_k + c_k\Delta_k, \cdot) - L(\hat{\theta}_k - c_k\Delta_k, \cdot)}{2c_k\Delta_{k1}} \\ \frac{L(\hat{\theta}_k + c_k\Delta_k, \cdot) - L(\hat{\theta}_k - c_k\Delta_k, \cdot)}{2c_k\Delta_{k2}} \\ \cdots \\ \frac{L(\hat{\theta}_k + c_k\Delta_k, \cdot) - L(\hat{\theta}_k - c_k\Delta_k, \cdot)}{2c_k\Delta_{kp}} \end{bmatrix} \tag{4}$$

where $\{c_k\}$ is a sequence of positive scalars, $\Delta_k \in R^p$ is a vector with Bernoulli distribution at the kth step of iteration where Δ_{ki} for $i = 1, 2, \cdots, p$ is the ith component of Δ_k, $L(\cdot, \cdot)$ is the cost function (1).

Note that the the gradient approximation from the standard finite difference approximation is proportional to the number of the parameters. For high dimensional parameters, the evaluation of the objective function will be costly. Since each cost function need one full cycle of step response, tuning process could be time consuming. By contrast for the conventional SPSA, only two measurements of the cost functions, $L(\hat{\theta}_k + c_k\Delta_k, \cdot)$ and $L(\hat{\theta}_k - c_k\Delta_k, \cdot)$, are required [7] [8].

C. Modified SPSA

1) Normalization for Bernoulli distribution: In order to balance the convergence in all parameter dimensions , we have to normalize the Bernoulli distribution with respect to the range of parameters. The probability mass function for Δ_{ki}, each element of $\Delta_k := [\Delta_{k1}, \Delta_{k2}, \cdots, \Delta_{kp}]^T$, is given by

$$f_{\Delta_i}(x) = \begin{cases} 0.5 & x = 0.5\delta(\bar{\theta}_i - \underline{\theta_i}) \\ 0.5 & x = -0.5\delta(\bar{\theta}_i - \underline{\theta_i}) \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

where $0 < \delta < 1$ is a scalar determining the update ratio, and $\underline{\theta_i}$ and $\bar{\theta}_i$ are the lower bound and upper bound of each element of θ, respectively, with $\underline{\theta_i} \le \theta_i \le \bar{\theta}_i$.

2) Convergence consideration: SPSA has the first order version and the second order version. For the second order SPSA, the similar technique used to approximate the gradient can be extended for evaluation of Hessian Matrix. The second order SPSA converges faster, and the first order SPSA will slow down the convergence rate once an optimum is approached. In our application, we would like to use the first order SPSA aforementioned due to its simplicity, and find a way to improve its convergence rate.

The following enhancements has been considered to speed convergence and to increase algorithm stability.

- Iteration rejection for overaggressive update

$$\hat{\theta}_k = IR_1(\hat{\theta}_{k-1}, \hat{\theta}_k) := \begin{cases} \hat{\theta}_{k-1} & \text{if } \|\hat{\theta}_k - \hat{\theta}_{k-1}\| > M_1 \\ \hat{\theta}_k & \text{otherwise} \end{cases} \tag{6}$$

  where $IR_1(\cdot)$ is a function reflecting iteration rejection for overaggressive update, $M_1 > 0$ is a large scalar.
- Iteration rejection based on cost function comparison For the typical SPSA algorithm, each iteration k needs two evaluation of the cost functions $L(\hat{\theta}_k + c_k\Delta_k, \cdot)$, $L(\hat{\theta}_k - c_k\Delta_k, \cdot)$. It has been suggested in [7] that the third evaluation is added on $L(\hat{\theta}_k, \cdot)$. For the extra cost function evaluation, the benefit would be directly comparison of the cost function between the adjacent cost functions on $\hat{\theta}_k$ and $\hat{\theta}_{k-1}$.

$$\begin{aligned} \hat{\theta}_k &= IR_2[\hat{\theta}_{k-1}, L(\hat{\theta}_{k-1}, \cdot), L(\hat{\theta}_k, \cdot)] \\ &:= \begin{cases} \hat{\theta}_{k-1} & \text{if } L(\hat{\theta}_k, \cdot) - L(\hat{\theta}_{k-1}, \cdot) > M_2 \\ \hat{\theta}_k & \text{otherwise} \end{cases} \end{aligned} \tag{7}$$

  where $IR_2(\cdot)$ is a function reflecting iteration rejection based on cost function comparison, $M_2 > 0$ is a large scalar.

D. Automatic Tuning Procedure

Considering discreteness and distributiveness of the cost function, the overall automatic tuning procedure is proposed as follows:

1) Initialization at $t = 0$.

$$\begin{aligned} k &= 1 \\ \hat{\theta}_0 &= 0.5(\underline{\theta} + \bar{\theta}) \\ \hat{\theta}_1 &= 0.5(\underline{\theta} + \bar{\theta}) \\ L_0^O := L(\hat{\theta}_0, \cdot) &= M_3 \\ \theta_n &= \hat{\theta}_1 + c_1\Delta_1 \end{aligned} \tag{8}$$

where $M_3 > 0$, θ_n is the parameter for the next period. In other words, the controller will be in the form of $C(\theta_n)$ for the next period.

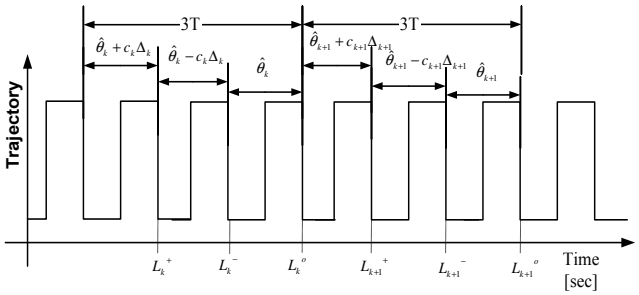

Fig. 3. Auto tuning procedure. The values of the parameters in each step cycle are shown in the top of the profile. Three cycles are needed for $\hat{\theta}_k$ iteration.

2) At $t = (3k-2)T$,
Evaluate the cost function $L_k^+ = L(\theta_n, 3k-2)$.
Update the parameter for the next cycle evaluation $\theta_n = \hat{\theta}_k - c_k\Delta_k$.
3) At $t = (3k-1)T$
Evaluate the cost function $L_k^- = L(\theta_n, 3k-1)$.
Update the parameter for the next cycle evaluation $\theta_n = \hat{\theta}_k$.
4) At $t = 3kT$
   a) Evaluate the cost function $L_k^O = L(\theta_n, 3k)$
   b) Check iteration rejections in Eq. (6) (7). Or, evaluate $\hat{\theta}_k = IR_1(\hat{\theta}_{k-1}, \hat{\theta}_k)$ and $\hat{\theta}_k = IR_2(\hat{\theta}_{k-1}, L_{k-1}^O, L_k^O)$.
   c) Update a_k, c_k.
   d) Generate Δ_k with probability distribution in Eq. (5).
   e) The gradient approximate is rewritten from Eq. (4)

$$\hat{g}(\hat{\theta}_k) = \left[\frac{L_k^+ - L_k^-}{2c_k\Delta_{k1}} \frac{L_k^+ - L_k^-}{2c_k\Delta_{k2}} \cdots \frac{L_k^+ - L_k^-}{2c_k\Delta_{kp}} \right]^T \quad (9)$$

   Note that a variant of gradient approximation method [7] can be easily integrated for smoothness.
   f) Update the parameter vector $\hat{\theta}_k$ via

$$\hat{\theta}_{k+1} = \hat{\theta}_k - a_k\hat{g}(\hat{\theta}_k) \quad (10)$$

   g) Update iteration index $k = k+1$. If $k > N$ where $N > 0$ is a scalar, the procedure stops.
   h) Check boundary constraints. For $\hat{\theta}_{ki}$, each element of $\hat{\theta}_k$, constrain the update within the bound via

$$\hat{\theta}_{ki} = \begin{cases} \underline{\theta}_i & \text{if } \hat{\theta}_{ki} < \underline{\theta}_i \\ \overline{\theta}_i & \text{if } \hat{\theta}_{ki} > \overline{\theta}_i \\ \hat{\theta}_{ki} & \text{otherwise} \end{cases} \quad (11)$$

   for $i = 1, 2, \cdots, p$.
   i) Update parameter for the next cycle evaluation $\theta_n = \hat{\theta}_k + c_k\Delta_k$.
   j) Go to 2)

Since we add the third evaluation of the cost function to improve convergence rate and stability, $3T$ is the minimal time required for iteration procedure $2) - 4)$, as illustrated in Fig. 3.

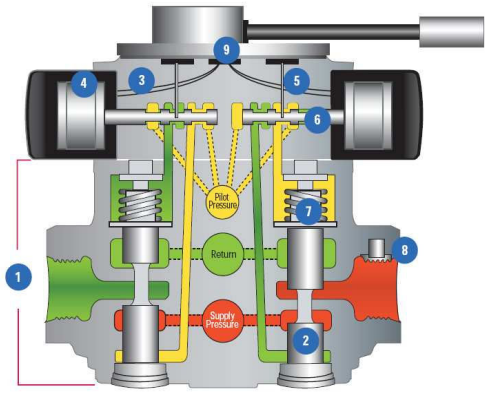

Fig. 4. The cross section of Ultronics valve (Courtesy of Eaton Corp.) 1- main stage valve block, 2 - independent spool for metering, 3 - pilot valve, 4 - voice coil actuator, 5 - centering spring, 6 - pilot spool, 7 - LVDT position sensor, 8 - thin film pressure sensor, 9 - microcontroller [10].

It is worth mentioning that the procedure in 4) is more computation intensive than 1), 2) or 3). However, the task can be divided and executed in more than one sampling loop. The error will be negligible since the sampling rate is much higher than the step response frequency.

III. Case Study

In this section, an Electrohydraulic system is targeted to apply the modified SPSA based auto tuning algorithm. In this case study, we select a twin spool two stage valve (Ultronics, Eaton Corp., US) as the platform. The Ultronics valve system is an advanced electro-hydraulic control valve. It has the unique two-stage twin spool configuration. Unlike its traditional predecessors, the Ultronics valve system combines the flexibility of software along with its independent metering spools technology to give the complete control over the machines hydraulic equipment. Ultronics are used in mobile applications in construction, forestry, agriculture, and other markets [9] [10].

The schematic of an Ultronics flow control valve is illustrated in Fig. 4. The sensors are embedded inside the valve so that the valve port pressures and the main stage spool position can be measured. Therefore, the closed loop control algorithm can be loaded into the embedded electronics to achieve a variety of control objectives, like pressure/flow/position regulation. Many extra advanced features in the vehicle level application can even be possible. However, ability to control the main stage spool position is key for all other functionalities. In this case study, we will consider a restricted structured controller for position regulation.

The first principle full order valve model is highly nonlinear. The detailed description of modeling of such a twin spool two stage valves can be seen in our previous work [11]. However, based on the assumption that the pilot stage has faster response compared to the main stage, we can get the reduced order linear system approximation. Simple analysis shows that the control command is positive associated with the main stage spool velocity. A velocity feedforward PI controller [12] can be selected for trajectory tracking of spool

position.

$$u = K_d \frac{d}{dt} y_d + K_p(y_d - y_m) + K_i \int (y_d - y_m) dt \quad (12)$$

where y_d is the desired position, y_m is the actual main stage spool position, u is the current through the voice coil actuator, as can be seen in Fig. 1. The selected controller falls into the category of a restricted structured controller $C(\theta)$ with the parameters $\theta = \begin{bmatrix} K_d & K_p & K_i \end{bmatrix}^T$.

As aforementioned, for this time domain based auto tuning algorithm, we will use the periodic step response. In the case study, we particularly customize the step profile during the portion of the steady state range to emphasize tracking (minimizing integral of absolute error). A mathematic representation of such a profile is given by

$$y_d = \begin{cases} A_d + O_d, & t_m \in [0, 0.5dT); \\ A_d + O_d + A_s \sin(\omega_s t + \phi_s), & t_m \in [0.5dT, dT); \\ O_d, & t_m \in [dT, T) \end{cases} \quad (13)$$

where A_d O_d are the amplitude and the offset of the step profile, respectively, A_s, ω_s and ϕ_s are the amplitude, the frequency, and the phase of the sinusoidal signal, d is the duty ratio, T is the period of the step profile, and $t_m := mod(t, T)$.

A. Simulation study

The physical model of the Ultronics valve is developed in Matlab/Simulink (Mathworks, US). The proposed restricted structured controller in Eq. (12) is implemented to regulate the plant model.

Monte Carlo simulation is conducted in terms of various combination of controller gains. As can be seen in Fig. 5 6, we exhaustively visit the possible combination for K_d, K_p and K_i. Each period of the step profile corresponds to a set of control gains. For instance, the plot in the most upper left corner of Fig. 5 contains eight step cycles. The first cycle corresponds to a set of gains $\{K_d = 0.5, K_p = 0.02, K_i = 0.45\}$, and the second one refers to $\{K_d = 0.5, K_p = 0.02, K_i = 1.32\}$, and so on. In Figs. 5 6, K_p values increase from top to bottom on the left side, and then from top to bottom on the right side. K_i values monotonically increase from left to right in each plot.

Several observations from Monte Carlo analysis in Figs. 5 6:

- Larger K_i causes greater natural frequency and smaller damping ratio, as can be seen in each plot in Fig. 5 6. This is consistent with the closed loop transfer function analysis.
- As K_p increases, the overall performance is getting improved. However, if K_p is larger than 0.15, the performance starts to get degraded dramatically.
- Given K_p, the larger K_i induces the larger overshoot. On the other hand, it is not clear for the relationship between the overshoot and K_p.
- K_d plays a role on determining optimal solution. Compared to the case with $K_d = 0.5$, the optimal solution for $K_d = 1.5$ shifts from the larger gains ($K_p = 0.14, K_i = 1.7$) to the smaller ones ($K_p = 0.095, K_i = 1.4$). This can be explained by the fact that the feedforward term contributes more to the control command as K_d increases. The consequence is that the optimal proportional and integral gain will be reduced.

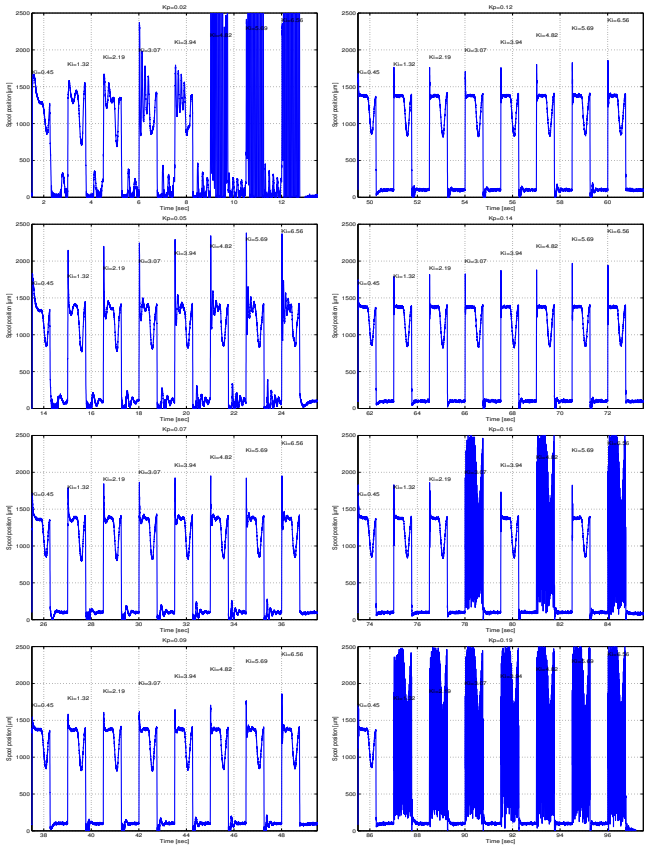

Fig. 5. The main stage spool trajectory for various control gains. $K_d = 0.5$.

Next, convexity of the optimization problem is investigated. If this is not a convex problem, then there is no guarantee that the solution will be global optimal. Based the collected data in Figs. 5 6, we represent the results in a contour format in Fig. 7. The figure shows that for each K_d, it is a convex problem in the search domain. It may not preserve its convexity in the entire feasible domain since not all possible solutions are exhaustively investigated. In practice, however, it implies that we will be solving a convex problem for an appropriately defined domain. The above analysis ensures the convergence of the SPSA auto-tuning algorithm.

The auto-tuning algorithm is implemented in Stateflow (Mathworks, US). The diagram is illustrated in Fig. 1. We can define a second order system $G_{cl}(s) = \frac{\omega_d^2}{s^2 + 2\zeta_d \omega_d s + \omega_d^2}$ where ζ_d and ω_d are the desired damping ratio and natural freqency. The auto-tuning process follows the description in Section II-C. Based on the empirical experience, the auto-tuning domains can be roughly specified as $K_d \in [\underline{K_d}, \overline{K_d}]$, $K_p \in [\underline{K_p}, \overline{K_p}]$, and $K_i \in [\underline{K_i}, \overline{K_i}]$.

The parameters used in the auto tuning method are given as follows: $\underline{K_d} = 0.01$, $\overline{K_d} = 1.75$, $\underline{K_p} = 0.01$, $\overline{K_p} = 0.375$,

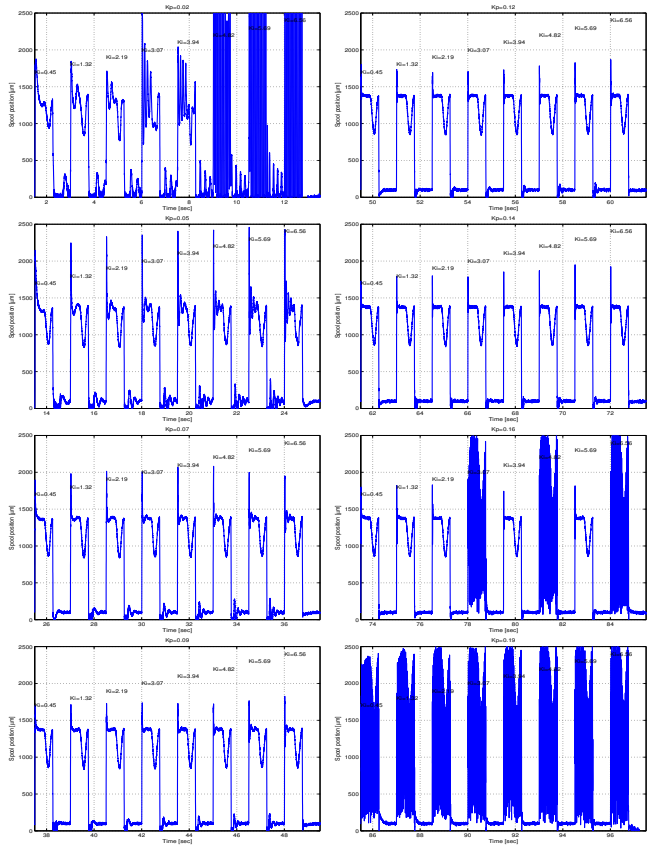

Fig. 6. The main stage spool trajectory for various control gains. K_d is fixed to be 1.5.

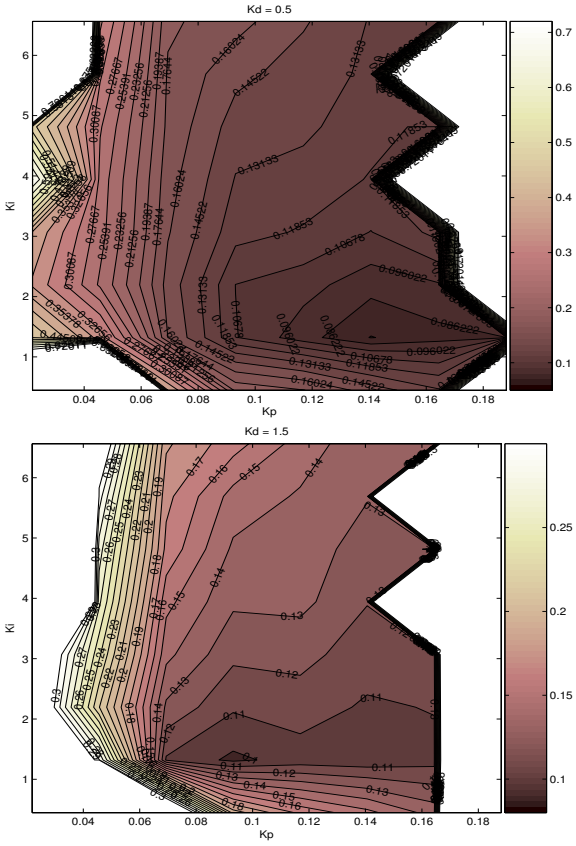

Fig. 7. The contour of the cost function in Eq. 1 with respect to K_p and K_i. Upper: $K_d = 0.5$; Lower: $K_d = 1.5$. Note that the optimal solution has changed according to K_d. In this case, as K_d increases, the feedforward term contributes more to the control command. The consequence is that the optimal proportional and integral gain will be reduced.

$\underline{K_i} = 0.01$, $\overline{K_i} = 1.5$, $a_k = (1+k)^{-0.5}$, $c_k = 0.125(1+k)^{-0.5}$, $M_1 = 1$, $M_2 = 0.1$, $M_3 = 10$, $\delta = 1$, $w_1 = 0.5$, $w_2 = 0.3$, $w_3 = 0.2$.

In the auto tuning process, $\hat{\theta}_k$ is iterated as illustrated in Fig. 8. Note that its iteration rate is 4.5 [sec]. This is consistent with the fact that the iteration period of SPSA tuning will be three times of that of the step profiles (see Section II-C) with $T = 1.5$ [sec]. The convergence rate is relatively fast. After 10 iterations (at $t = 45$ [sec]) , the gains are already very close to the final solution. The optimal gains from the SPSA algorithm is $\{K_d = 1.4, K_p = 0.1, K_i = 1.25\}$.

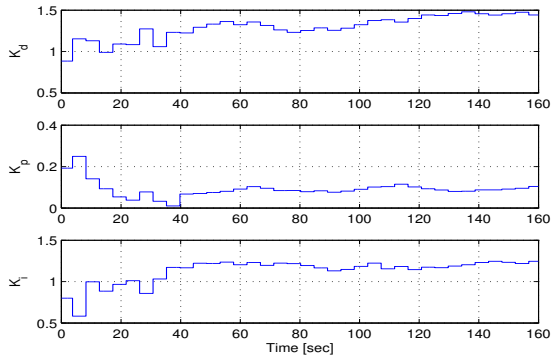

Fig. 8. $\hat{\theta}_k$ iteration in the process of auto tuning.

The performance index (cost function) has been optimized during auto-tuning process. In Fig. 9, the desired and actual main stage spool positions are illustrated. The left plot shows those in the beginning of auto-tuning; while the right one is the profiles in the end of auto-tuning. The performance has been significantly improved.

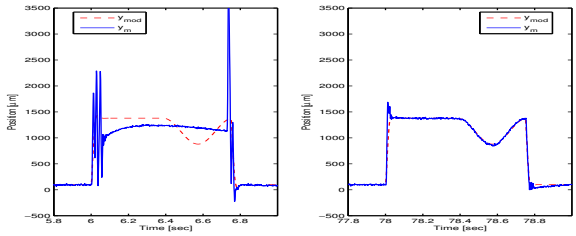

Fig. 9. The desired closed loop system response y_{mod} and the actual main stage spool position y_m. Left: the beginning of the auto-tuning; right: the end of the auto-tuning.

It is worth mentioning that instability may occur during auto tuning process for a certain combination of parameter values. In order to protect the system under test, it will be useful to include the mechanism for instability detection and suppression. In the left plot in Fig. 9, vibration occurs at $t = 6$ [sec]. Instability has been detected at $t = 6.05$ [sec], and eliminated by resetting the parameter values.

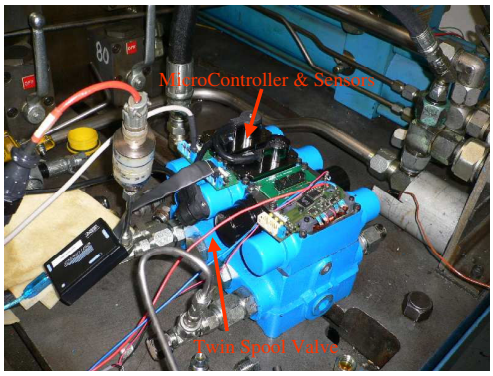

Fig. 10. Experimental setup of the twin spool two stage valve.

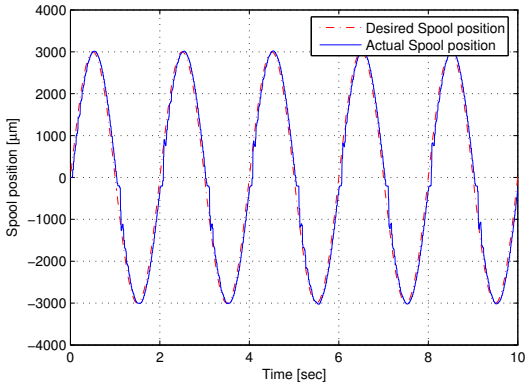

Fig. 11. The tracking performance of the auto-tuned restricted structured controller for the 0.5 [Hz] sinusoidal signal.

B. Experimental Study

The proposed auto-tuning algorithm has been validated experimentally. The experimental setup is illustrated in Fig. 10. TI C2000 series (TI, US) is selected as target [13]. Target TI C2000 Toolbox (Mathworks, US) is integrated with TI code composer studio for code generation, compiling, linking, and downloading. TI JTAG emulator is used for control and monitoring the signals, shown as a black block in Fig. 10.

In the study, we turn on auto-tuning for two minutes. After the SPSA algorithm stops, the tuned optimal gains will be automatically saved. The velocity feedforward PI controller is with the tuned gains $K_d = 0.215, K_p = 0.047, K_i = 0.08$. Note that the gains from the experimental study is different from those from simulation. The reason is that for the quite complex plant model, not all the parameters are precisely calibrated. Variation of the actual plant from model causes different tuning results. In addition, the tuning results for one individual valve are different from another in experimental study, since manufacturing tolerance will introduce the slightly different dynamic behavior. The level of gain variation depends on the level of variation of electro-hydro-mechanical property.

The position control is tested for the auto-tuned velocity feedforward PI controller. Without loss of generality, a sinusoidal signal with 0.5 [Hz] frequency and 3000 $[\mu m]$ amplitude is set to be the tracking objective. As shown in Fig. 11, the actual position tracks the desired position quite well for the auto tuned parameter values of the controller. Note that the tracking performance is disturbed around $y_m = -100$ $[\mu m]$. This is caused by nonlinear behaviors of the mechanical system that are not considered by the selected restricted structured controller.

IV. Conclusion

In the paper, a model free auto tuning algorithm for restricted structured controllers is developed. A unique discrete distributed (DD) cost function is defined so that the auto tuning can directly reflect some typical performance requirements in industry, like integral of absolute error, overshoot and settling time. In addition, the industrial solution needs fast convergence even with the limited computation capability. In the paper, Simultaneous Perturbation Stochastic Approximation (SPSA) technology is utilized to optimize the parameters. The benefit of this approaches is that only three measurement of the cost function are needed independent the dimension of the parameters. It is especially efficient for high dimensional problem. The modification of SPSA in terms of Normalization of Bernoulli distribution, and iteration rejection has been made in order to improve convergence stability and rate. A case study of auto tuning of a twin spool two stage valve is conducted. The simulation study verifies that the Modified SPSA is effective to locate the optimal parameters of the proposed restricted structured controller. The algorithm has also been implemented in TI C2000 embedded target. The experimental study shows that the auto-tuning process can be successfully executed in the target. The controller with the tuned gains provides a good tracking performance.

References

[1] W. S. Levine, *The Control Handbook*. Prentice Hall, 1996.
[2] C. C. Hang, K. J. Astrom, and W. K. Ho, "Refinements of ziegler-nichols tuning formula," in *IEE, Part D*, vol. 138, no. 2, 1991.
[3] M. A. Johnson and M. H. moradi, *PID Control: New Identification and Design Methods*. Springer, 2005.
[4] T. Hagglund and K. J. Astrom, "Industrial adaptive controllers based on frequency response techniques," *Automatica*, vol. 27, no. 1, 1991.
[5] D. C. Chin, "Comparative study of stochastic algorithm for system optimization based on gradient approximation," *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, vol. 27, no. 2, April 1997.
[6] J. L. Maryak and D. C. Chin, "Global random optimization by simultaneous perturbation stochastic approximation," *Johs Hopkins APL Technical Digest*, vol. 25, no. 2, pp. 91–99, 2004.
[7] A. V. Wouwer, C. Renotte, and M. Remy, "Application of stochastic approximation techniques in neural modeling and control," *International Journal of System Science*, vol. 34, no. 14-15, pp. 851–863, Nov-Dec 2003.
[8] J. C. Spall, "Multivariate stochastic approximation using a simultaneous perturbation gradient approximation," *IEEE Transactions on Automatic Control*, vol. 37, no. 3, pp. 332–340, March 1992.
[9] Q. Yuan, S. Bengea, D. Piyabongkarn, and P. Brenner, "Dynamic control of a distributed embedded electro-hydraulic system," *SAE 2007 Transactions Journal of Engines*, vol. V116-3, no. 2007-01-1626, May 2008.
[10] Eaton-Corp., "Ultronics management system," 2005.
[11] Q. Yuan and J. Y. Lew, "Modeling and control of a two-stage twin-spool valve for energy-saving," in *the 2005 American Control Conference (ACC)*, vol. 6, Portland, OR, Jun 2005, pp. 4364–68.
[12] M. Jelali and A. Kroll, *Hydraulic Servo-systems: Modeling, Identification and Control*. Springer, 2003.
[13] Texas-Instruments, "Tms320 data manual," 2006.