A FREQUENCY DOMAIN BASED SELF-TUNING PID CONTROLLER

J.V. Ringwood1 and A. O'Dwyer2

1School of Electronic Engineering
Dublin City University
Dublin 9, IRELAND

2Dept. of Control Systems and Electrical Eng.
Dublin Institute of Technology
Kevin St., Dublin 8, IRELAND

Abstract. Traditionally, both explicit and implicit self-tuning controllers have employed time domain techniques for the identification and tracking of plant and controller parameters. The use of the frequency domain offers certain advantages compared to their time domain counterparts. This paper demonstrates methods employing recursive, on-line measurement of the process frequency response, with a straightforward calculation of PID controller parameters. The computational effort involved is comparable with that of a time domain technique.

Keywords. Self-tuning regulator, Frequency domain, PID control

I. INTRODUCTION

PID self-tuning algorithms utilising time domain identification techniques are widely reported in the literature. Traditionally, such methods incorporate some form of time domain identification based on a parameterised model with a set of design equations enabling controller parameters to be updated in real-time. Examples include methods by Banzoiz and Keviczky (1982) and Tjakra and Abhishek (1985). This paper, however, introduces a new approach to improve the self-tuning process. A further difficulty is the time delay between the identification of the process frequency domain and the parameterised models. An explicit delay term cannot be incorporated into linear parameterised models: frequency domain model parameters become impractical for more than 6 signal samples delay and non-linear schemes have achieved limited success (Durbin 1985).

Another significant factor in designing PID self-tuners is the difficulty of relating PID parameters to process transfer function parameters. This difficulty is often exacerbated by the frequency domain, where the design equations are not as straightforward. To avoid this, the self-tuning process can be approximated as follows: (1)

\[ G(p) = C(p)/N(p) \]

with \( C(p) \) and \( N(p) \) being the Fourier transforms of \( c(t) \) and \( n(t) \), respectively. A recursive technique for calculating the product of the transforms is appropriate. One such method is to use the Discrete Time Fourier Transform (DTFT), defined as follows:

\[ F(x) = \sum_{k=-\infty}^{\infty} f(k)e^{-j2\pi kx} \]

This transform has the advantage that a new term may be added as new data points become available; a further advantage is that the frequency variable is continuous, which allows for more accurate calculation of the phase crossover frequency. The DTFT could be simplified by taping into the data window at the start of each evaluation period of the summation; this proposal would reduce spectral leakage. The inclusion of a non-rectangular data window would however increase the computational complexity of the calculation. An alternative recursive method for finding the transforms is to apply a numerical integration technique to the Fourier transform. An example of suitable techniques is the Adams-Moulton set, as discussed by Johnson and Raine (1982). The first four of this set are as follows:
\[ F_{31} - F_i = T x_{31} \]  
\[ F_{32} - F_{31} = \frac{1}{T} (x_{32} + x_{31}) \]  
\[ F_{33} - F_{32} = \frac{1}{T} (5x_{33} + 8x_{32} + x_{31}) \]  
\[ F_{34} - F_{33} = \frac{1}{T} (9x_{34} + 19x_{33} - 5x_{32} + x_{31}) \]

where \( x_i = f(kT) e^{-\lambda kT} \).

Equations (3) and (4) may be readily identified as the backward difference and trapezoidal rule (bilinear transform) respectively. Assuming a start from \( k=0 \) and zero initial conditions, the first four terms of the integrals in (3) and (4) become:

\[ I_1 = \int_0^t (x_i + x_{i-1}) e^{-\lambda (i-1)T} dt \]  
\[ I_2 = \int_0^t (x_i + x_{i-1}) e^{-\lambda (i-1)T} dt + x_{i-1} e^{-\lambda (i-1)T} \]  
\[ I_3 = \int_0^t (x_i + x_{i-1}) e^{-\lambda (i-1)T} dt + x_{i-1} e^{-\lambda (i-1)T} + \frac{x_{i-2}}{e^{-\lambda T}} \]  
\[ I_4 = \int_0^t (x_i + x_{i-1}) e^{-\lambda (i-1)T} dt + x_{i-1} e^{-\lambda (i-1)T} + \frac{x_{i-2}}{e^{-\lambda T}} + \frac{x_{i-3}}{e^{-2\lambda T}} \]

Note that (7) displays a DTFT. However, (8) demonstrates a DTFT with a data window which is tapered at each end. Higher order techniques exaggerate this windowing effect.

2.2 Beat Frequencies

From the definition of the DTFT in equation (2), it can be seen that product terms arise between sinusoidal signals in \( f(kT) \) and the exponential term. Since an average (or sum) of the product of sinusoids of different frequencies is zero, the only term which is non-zero is the product term involving a sinusoidal at the DTFT frequency. This \( \sin^2(kT) \) term may be recast into a \( \frac{1}{2} (1 - \cos(2\pi kT)) \) term, involving a beat frequency at twice the DTFT frequency. A difference equation for the phase of the system evaluated using the DTFT can be found as:

\[ \phi_i = \phi_{i-1} - \frac{1}{T} \left\{ \frac{\cos(\Phi) - \cos(2\pi kT + \Phi)}{\sin(\Phi) + \sin(2\pi kT + \Phi)} \right\} \]

After convergence, \( \phi_i = \phi_{i-1} = \Phi \) (on average), but the phase measurement continues to vary according to the latter two terms in (9) which involve the beat frequency. However, as the DTFT frequency approaches the phase crossover frequency, where \( \Phi - \beta \gg 1 \), a trivial calculation shows that these terms go to zero. It may be demonstrated that an attenuation inversely proportional to the difference between the DTFT frequency and the phase crossover frequency is achieved.

Low pass filters on gain and phase estimates are used to reduce the effect of beat frequencies. These are based on first order differences and have a cut-off frequency below \( 2\omega_c \). Alternatively, band pass filters or filters with a variable cut-off frequency could be employed for improved performance.

2.3 Data Forgetting

An important feature of either of the recursive schemes outlined above is that new terms are constantly being added as time progresses. This may lead to two difficulties:

(a) The size of the DTFT's may become very large, and
(b) The algorithm may become insensitive to changes in the process dynamics or evaluation frequency, due to the magnitude difference between the new terms being added and the current size of this transform.

The magnitude difference in (b) is typically of the order of \( 10^9 \). A form of data forgetting may be implemented to maintain a reasonable balance between the orders of magnitude of the transforms and their increments. An example of such a method involves weighting the data values by progressively smaller amounts as they recede in time. A forgetting factor, \( \lambda \), is introduced as follows:

\[ R_{31}(\omega) = \lambda R_i + g(x) \]

The first order DTFT with a rectangular data window has the form:

\[ R_{31}(\omega) = \lambda R_i(\omega) + T x_i \]

with \( 0 < \lambda \leq 1 \).

2.4 Identification in Closed Loop

To aid identification in closed-loop, an excitation signal at the appropriate (Fourier transform), frequency is added to the control signal. This signal, while not having any adverse effects on the regulation properties of the system, would seem to be sufficient to allow consistent identification of the open-loop frequency response in closed-loop, based on a related analysis by Wallsteadt (1986). The amplitude, \( A_\omega \), of the sinusoidal excitation signal should be commensurate with the amplitude of the measurement noise at \( d(t) \). This excitation signal is preferable, from a regulation point of view, to the sharp- edged excitation signals associated with time-domain identification.

A further practical addition of band-pass filters with moveable centre frequency is included to concentrate calculations on the frequency range of interest. This helps to improve the disturbance and noise rejection properties of the adaptation algorithm. A Butterworth design is used with transfer function:

\[ G_p(s) = \frac{1}{s^2 - (1+\alpha)\beta s + \alpha} \]

where

\[ \beta = \frac{\cos(\alpha \omega_c)}{\cos(\frac{\alpha \omega_c}{2})} \]

\( \alpha \) is a parameter determined from the equivalent low-pass design and depends only on the filter bandwidth, \( \omega_c \), and the sampling period, \( T_0 \), is the centre frequency of the band-pass filter.

---

*Fig. 1. Block diagram of the closed loop system.*
\[ \frac{K_{21} - K_{22}}{K_{21} + K_{22}} = \frac{K_{31} - K_{32}}{K_{31} + K_{32}} \]
\[ \frac{K_{23} - K_{24}}{K_{23} + K_{24}} = \frac{K_{33} - K_{34}}{K_{33} + K_{34}} \]
\[ \frac{K_{25} - K_{26}}{K_{25} + K_{26}} = \frac{K_{35} - K_{36}}{K_{35} + K_{36}} \]

Equations (3) and (4) may be readily identified as the backward difference and trapezoidal rules (implicit trapezoidal) respectively. Assuming a small angle, one can simplify the expressions for the trapezoidal and their increments. An example of such a method involves weighting the data values by progressively smaller amounts as the time proceeds. A forgetting factor, \( \lambda \), is introduced as follows:

\[ K_{1}(o) = \lambda F_{1} + g(x) \]

The first order DFTT with a rectangular data window has the form:

\[ X_{k}(o) = \frac{1}{T} \sum_{n=0}^{T} x_{n} e^{-j2\pi k n / T} \]

where \( o \) is the frequency of interest and \( T \) is the time period.

2.2 Fast Fourier Analyser

From the definition of the DFTT in equation (2), it can be seen that the product terms arise from sinusoidal signals in the input signal and the complex exponential terms. Since the average or sum of the product of sinusoids of different frequencies is zero, the only term which is not zero is the product term involving a sinusoid of the DFT frequency. This sin integral term may be recast into a \( e^{j \omega T} \) form, involving a beat frequency twice the DFT frequency. A difference equation for the phase of the system evaluated using the DFTT can be found as:

\[ (\alpha o - 2 \pi / T) \approx \alpha o - 2 \pi / T \]

Where \( \alpha o \) is the angle, \( \alpha \) is the phase shift due to the difference between the two terms, and \( o \) is the sampling frequency.

3. FREQUENCY UPDATING

3.1 Update Method

The procedure for controller tuning discussed in Section 1 demonstrates that adjustment must be made to the evaluation of the Fourier transform until the phase crossover frequency is calculated. It is proposed to estimate the previous phase and frequency values to determine the phase crossover frequency. Gradient algorithms, which allow updating of the frequency based on the slope of the phase frequency curve, are appropriate for a large class of plants in which phase lag increases continuously as frequency increases. One such algorithm is the Least Mean Square (LMS) algorithm, as described by Widrow and Stearns (1985):

\[ a_{n+1} = a_{n} - 2 \mu \hat{e}_{n} \]

where \( \hat{e}_{n} \) is the current estimate of the phase crossover frequency, \( \mu \) is the current estimate of the phase crossover frequency, and \( \hat{e}_{n} \) is the estimated change in phase (with \( \hat{e}_{n} \) current phase estimate). The transfer function of the plant is unknown, then one approximation for \( \hat{e}_{n} / \hat{e}_{n+1} \) is:

\[ \frac{\hat{e}_{n}}{\hat{e}_{n+1}} = \frac{\hat{e}_{n}}{\hat{e}_{n+1}} = \frac{\hat{e}_{n}}{\hat{e}_{n+1}} \]

In these circumstances, the algorithm becomes:

\[ a_{n+1} = a_{n} - 2 \mu (\pi - 1) (\hat{e}_{n} - \hat{e}_{n+1}) \]

Other more computationally intensive gradient algorithms that may be used include the steepest descent algorithm, the Conjugate Gradient algorithm, and the Least Mean Square algorithm (Ljung, 1987). In general, these algorithms would facilitate faster adaptation than would the LMS algorithm. An alternative approach to that discussed above is to use a number of data points and fit a high order polynomial for the phase to the data. The parameters of the polynomial could be found using an estimation strategy such as least squares. The general strategy of this type would be to fit a straight line to two data points; the updated estimate of the phase crossover frequency is then given by:

\[ a_{n+1} = a_{n} - 2 \mu (\pi - 1) (\hat{e}_{n} - \hat{e}_{n+1}) \]

where \( \mu \) is the step size and \( \pi \) is the radius of the circular frequency. The step size parameter determines the degree to which the controller will be affected. A value of 0.5 has been found to be appropriate. At initialization, small values of \( \hat{e}_{n} \) and \( \hat{e}_{n+1} \) are assigned. These values guarantee safe control. \( \hat{e}_{n+1} \) is the nominal controller gain.

4. CONTROL SYSTEM DESIGN

4.1 PID Controller Design

In the continuous time domain, the Ziegler-Nichols tuning rules are not applicable to the present case, since the plant is discrete. The discrete-time controller, which is the Ziegler-Nichols tuning rules for a discrete-time controller, is defined by:

\[ K_{c}(s) = K_{1} \frac{1}{T_{i}} \frac{1}{s} \]

with:

- \( K_{1} = 0.63 \frac{T_{i}}{T}_{p} \)
- \( T_{i} = 0.5 T_{p} \)
- \( T_{p} = 0.127 \)

5. RESULTS

The performance of the proposed algorithm is demonstrated using simulation tests. The model used for the process is:

\[ G_{p}(s) = \frac{0.1113 e^{-0.5 \pi/4} \sin(\pi)}{1 - 0.64 e^{-0.7 \pi/4}} \]

\[ G_{p}(s) = \frac{0.1113 e^{-0.5 \pi/4} \sin(\pi)}{1 - 0.64 e^{-0.7 \pi/4}} \]

For a sampling period of 0.2 sec. The following values for the design parameters were used:
6. CONCLUSIONS

A self-tuning controller has been presented, based on frequency-domain calculations. One advantage of this is that no process parameterisation is required. The computational effort is comparable with that of a time-domain self-tuner. In addition, the algorithm contains design parameters not dissimilar to a time-domain algorithm. These generally involve a trade-off between speed of tuning and noise immunity. One feature of the technique in this paper is the easy addition of caution control, since a direct measure of the tuning error i.e. error in estimate of \( \omega_{\text{cdd}} \) via \( (\pi+\theta_{\text{cdd}}) \) is available.

The algorithm could also be extended to include explicit time delay estimation, since this effect (linear phase shift with frequency) can be resolved from the overall gain and phase measurements. Such an extension is not possible with parametric time-domain schemes.

7. REFERENCES


