International Journal of Intelligence Science, 2014, 4, 70-79
Published Online July 2014 in SciRes. http://www.scirp.org/journal/ijis
http://dx.doi.org/10.4236/ijis.2014.43009

Adaptive Cascade Generalized Predictive Control

Tao Geng[1], Jin Zhao[2]

[1]Academy of Physics and Electroics, Henan University, Kaifeng, China
[2]Academy of Control Science and Engineering, Huazhong University of Science and Technology (HUST), Wuhan, China
Email: gengtao@henu.edu.cn, jinzhao617@mail.hust.edu.cn

Received 25 May 2014; revised 25 June 2014; accepted 2 July 2014

Abstract

Cascade control is one of the most popular structures for process control as it is a special architecture for dealing with disturbances. However, the drawbacks of cascade control are obvious that primary controller and secondary controller should be tuned together, which influences each other. In this paper, a new Adaptive Cascade Generalized Predictive Controller (ACGPC) is introduced. ACGPC is a method issued from GPC and the inner and outer controllers of a cascade system are replaced by one cascade generalized predictive controller, where both loops model are updated by Recursive Least Squares method. Compared with existing methods, the new method is simpler and yet more effective. It can be directly integrated into commercially available industrial auto-tuning systems. Some examples are given to illustrate the effectiveness and robustness of the proposed method.

Keywords

Adaptive Cascade Generalized Predictive Control, Model Identification, Cascade Control

1. Introduction

Cascade control is one of the most popular structures for process control as it is a special architecture for dealing with disturbances. It is widely used in practice, sometimes in a transparent way (embedded into the electronics of a servoactuator [1], controlled power supply [2], process control [3] etc.). The general block diagram of cascade control is shown in **Figure 1**. Cascade control refers to the design of a control loop for one primary variable by means of multiple sensors and/or actuators and, in its basic configuration; it consists of two cooperative SISO control loops with different time constants. The time constants in inner loop and outer loop are always different. And this difference allows for separate design using conventional techniques.

How to cite this paper: Geng, T. and Zhao, J. (2014) Adaptive Cascade Generalized Predictive Control. *International Journal of Intelligence Science*, **4**, 70-79. http://dx.doi.org/10.4236/ijis.2014.43009

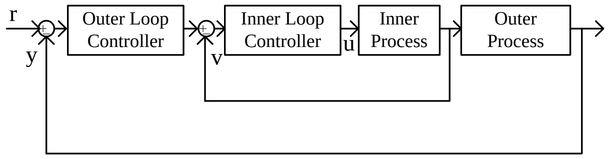


Figure 1. General block diagram of cascade control.

Previous researchers have proposed relay-based auto-tuning techniques to facilitate the design of cascade control systems. The methods proposed by Hang *et al*. [4] and Vivek and Chidambaram [5] need sequential application of the conventional relay-based auto-tuning approach, and are therefore still time consuming. The sequential tuning procedure has been improved so that only single relay experiment is required for auto-tuning [6]-[9]. However, an off-line or ad hoc experiment must be performed in these methods. For example, Leva and Donida [6] performed test with relay cascaded to an integrator, and Mehta and Majhi [8] restricted the secondary controller to controller during the relay test. Besides the relay-based method, Visioli and Piazzi [10] proposed an automatic tuning method consisting of an open-loop test for cascade control system. Veronesi and Visioli [11] recently proposed simultaneous closed-loop automatic tuning method for cascade controllers. Their method evaluates the set-point step response of cascade control system.

Whatever, based on the PID there are always two controllers needed to be configured. And obviously, the drawbacks of cascade control are obvious that primary controller and secondary controller should be tuned together, which influences each other. If there is not a substantial difference in time constants, although this strategy can still be pursued, the loop design cannot be made independently and based on SISO techniques. So tuning is not intuitive: centralized configurations might be preferable. CGPC [12] [13] is a method issued from GPC and the inner and outer controllers of a cascade system are replaced by one cascade generalized predictive controller. In this control paradigm, there is only one controller configured. If the models of inner process and out process were known, the controller can be auto-configured. This paper proposed an adaptive CGPC cascade controller shown in **Figure 2**. The both loops model are SISO. And the model can be identified by the classical method, respectively.

2. GPC with Constraints

GPC adapts the model so-called Controlled auto regressive integrated moving average (CARIMA) model [14] [15].

$$A\left(z^{-1}\right)\boldsymbol{y}=B\left(z^{-1}\right)\boldsymbol{u}+\frac{T\left(z^{-1}\right)}{\Delta}\boldsymbol{e} \tag{1}$$

where, $T\left(z^{-1}\right)$ is the model of noise, but it is commonplace to treat $T\left(z^{-1}\right)$ as a design parameter. Because it has direct effects on loop sensitivity and so better closed-loop performance will be got with a $T\left(z^{-1}\right)$.

Then, a general form of future predictions is

$$\begin{gathered}\underset{\rightarrow}{\boldsymbol{y}}=\Gamma\Delta\underset{\rightarrow}{\boldsymbol{u}}+\boldsymbol{y}_{\text{free}}\\ \boldsymbol{y}_{\text{free}}=\tilde{P}\Delta\underset{\leftarrow}{\tilde{\boldsymbol{u}}}+\tilde{Q}\underset{\leftarrow}{\tilde{\boldsymbol{y}}}\end{gathered} \tag{2}$$

And the followed can be derived

$$\Gamma=C_D^{-1}C_B \tag{3}$$

where, $D\left(z^{-1}\right)=\Delta A\left(z^{-1}\right)$

$$\begin{cases}\tilde{P}=C_TP-\Gamma H_T\\ \tilde{Q}=C_TQ+H_T\end{cases} \tag{4}$$

C_T is the toeplitz matrix of $T\left(z^{-1}\right)$ and H_T is the hankel matrix of $T\left(z^{-1}\right)$ [16]. Constraints on process inputs and outputs make the controller and consequently the entire closed-loop, nonlinear.

Use the cost function and optimization

$$J=\sum_{j=1}^{N_y}\left\|r\left(k+j\right)-\underset{\rightarrow}{y}\left(k+j\right)\right\|_2^2W_y\left(j\right)+\sum_{j=1}^{N_u}\left\|\Delta\underset{\rightarrow}{u}\left(k+j\right)\right\|_2^2W_u\left(j\right) \tag{5}$$

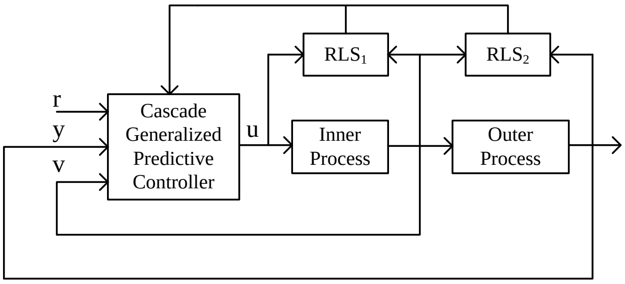


Figure 2. Proposed control paradigm of adaptive cascade control.

where, $r(k+j)$ is j step ahead reference. N_y is receding horizon and N_u is control step. W_y, W_u is the output and input weight factor. Described in matrix form, the GPC with constraints is converted to be the optimization problem which is

$$\begin{aligned} & J = \left\| \underrightarrow{r} - \Gamma \Delta \underrightarrow{u} - y_{\text{free}} \right\|_{W_y}^2 + \left\| \Delta \underrightarrow{u} \right\|_{W_u}^2 \\ & \min_{\Delta \underrightarrow{\boldsymbol{u}}} J = \Delta \underrightarrow{\boldsymbol{u}} S \Delta \underrightarrow{\boldsymbol{u}}^{\mathrm{T}} + 2 \boldsymbol{f}^{\mathrm{T}} \Delta \underrightarrow{\boldsymbol{u}} \\ & \text{s.t. } C \Delta \underrightarrow{\boldsymbol{u}} - \boldsymbol{d}_k \le 0 \end{aligned} \tag{6}$$

where, J is subject of the following constraints. Where S is positive definite and $\boldsymbol{f}$, $\boldsymbol{d}_k$ are time varying (dependent on the current state).

$$S = \Gamma^{\mathrm{T}} W_y \Gamma + W_u, \quad \boldsymbol{f} = -\Gamma^{\mathrm{T}} W_y \left(\underrightarrow{\boldsymbol{r}} - \boldsymbol{y}_{\text{free}} \right) \tag{7}$$

The (6) is a standard quadratic programming with constraints. The constraints in GPC will be described as the following.

- The Control Law without Constrains

Solve the $\dfrac{\partial J}{\partial u} = 0$

The control law without constrains is derived as

$$\Delta \underrightarrow{\boldsymbol{u}} = -S^{-1} \boldsymbol{f} = \left(\Gamma^{\mathrm{T}} W_y \Gamma + W_u \right)^{-1} \Gamma^{\mathrm{T}} W_y \left(\underrightarrow{\boldsymbol{r}} - \boldsymbol{y}_{\text{free}} \right) \tag{8}$$

- Input move constraints

$\Delta \underline{\boldsymbol{u}}$ is the lower bounds of input move constraints, and $\Delta \bar{\boldsymbol{u}}$ is the upper bounds of input move constraints.

$$\begin{bmatrix} \Delta \underline{\boldsymbol{u}} \\ \Delta \underline{\boldsymbol{u}} \\ \vdots \\ \Delta \underline{\boldsymbol{u}} \end{bmatrix} \le \begin{bmatrix} \Delta \boldsymbol{u}_k \\ \Delta \boldsymbol{u}_{k+1} \\ \vdots \\ \Delta \boldsymbol{u}_{k+Nu-1} \end{bmatrix} \le \begin{bmatrix} \Delta \bar{\boldsymbol{u}} \\ \Delta \bar{\boldsymbol{u}} \\ \vdots \\ \Delta \bar{\boldsymbol{u}} \end{bmatrix}$$

Which can be described in vector form as (9).

$$\Delta \underline{U} \le \Delta \underrightarrow{\boldsymbol{u}} \le \Delta \bar{U} \tag{9}$$

And satisfy the matrix inequality (10)

$$\begin{bmatrix} I \\ -I \end{bmatrix} \Delta \underrightarrow{\boldsymbol{u}} - \begin{bmatrix} \Delta \bar{U} \\ -\Delta \underline{U} \end{bmatrix} \le 0 \tag{10}$$

- Input constraints

$\underline{\boldsymbol{u}}$ is the lower bounds of input constraints, and $\bar{\boldsymbol{u}}$ is the upper bounds of input constraints.

$$\underrightarrow{\boldsymbol{u}} = \boldsymbol{C}_{I/\Delta} \Delta \underrightarrow{\boldsymbol{u}} + \begin{bmatrix} I \\ I \\ \vdots \\ I \end{bmatrix} \boldsymbol{u}_{k-1}$$

where, $\boldsymbol{C}_{I/\Delta}=\begin{bmatrix} I & 0 & \cdots & 0 \\ I & I & 0 & \cdots \\ \vdots & \ddots & \ddots & \ddots \\ I & \cdots & I & I \end{bmatrix}$.

$$\begin{bmatrix} \underline{\boldsymbol{u}} \\ \underline{\boldsymbol{u}} \\ \vdots \\ \underline{\boldsymbol{u}} \end{bmatrix} \le \begin{bmatrix} \boldsymbol{u}_k \\ \boldsymbol{u}_{k+1} \\ \vdots \\ \boldsymbol{u}_{k+Nu-1} \end{bmatrix} \le \begin{bmatrix} \bar{\boldsymbol{u}} \\ \bar{\boldsymbol{u}} \\ \vdots \\ \bar{\boldsymbol{u}} \end{bmatrix}$$

$$\underline{U} \le \boldsymbol{C}_{I/\Delta}\Delta\underset{\rightarrow}{\boldsymbol{u}} + L\boldsymbol{u}_{k-1} \le \bar{U} \tag{11}$$

The corresponding linear inequalities are

$$\begin{bmatrix} \boldsymbol{C}_{I/\Delta} \\ -\boldsymbol{C}_{I/\Delta} \end{bmatrix} \Delta\underset{\rightarrow}{\boldsymbol{u}} - \begin{bmatrix} \bar{U} - L\boldsymbol{u}_{k-1} \\ -\underline{U} + L\boldsymbol{u}_{k-1} \end{bmatrix} \le 0$$

- Output constrains

The output of plants always needs to be constrained in the bounds, as demand of process requirements. And they always are treated as soft constraints. $\underline{Y}$ is the lower bounds of output constraints, and $\bar{Y}$ is the upper bounds of output constraints.

$$\begin{aligned} &\underline{Y} \le \boldsymbol{y} \le \bar{Y} \\ &\underline{Y} \le \Gamma\Delta\underset{\rightarrow}{\boldsymbol{u}} + \boldsymbol{y}_{\text{free}} \le \bar{Y} \\ &\begin{bmatrix} \Gamma \\ -\Gamma \end{bmatrix} \Delta\underset{\rightarrow}{\boldsymbol{u}} - \begin{bmatrix} \bar{Y} - \boldsymbol{y}_{\text{free}} \\ -\underline{Y} + \boldsymbol{y}_{\text{free}} \end{bmatrix} \le 0 \end{aligned} \tag{12}$$

If all constraints are satisfied, the (6) can be described as

$$\boldsymbol{C} = \begin{bmatrix} I \\ -I \\ \boldsymbol{C}_{I/\Delta} \\ -\boldsymbol{C}_{I/\Delta} \\ \Gamma \\ -\Gamma \end{bmatrix}, \quad \boldsymbol{d}_k = \begin{bmatrix} \Delta\bar{U} \\ -\Delta\underline{U} \\ \bar{U} - L\boldsymbol{u}_{k-1} \\ -\underline{U} + L\boldsymbol{u}_{k-1} \\ \bar{Y} - \tilde{P}\Delta\underset{\leftarrow}{\tilde{\boldsymbol{u}}} - \tilde{Q}\underset{\leftarrow}{\tilde{\boldsymbol{y}}} \\ -\underline{Y} + \tilde{P}\Delta\underset{\leftarrow}{\tilde{\boldsymbol{u}}} + \tilde{Q}\underset{\leftarrow}{\tilde{\boldsymbol{y}}} \end{bmatrix} \tag{13}$$

qpOASES (quadratic program Online Active SEt Strategy) is an open-source implementation of the recently proposed online active set strategy, which was inspired by important observations from the field of parametric quadratic programming [16] [17]. The standard form is

$$\begin{aligned} &\min_{\Delta\underset{\rightarrow}{\boldsymbol{u}}} J = \Delta\underset{\rightarrow}{\boldsymbol{u}}^{\mathrm{T}} S \Delta\underset{\rightarrow}{\boldsymbol{u}} + 2\Delta\underset{\rightarrow}{\boldsymbol{u}}^{\mathrm{T}} \boldsymbol{f} \\ &\text{s.t. } lb \le \Delta\underset{\rightarrow}{\boldsymbol{u}} \le ub \\ &lbA \le A\Delta\underset{\rightarrow}{\boldsymbol{u}} \le ubA \end{aligned}$$

The (6) can be converted to be in the qpOASES form. And the followed can be derived from (9), (11), (12)

$$\underbrace{\Delta\underline{U}}_{lb} \le \Delta\underset{\rightarrow}{\boldsymbol{u}} \le \underbrace{\Delta\bar{U}}_{ub}$$

$$\left(\underline{U} - L\boldsymbol{u}_{k-1}\right) \le \boldsymbol{C}_{I/\Delta}\Delta\underset{\rightarrow}{\boldsymbol{u}} \le \left(\bar{U} - L\boldsymbol{u}_{k-1}\right)$$

$$\left(\underline{Y} - \tilde{P}\Delta\underset{\leftarrow}{\tilde{\boldsymbol{u}}} - \tilde{Q}\underset{\leftarrow}{\tilde{\boldsymbol{y}}}\right) \le \Gamma\Delta\underset{\rightarrow}{\boldsymbol{u}} \le \left(\bar{Y} - \tilde{P}\Delta\underset{\leftarrow}{\tilde{\boldsymbol{u}}} - \tilde{Q}\underset{\leftarrow}{\tilde{\boldsymbol{y}}}\right)$$

And we can get

$$\underbrace{\begin{bmatrix} \underline{U} - L\boldsymbol{u}_{k-1} \\ \underline{Y} - \tilde{P}\Delta\underset{\leftarrow}{\tilde{\boldsymbol{u}}} - \tilde{Q}\underset{\leftarrow}{\tilde{\boldsymbol{y}}} \end{bmatrix}}_{lbA} \le \underbrace{\begin{bmatrix} \boldsymbol{C}_{I/\Delta} \\ \Gamma \end{bmatrix}}_{A} \Delta\underset{\rightarrow}{\boldsymbol{u}} \le \underbrace{\begin{bmatrix} \bar{U} - L\boldsymbol{u}_{k-1} \\ \bar{Y} - \tilde{P}\Delta\underset{\leftarrow}{\tilde{\boldsymbol{u}}} - \tilde{Q}\underset{\leftarrow}{\tilde{\boldsymbol{y}}} \end{bmatrix}}_{ubA}$$

Algorithm 1-GPC with constraints

1) Firstly, we can identify the plant $A\left(z^{-1}\right)$, $B\left(z^{-1}\right)$, and $\Gamma,\ \tilde{P},\ \tilde{Q}$ can be calculated from (3), (4), Specify the factor W_y, W_u, $T\left(z^{-1}\right)$, receding horizon N_y, control step N_u, bound of input $\underline{U},\ \bar{U}$, bound of input rate $\Delta\underline{U},\ \Delta\bar{U}$, bound of output $\underline{Y},\ \bar{Y}$, calculate the $lb,\ ub,\ A$, initialize QProblem object (qpOASES).

2) sample the output, update $\underset{\leftarrow}{\boldsymbol{y}}$.

3) update $lbA,\ ubA$ and QProblem object, return the optimization value $\Delta\underset{\rightarrow}{\boldsymbol{u}}$.

4) update $\Delta\underset{\leftarrow}{\boldsymbol{u}}$, output $\Delta\boldsymbol{u}_k$.

5) go to (2).

3. Cascaded Generalized Predictive Control

As shown in **Figure 2**, Inner process model is described as

$$A_1\left(z^{-1}\right)\boldsymbol{v} = B_1\left(z^{-1}\right)\boldsymbol{u} + \frac{T_1\left(z^{-1}\right)}{\Delta}\boldsymbol{e}_1 \tag{14}$$

And outer process model is described as

$$A_2\left(z^{-1}\right)\boldsymbol{y} = B_2\left(z^{-1}\right)\boldsymbol{v} + \frac{T_2\left(z^{-1}\right)}{\Delta}\boldsymbol{e}_2 \tag{15}$$

Similar to Formula (1) and (2), future predictions for Formula (14) is

$$\Delta\underset{\rightarrow}{\boldsymbol{v}} = \Gamma_1\Delta\underset{\leftarrow}{\boldsymbol{u}} + \tilde{P}_1\Delta\underset{\leftarrow}{\boldsymbol{u}} + \tilde{Q}_1\Delta\underset{\leftarrow}{\tilde{\boldsymbol{v}}} \tag{16}$$

$\tilde{\square}$ means filter by T_1

$$\Gamma_1 = C_{A_1}^{-1}C_{B_1}, P_1 = C_{A_1}^{-1}H_{B_1}, Q_1 = -C_{A_1}^{-1}H_{A_1}$$

$$\begin{cases} \tilde{P}_1 = C_{T_1}P_1 - \Gamma_1 H_{T_1} \\ \tilde{Q}_1 = C_{T_1}Q_1 + H_{T_1} \end{cases}$$

And the future predictions for Formula (15) is

$$\underset{\leftarrow}{\boldsymbol{y}} = \Gamma_2\Delta\underset{\rightarrow}{\boldsymbol{v}} + \tilde{P}_2\Delta\underset{\leftarrow}{\dot{\tilde{\boldsymbol{v}}}} + \tilde{Q}_2\underset{\leftarrow}{\dot{\tilde{\boldsymbol{y}}}} \tag{17}$$

$\dot{\tilde{\square}}$ means filter by T_2

$$\Gamma_2 = C_{D_2}^{-1}C_{B_2}, P_2 = C_{D_2}^{-1}H_{B_2}, Q_2 = -C_{D_2}^{-1}H_{D_2}$$

$$\begin{cases} \tilde{P}_2 = C_{T_2}P_2 - \Gamma_2 H_{T_2} \\ \tilde{Q}_2 = C_{T_2}Q_2 + H_{T_2} \end{cases}$$

By substitution of Formula (16) into Formula (17)

$$\underset{\rightarrow}{\boldsymbol{y}} = \Gamma_2\left(\Gamma_1\Delta\underset{\rightarrow}{\boldsymbol{u}} + \tilde{P}_1\Delta\underset{\leftarrow}{\tilde{\boldsymbol{u}}} + \tilde{Q}_1\Delta\underset{\leftarrow}{\tilde{\boldsymbol{v}}}\right) + \tilde{P}_2\Delta\underset{\leftarrow}{\dot{\tilde{\boldsymbol{v}}}} + \tilde{Q}_2\underset{\leftarrow}{\dot{\tilde{\boldsymbol{y}}}} = \Gamma_2\Gamma_1\Delta\underset{\rightarrow}{\boldsymbol{u}} + \Gamma_2\tilde{P}_1\Delta\underset{\leftarrow}{\tilde{\boldsymbol{u}}} + \Gamma_2\tilde{Q}_1\Delta\underset{\leftarrow}{\tilde{\boldsymbol{v}}} + \tilde{P}_2\Delta\underset{\leftarrow}{\dot{\tilde{\boldsymbol{v}}}} + \tilde{Q}_2\underset{\leftarrow}{\dot{\tilde{\boldsymbol{y}}}} = \Gamma\Delta\underset{\rightarrow}{\boldsymbol{u}} + \boldsymbol{y}_{\text{free}}$$

where, $\Gamma = \Gamma_2\Gamma_1$, $\boldsymbol{y}_{\text{free}} = \Gamma_2\tilde{P}_1\Delta\underset{\leftarrow}{\tilde{\boldsymbol{u}}} + \Gamma_2\tilde{Q}_1\underset{\leftarrow}{\tilde{\boldsymbol{v}}} + \tilde{P}_2\Delta\underset{\leftarrow}{\dot{\tilde{\boldsymbol{v}}}} + \tilde{Q}_2\underset{\leftarrow}{\dot{\tilde{\boldsymbol{y}}}}$

The control law without constrains corresponds with (8).

The control law with constrains for intermediate output

$$\begin{aligned} \underset{\rightarrow}{\boldsymbol{v}} &= \Gamma_3\Delta\underset{\rightarrow}{\boldsymbol{u}} + \tilde{P}_3\Delta\underset{\leftarrow}{\tilde{\boldsymbol{u}}} + \tilde{Q}_3\underset{\leftarrow}{\tilde{\boldsymbol{v}}} = \Gamma_3\Delta\underset{\rightarrow}{\boldsymbol{u}} + \boldsymbol{v}_{\text{free}} \\ \boldsymbol{v}_{\text{free}} &= \tilde{P}_3\Delta\underset{\leftarrow}{\tilde{\boldsymbol{u}}} + \tilde{Q}_3\Delta\underset{\leftarrow}{\tilde{\boldsymbol{v}}} \end{aligned} \tag{18}$$

$\tilde{\square}$ means filter by T_1

$$\begin{aligned}
&\Gamma_1 = C_{D_1}^{-1} C_{B_1}, P_1 = C_{D_1}^{-1} H_{B_1}, Q_1 = -C_{D_1}^{-1} H_{D_1} \\
&\begin{cases} \tilde{P}_3 = C_{T_1} P_3 - \Gamma_3 H_{T_1} \\ \tilde{Q}_3 = C_{T_1} Q_3 + H_{T_1} \end{cases} \\
&\underline{V} \le v \le \bar{V}
\end{aligned} \tag{19}$$

$\underline{V}$ is the lower bounds of intermediate output v constraints, and $\bar{V}$ is the upper bounds. By substitution of Formula (18) into Formula (19)

$$\underline{V} \le \Gamma_3 \Delta \underset{\rightarrow}{\boldsymbol{u}} + \tilde{P}_3 \Delta \underset{\leftarrow}{\tilde{\boldsymbol{u}}} + \tilde{Q}_3 \underset{\leftarrow}{\tilde{\boldsymbol{v}}} \le \bar{V}$$

The corresponding linear inequalities are:

$$\begin{bmatrix} \Gamma_3 \\ -\Gamma_3 \end{bmatrix} \Delta \underset{\rightarrow}{\boldsymbol{u}} - \begin{bmatrix} \bar{V} - \tilde{P}_3 \Delta \underset{\leftarrow}{\tilde{\boldsymbol{u}}} - \tilde{Q}_3 \underset{\leftarrow}{\tilde{\boldsymbol{v}}} \\ -\underline{V} + \tilde{P}_3 \Delta \underset{\leftarrow}{\tilde{\boldsymbol{u}}} + \tilde{Q}_3 \underset{\leftarrow}{\tilde{\boldsymbol{v}}} \end{bmatrix} \le 0$$

$$\boldsymbol{C} \Delta \underset{\rightarrow}{\boldsymbol{u}} - \boldsymbol{d}_k \le 0$$

where,

$$\boldsymbol{C} = \begin{bmatrix} I \\ -I \\ \boldsymbol{C}_{I/\Delta} \\ -\boldsymbol{C}_{I/\Delta} \\ \Gamma_3 \\ -\Gamma_3 \end{bmatrix}, \quad \boldsymbol{d}_k = \begin{bmatrix} \Delta \bar{U} \\ -\Delta \underline{U} \\ \bar{U} - L\boldsymbol{u}_{k-1} \\ -\underline{U} + L\boldsymbol{u}_{k-1} \\ \bar{V} - \boldsymbol{v}_{\text{free}} \\ -\underline{V} + \boldsymbol{v}_{\text{free}} \end{bmatrix}$$

The control law with constrains is converted to be standard quadratic programming with constraints problem and can be solved by Algorithm 1.

4. Classical RLS Method

The SISO system can be identified by classical RLS method, which is described as followed.

$$\begin{aligned}
&\tilde{y}_k = \boldsymbol{\theta}^{\mathrm{T}} \boldsymbol{\varphi} + e_k \\
&\boldsymbol{\theta} = \begin{bmatrix} a_1 & \cdots & a_n & b_1 & \cdots & b_m \end{bmatrix}^{\mathrm{T}} \\
&\boldsymbol{\varphi} = \begin{bmatrix} \tilde{y}_{k\text{-}1} & \cdots & \tilde{y}_{k\text{-}n} & \tilde{u}_{k\text{-}1} & \cdots & \tilde{u}_{k\text{-}m} \end{bmatrix}^{\mathrm{T}}
\end{aligned} \tag{20}$$

$\tilde{\square}$ means filter by T in Equation (1). where θ is a vector of adjustable model parameters and e_k is the corresponding error at time k. The aim is to select θ so that overall modeling error is minimized. The classical recursive least square algorithm is

$$\begin{aligned}
&\boldsymbol{P}_k = \left[1 - \boldsymbol{\varphi} \left[\boldsymbol{\varphi}^{\mathrm{T}} \boldsymbol{P}_{k-1} \boldsymbol{\varphi} + \mu \right]^{-1} \boldsymbol{\varphi}^{\mathrm{T}} \right] \boldsymbol{P}_{k-1} \\
&\hat{\boldsymbol{\theta}}_{\mathrm{k}} = \hat{\boldsymbol{\theta}}_{\mathrm{k-1}} + \boldsymbol{K} (\tilde{y}_{\mathrm{k}} - \hat{\boldsymbol{\theta}}_{\mathrm{k-1}} \boldsymbol{\varphi}^{\mathrm{T}}) \\
&\boldsymbol{K} = \boldsymbol{P}_k \boldsymbol{\varphi} / \mu
\end{aligned} \tag{21}$$

In the RLS algorithm, the $\boldsymbol{P}_k$ update equation Equation (21) is very sensitive to the truncation errors and there is no guarantee that $\boldsymbol{P}_k$ will always be positive and symmetric defined. The parameter identified by RLS will have biases from the true parameters, the stability and robustness of the algorithm is very poor. And then $\boldsymbol{P}_k$ is revised as

$$\boldsymbol{P}_k = \left(\boldsymbol{P}_k + \boldsymbol{P}_k^{\mathrm{T}} \right) / 2$$

5. Verification on Proposed Scheme

Two examples are presented here to illustrate the effectiveness of the proposed tuning method for cascade control systems shown in **Figure 2**. The parameters of the ACGPC are in the **Table 1**.

In simulation, there is an identification process at first 100 s. Both inner and an outer model are identified by the classical RLS, respectively. The final value of model identified by the RLS is shown in the **Table 2** and **Table 3**, respectively. And the **Figures 3-6** show model parameters estimated by RLS. There is rarely fluctuation in convergence of parameter estimated. Based on the model, the CGPC is applied to control the cascade system. The outer process outputs (primary outputs) of the control loops are presented in **Figures 5-8**. The figure clearly shows the CGPC have excellent transient performance. This implies that all the good properties of the cascade control are kept in the ACGPC. This behavior is specific to the cascade structures. And the ACGPC can auto-tune the CGPC.

Table 1. Numerical simulation parameters setting.

parameters	Symbol	Value
GPC	Sample time T_s	1 s
	$T_1\left(z^{-1}\right)$	$\left(1-0.9\,z^{-1}\right)$
	$T_1\left(z^{-1}\right)$	$\left(1-0.9\,z^{-1}\right)$
	Output weighting factor W_y	1
	input weighting factor W_u	0.4
	predictive horizon N_y	20
	control horizon N_u	3
	maximum rate $\Delta\bar{\boldsymbol{u}}$	1
	minimum rate $\Delta\underline{\boldsymbol{u}}$	−1
	maximum level limits $\bar{\boldsymbol{u}}$	2.5
	minimum level limits $\underline{\boldsymbol{u}}$	−2.5
RLS	μ	0.95
	θ_0	$0.1 * 1_{3\times1}$

Table 2. Identification result of RLS for the real model.

Symbol	Value
Inner loop model A_1	$\left[1.0000 \quad -0.9048\right]$
Inner loop model B_1	$\left[0.0952\right]$
Outer loop model A_2	$\left[1.0000 \quad -0.9458\right]$
Outer loop model B_2	$\left[0.03\right]$

Table 3. Identification result of RLS for the real model.

Symbol	Value
Inner loop model A_1	$\left[1.0000 \quad -0.8187\right]$
Inner loop model B_1	$\left[0.1813\right]$
Outer loop model A_2	$\left[1.0000 \quad -0.9625\right]$
Outer loop model B_2	$\left[0.0202\right]$

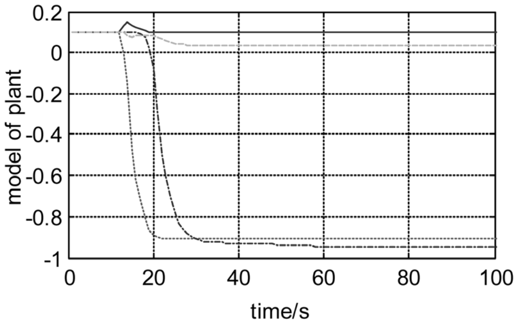


Figure 3. Identification result of RLS for the real model $G_1(s)$ and $G_2(s)$.

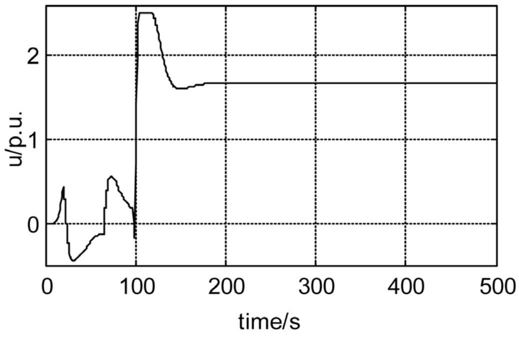


Figure 4. The control signals with and with control input constraints.

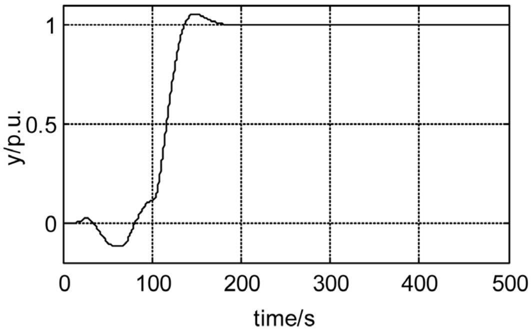


Figure 5. The tracking and regulation performance of the CGPC.

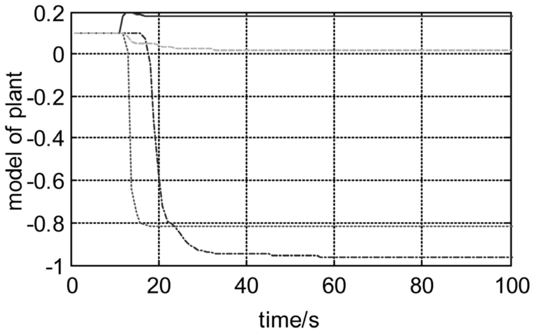


Figure 6. Regulation performance of the CGPC under inner disturbance disturbance which zoomed in **Figure 5**.

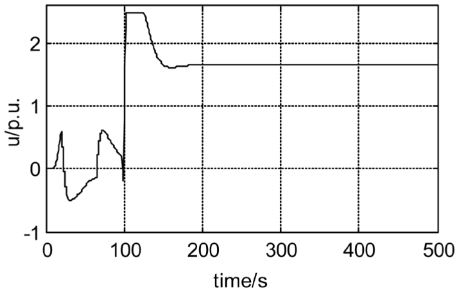


Figure 7. The control signals with and with control input constraints.

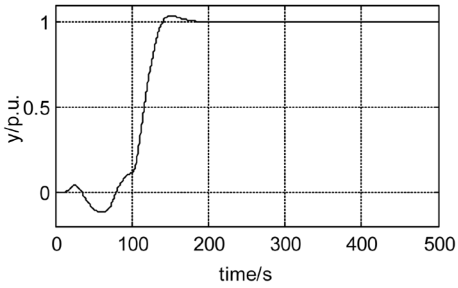


Figure 8. The control signals with and with control input constraints.

- **Example 1**

  The inner process is:

$$G_1(s) = \frac{1}{10s+1}$$

  The outer process is:

$$G_2(s) = \frac{0.6}{20s+1}$$

- **Example 2**

  The inner process is:

$$G_1(s) = \frac{1}{5s+1}$$

  The outer process is:

$$G_2(s) = \frac{0.6}{30s+1}$$

6. Conclusion

This paper developed an ACGPC method for the cascade control system, which gives the possibility to identify and control some different variables together. Both inner loop and outer loop process model, parameters can be identified using classical RLS method. Consequently, well-identified model based on CGPC can be applied to cascade control system. Finally, two examples were given to show the effectiveness of the proposed method. The method is very straightforward and has been integrated into an existing auto-tuning system. It is now being tested in an electrical drives system and the field results will be reported soon.

Acknowledgements

This work was supported by Henan University Science Foundation 2013YBZR013 and National Natural Science Foundation of China with grant number 61273174.

References

[1] Pisano, A., Davila, A., Fridman, L., *et al*. (2008) Cascade Control of PM DC Drives via Second-Order Sliding-Mode Technique. *IEEE Transactions on Industrial Electronics*, **55**, 3846-3854. http://dx.doi.org/10.1109/TIE.2008.2002715

[2] Tsang, K.M. and Chan, W.L. (2008) Non-Linear Cascade Control of DC/DC Buck Converter. *Electric Power Components and Systems*, **36**, 977-989. http://dx.doi.org/10.1080/15325000801960937

[3] Wolff Erik, A. and Skogestad, S. (1996) Temperature Cascade Control of Distillation Columns. *Industrial & Engineering Chemistry Research*, **35**, 475-484.

[4] Hang, C.C., Loh, A.P. and Vasnani, V.U. (1994) Relay Feedback Auto-Tuning of Cascade Controllers. *IEEE Transactions on Control Systems Technology*, **2**, 42-45. http://dx.doi.org/10.1109/87.273109

[5] Vivek, S. and Chidambaram, M. (2004) Cascade Controller Tuning by Relay Auto Tune Method. *Journal of the Indian Institute of Science*, **84**, 89-97.

[6] Leva, A. and Donida, F.F. (2009) Autotuning in Cascaded Systems Based on a Single Relay Experiment. *Journal of Process Control*, **19**, 896-905. http://dx.doi.org/10.1016/j.jprocont.2008.11.013

[7] Song, S., Cai, W. and Wang, Y.G. (2003) Auto-Tuning of Cascade Control Systems. *ISA Transactions*, **42**, 62-72. http://dx.doi.org/10.1016/S0019-0578(07)60114-1

[8] Tan, K.K., Lee, T.H. and Ferdous, R. (2000) Simultaneous Online Automatic Tuning of Cascade Control for Open Loop Stable Processes. *ISA Transactions*, **39**, 233-242. http://dx.doi.org/10.1016/S0019-0578(00)00006-9

[9] Visioli, A. and Piazzi, A. (2006) An Automatic Tuning Method for Cascade Control Systems. *Proceedings of the IEEE International Conference on Control Applications*, Munich, July 2006, 2968-2973.

[10] Veronesi, M. and Visioli, A. (2011) Simultaneous Closed-Loop Automatic Tuning Method for Cascade Controllers. *Control Theory and Applications*, **5**, 263-270.

[11] Benyó, I. (2006) Cascade Generalized Predictive Control—Applications in Power Plant Control. University of Oulu, Oulu.

[12] Benyó, I. (2007) Cascade Generalized Predictive Controller: Two in One. *International Journal of Control*, **79**, 866-876.

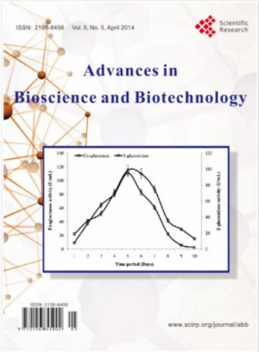
[13] Clarke, D.W., Mohtadi, C. and Tuffs, P.S. (1987) Generalized Predictive Control—Part II. Extensions and Interpretations. *Automatica*, **23**, 149-160. http://dx.doi.org/10.1016/0005-1098(87)90088-4

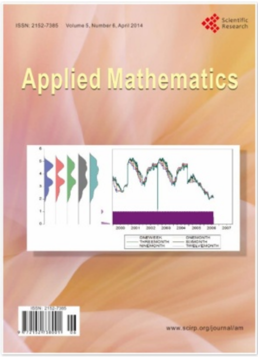
[14] Clarke, D.W., Mohtadi, C. and Tuffs, P.S. (1987) Generalized Predictive Control—Part I. The Basic Algorithm. *Automatica*, **23**, 137-148. http://dx.doi.org/10.1016/0005-1098(87)90087-2

[15] Rossiter, J.A. (2003) Model-Based Predictive Control-A Practical Approach. CRC Press, Washington DC, 54-58.

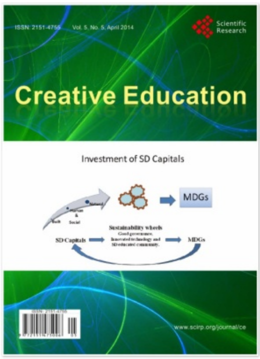
[16] Ferreau, H.J. (2009) qpOASES (Quadratic Program Online Active SEt Strategy) Version 2.0. http://www.qpoases.org/

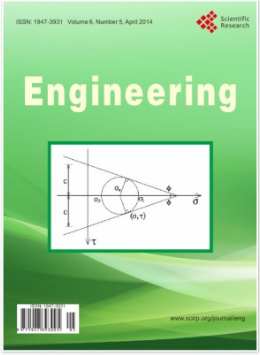
[17] Ferreau, H.J., Bock, H.G. and Diehl, M. (2006) An Online Active Set Strategy for Fast Parametric Quadratic Programming in MPC Applications. *Proceedings of the IFAC Workshop on Nonlinear Model Predictive Control for Fast Systems*, Grenoble, 2006, 21-30.


Advances in
Bioscience and Biotechnology
www.scirp.org/journal/abb

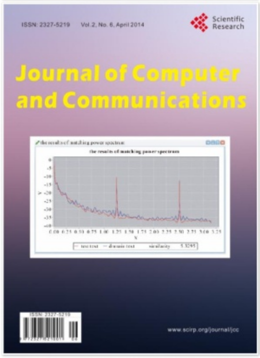

Applied Mathematics
www.scirp.org/journal/am

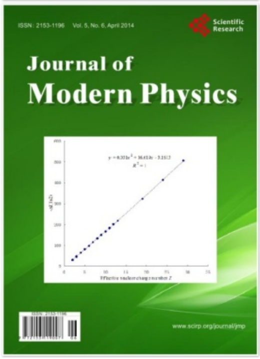
American Journal of
Plant Sciences
www.scirp.org/journal/ajps


Creative Education
Investment of SD Capitals
MDGs
www.scirp.org/journal/ce

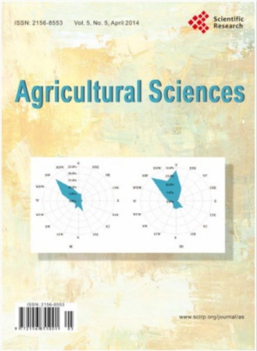

Engineering
www.scirp.org/journal/eng

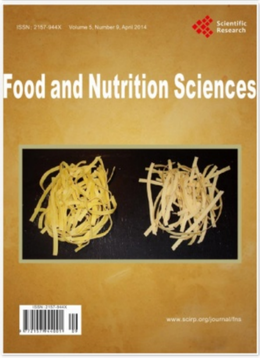
HEALTH
Editor-in-Chief
Prof. James J. Chou
www.scirp.org/journal/health

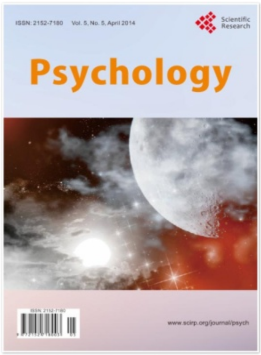

Journal of Computer
and Communications
www.scirp.org/journal/jcc


Journal of
Modern Physics
www.scirp.org/journal/jmp

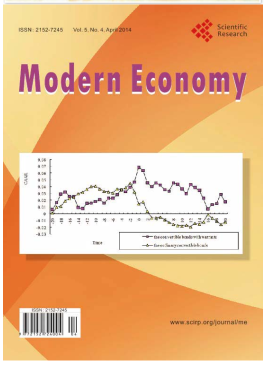
Journal of
Environmental Protection


Agricultural Sciences
www.scirp.org/journal/as


Food and Nutrition Sciences
www.scirp.org/journal/fns


Psychology
www.scirp.org/journal/psych

Natural
Science
Editor-in-Chief
Kuo-Chen Chou


Modern Economy
www.scirp.org/journal/me

Journal of Cancer Therapy
www.scirp.org/journal/jct

American Journal of
Analytical Chemistry
www.scirp.org/journal/ajac