

11-1996

A Real-Time Self-Tuning Web Tension Regulation Scheme

Brian T. Boulter
Reliance Electric Corporation

Zhiqiang Gao
Cleveland State University, Z.GAO@csuohio.edu

Follow this and additional works at: https://engagedscholarship.csuohio.edu/enece_facpub

 Part of the [Controls and Control Theory Commons](#)

How does access to this work benefit you? Let us know!

Original Citation

Boulter, B. T., , & Gao, Z. (1996). A real-time self-tuning web tension regulation scheme. *Real-Time Systems*, 11(3), 265 - 287. doi:10.1007/BF00337682

Repository Citation

Boulter, Brian T. and Gao, Zhiqiang, "A Real-Time Self-Tuning Web Tension Regulation Scheme" (1996). *Electrical Engineering and Computer Science Faculty Publications*. 77.
https://engagedscholarship.csuohio.edu/enece_facpub/77

This Article is brought to you for free and open access by the Electrical and Computer Engineering Department at EngagedScholarship@CSU. It has been accepted for inclusion in Electrical Engineering and Computer Science Faculty Publications by an authorized administrator of EngagedScholarship@CSU. For more information, please contact library.es@csuohio.edu.

A Real-Time Self-Tuning Web Tension Regulation Scheme

BRIAN T. BOULTER

Systems Division, Reliance Electric Corporation, 24703 Euclid Avenue, Cleveland, Ohio 44117

ZHIQIANG GAO

Dept. of Electrical Engineering, Cleveland State University, Euclid Avenue at E 24th. Street, Cleveland, Ohio 44115

1. Introduction

WEB tension regulators are typically implemented with a PI controller as shown in Figure 5. The PI parameters are tuned to provide a stable responsive system for the entire range of product processed through the system. This results in a system that is de-tuned for a large range of products and optimally tuned for a small range of products.

Unknowns such as web damping, friction, and slippage make the derivation of analytical tuning algorithms difficult. The usual approach is to heuristically tune the PI controller on-site based on observed system performance. Typically this results in stable tension regulation until such time as a product with extreme physical parameters is processed through the system. All too often this results in unstable tension regulation which in turn requires re-tuning of the loop. A better approach would be to provide a self-tuning regulation scheme that obviates the need for empirically finding a single unique set of stable tuning parameters for the entire range of products processed through the system.

To this point adaptive and/or self-tuning regulation schemes have not been considered for web transport systems. This fact can be attributed to the complexity of the existing time-domain algorithms, and the associated hardware and software implementation difficulties. K. J. Astrom in (Astrom, et al. 1993; Astrom and Wittenmark 1989; and Astrom 1993) and I. D. Landau in (Landau 1993) summarize a variety of self-tuning regulation approaches using PID and feedback/feedforward techniques. Some approaches such as the MRAS approach require multiple iterations to converge to a desired regulator design. Other approaches require the derivation of a parametric model of the plant. In general most of these approaches are computationally complex. Knowledge based systems have been implemented utilizing fuzzy logic and other rule based approaches with some success (Shirley 1987; Shin and Cui 1991; and Lee 1990). These approaches require customized rules based on the given application, and are not practical for systems that are required to be tuned on-site with minimal customization. Recent developments in the field of self-tuning

control have led to some exploration of frequency domain loop shaping self-tuning algorithms. These algorithms are based on iterative procedures that minimize a cost function (Goderd, Wang and Cluett 1992; Tzes and Yurkovich 1992; and Whitfield 1986). The time required to converge to a solution cannot be known a-priori. In general, the above self-tuning techniques are not computationally efficient enough, nor is the algorithm execution time predictable enough, to be used in self-tuning regulation (STR) for web transport systems. Distributed Control Systems (DCS), today's standard industrial control platform, are implemented with constrained computational resources requiring efficient algorithms with predictable execution times.

The STR method proposed in this paper is neither complex nor difficult to implement. It identifies controller parameters through an interpolation algorithm that utilizes a desired open loop frequency response and the frequency response of the plant. The core of the algorithm is expressed in finding the least squares solution of a system of linear equations. It is, therefore, extremely efficient and predictable. For these reasons this particular STR algorithm is attractive for self-tuning implementations in a DSP based DCS platform. The self-tuning algorithm is extensible and can be extended to any SISO control scheme.

In order to introduce the reader to the proposed STR algorithm and this particular implementation, the self-tuning algorithm is derived in section 2. A web transport plant is described in section 3. The simulation results of the proposed regulator are presented in section 4. A discussion of real-time implementation considerations along with conclusions can be found in section 5.

2. Self Tuning Regulation Scheme

2.1. Background

The most prevalent method of SISO industrial control system design in use today is the loop shaping approach. This approach hinges upon the design of a controller that satisfies a set of loop gain specifications. It requires the derivation of an approximate linear plant model. To arrive at an acceptable design, skill and experience are often required. The design procedure is typically carried out in a cut and try manner. The tools most commonly used to perform the design are Bode and/or Nichols plots. A new algorithmic loop shaping design approach is described in this section, which reduces the design problem to solving a set of linear algebraic equations.

Given that the proposed design algorithm can be coded and executed autonomously by a computer, the possibility of performing loop shaping SISO controller design on-line in an autonomous self-tuning regulation scheme becomes feasible. The purpose of this paper is to describe one such scheme. Figure 1 is the block diagram of the proposed STR scheme.

An FFT algorithm is used to obtain the plant frequency response $P(j\omega)$. The STR algorithm makes use of the obtained frequency response and the loop shaping algorithm described in the following section to update the controller parameters. The anti-windup switch and stimulus block are implemented to provide a frequency rich autonomous stimulus to the plant for the purpose of obtaining a meaningful plant frequency response. In

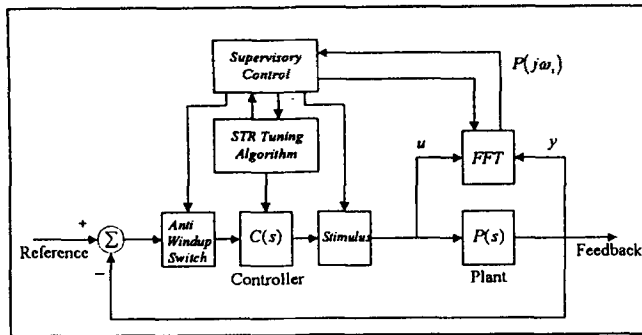


Figure 1. Proposed STR scheme.

other real-time applications a stimulus may not be required. This is discussed in the section covering real-time implementation issues.

2.2. A New Loop Shaping Algorithm

A popular approach used in loop shaping design translates desired closed-loop performance specifications to constraints on the loop gain. For example:

$$\begin{aligned} \text{Low Freq. Gain} &= 50 \text{ [db] at } 0.1 \text{ [rad/sec]} \\ \text{Crossover Freq.} &= 8 \text{ [rad/sec]} \\ \text{Phase Margin} &\geq 40^\circ \\ \text{High Freq. Atten.} &= -40 \text{ [db] at } 200 \text{ [rad/sec]} \end{aligned}$$

are typical specifications for a closed loop system with a step response time to peak of approximately 0.375 [sec], approximately 10% overshoot, negligible steady state error to a step, and good high frequency noise rejection and stability robustness. They can be considered typical for web tension regulation schemes.

While the above constraints are typical they are by no means all inclusive. A control system designer may want to design a controller such that non-typical constraints are accommodated. For example, stop band attenuation or pass band amplification at particular frequencies may be desired. The algorithm that is developed in the following derivation automates the design of a regulator to satisfy any such set of loop gain constraints. It minimizes the error, in a least squares sense, between any set of loop gain constraints and the loop gain resulting from the product of the measured plant frequency response and the tuned regulator. The design problem is formulated as a matrix interpolation problem. An important characteristic of the algorithm that separates it from current loop shaping approaches is that it is *not* iterative. The interpolation constraints are satisfied with controller parameters obtained by solving a system of linear equations.

Algorithm Derivation

A controller with a frequency response of $C(j\omega)$ can be designed for a given plant frequency response $P(j\omega)$ such that a desired loop gain $L(j\omega)$ is realized. That is:

$$L(j\omega) = C(j\omega)P(j\omega) \quad (1)$$

Let:

$$C(j\omega) = \frac{n_c(j\omega)}{d_c(j\omega)}$$

Where $n_c(j\omega)$, $d_c(j\omega)$ are the controller numerator and denominator polynomials respectively. (1) can be expressed as:

$$n_c(j\omega)P(j\omega) - d_c(j\omega)L(j\omega) = 0 \quad (2)$$

or:

$$[n_c(j\omega), -d_c(j\omega)] \begin{bmatrix} P(j\omega) \\ L(j\omega) \end{bmatrix} = 0 \quad (3)$$

The objective of the new STR algorithm is to find coefficients of the controller $C(j\omega)$ such that equations (1) through (3) hold at a set of chosen frequencies. The derivation of the STR algorithm is performed in two steps. The first step is to obtain a matrix equation based on (3) but expanded and expressed in terms of a finite set of independent frequencies. The second step is to perform matrix operations on the resulting equation such that a system of linear equations of the form $\mathbf{x}\mathbf{A} = \mathbf{b}$ is obtained.

Step 1) Represent (3) in terms of a finite set of independent frequencies

Represent the desired loop gain $L(j\omega)$ as a vector of complex scalars at l independent frequencies

For the example loop gain constraints given previously, the desired loop gain frequency response can be taken from Figure 2, and expressed as a vector of complex scalars taken at l frequencies:

$$[L(j\omega_i) |_{i=1,l}] = [L(j\omega_1), L(j\omega_2), \dots, L(j\omega_l)] \quad (4)$$

The frequencies in $L(j\omega)$ may also be weighted. Frequency weighting in the desired solution is described in (Gao and Anstaklis 1994) and (Gao 1993).

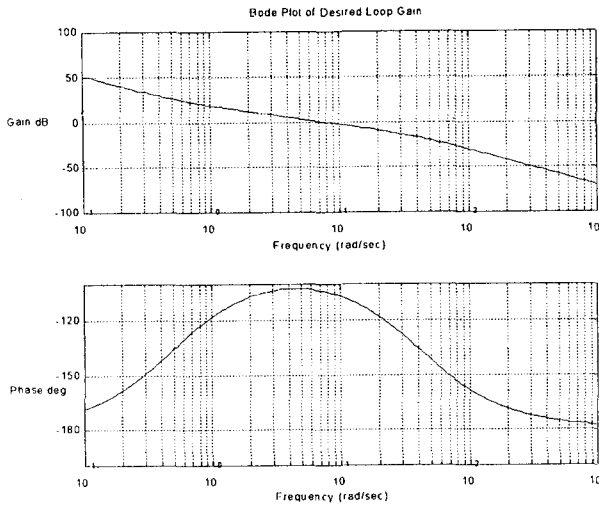


Figure 2. Example of a desired loop gain.

Represent the plant frequency response $P(j\omega)$ as a vector of complex scalars at l independent frequencies

The plant frequency response can be obtained via an FFT as a vector of complex scalars:

$$[P(j\omega_i) \mid_{i=1,l}] = [P(j\omega_1), P(j\omega_2), \dots, P(j\omega_l)] \quad (5)$$

The FFT algorithm must output a vector of complex scalars that is composed of elements taken at the same frequencies as those chosen for (4).

Represent the regulator $C(j\omega)$ as a matrix

Working with a controller representation in the s -domain, a regulator structure $C(s)$ is chosen for the given application. It must be proper (the order of the numerator is less than or equal to the order of the denominator and the leading coefficient of the denominator is 1).

$$C(s) = \frac{a_n s^n + a_{n-1} s^{n-1} + \dots + a_0}{s^m + b_{m-1} s^{m-1} + \dots + b_0}; n \leq m \quad (6)$$

The regulator can be expressed in matrix form in terms of its coefficients as:

$$[n_c(s), -d_c(s)] = [a_0, a_1, \dots, a_n, -b_0, -b_1, \dots, -b_{m-1}, -1] \begin{bmatrix} 1 & 0 \\ s & 0 \\ \vdots & \vdots \\ s^n & 0 \\ 0 & 1 \\ 0 & s \\ \vdots & \vdots \\ 0 & s^m \end{bmatrix} \quad (7)$$

Let C_p be the vector containing the parameters of the controller

$$C_p = [a_0, a_1, \dots, a_n, -b_0, -b_1, \dots, -b_{m-1}, -1] \quad (8)$$

and let:

$$S_i = \begin{bmatrix} 1 & 0 \\ j\omega_i & 0 \\ \vdots & \vdots \\ j^n \omega_i^n & 0 \\ 0 & 1 \\ 0 & j\omega_i \\ \vdots & \vdots \\ 0 & j^m \omega_i^m \end{bmatrix} \quad (9)$$

(7) can now be expressed in the frequency domain as:

$$[n_c(j\omega_i), -d_c(j\omega_i)] = C_p S_i \quad (10)$$

The plant frequency response $P(j\omega)$ and the desired loop gain response $L(j\omega)$ can be represented as C_i , a 2 row matrix with l columns corresponding to the number of frequency points specified in (4).

$$C_i = \begin{bmatrix} P(j\omega_1), P(j\omega_2), P(j\omega_3), \dots, P(j\omega_l) \\ L(j\omega_1), L(j\omega_2), L(j\omega_3), \dots, L(j\omega_l) \end{bmatrix} \quad (11)$$

Eq. (3) evaluated at ω_i , $i = 1, 2, \dots, l$ may now be represented as:

$$C_p S_i C_i = 0 \quad (12)$$

Where:

C_p is a $1 \times (m + n + 2)$ matrix

S_i is an $(m + n + 2) \times 2$ matrix

C_i is a $2 \times l$ matrix

Step 2) Perform matrix operations on (12) so that the resulting matrix equation is in the form $\mathbf{xA} \mathcal{D} \mathbf{b}$

Expanding $\mathbf{S}_i \mathbf{C}_i$ yields the following matrix

$$\mathbf{S}_i \mathbf{C}_i = \begin{bmatrix} P(j\omega_1) & P(j\omega_2) & \cdots & P(j\omega_l) \\ j\omega_1 P(j\omega_1) & j\omega_2 P(j\omega_2) & \cdots & j\omega_l P(j\omega_l) \\ \vdots & \vdots & \cdots & \vdots \\ j^n \omega_1^n P(j\omega_1) & j^n \omega_2^n P(j\omega_2) & \cdots & j^n \omega_l^n P(j\omega_l) \\ L(j\omega_1) & L(j\omega_2) & \cdots & L(j\omega_l) \\ j\omega_1 L(j\omega_1) & j\omega_2 L(j\omega_2) & \cdots & j\omega_l L(j\omega_l) \\ \vdots & \vdots & \cdots & \vdots \\ j^m \omega_1^m L(j\omega_1) & j^m \omega_2^m L(j\omega_2) & \cdots & j^m \omega_l^m L(j\omega_l) \end{bmatrix} \quad (13)$$

$\mathbf{S}_i \mathbf{C}_i$ may be represented as:

$$\mathbf{S}_i \mathbf{C}_i = \begin{bmatrix} \mathbf{S}_L \\ \mathbf{B}_L \end{bmatrix} \quad (14)$$

Where:

\mathbf{S}_L = the first $(n + m + 1)$ rows of the $\mathbf{S}_i \mathbf{C}_i$ matrix;
 $\mathbf{B}_L = [j^m \omega_1^m L(j\omega_1), j^m \omega_2^m L(j\omega_2), \dots, j^m \omega_l^m L(j\omega_l)]$

Given that the l th entry in \mathbf{C}_P is -1 (12) can be expressed as:

$$[\mathbf{C}_{\bar{P}} \quad -1] \begin{bmatrix} \mathbf{S}_L \\ \mathbf{B}_L \end{bmatrix} = 0 \quad (15)$$

where:

$$[\mathbf{C}_{\bar{P}}] = [a_0, \dots, a_n, -b_0, \dots, -b_{m-1}] \quad (16)$$

From (15) we obtain:

$$[\mathbf{C}_{\bar{P}}][\mathbf{S}_L] - [\mathbf{B}_L] = 0$$

and finally:

$$[\mathbf{C}_{\bar{P}}][\mathbf{S}_L] = [\mathbf{B}_L] \quad (17)$$

The matrix equation (17) is now in the desired form $\mathbf{xA} = \mathbf{b}$. (17) can now be expressed as:

$$[a_0, \dots, a_n, -b_0, \dots, -b_{m-1}] = \mathbf{B}_L [\mathbf{S}_L]^{-1} \quad (18)$$

Solving the set of linear equations in (17) yields the tuned regulator coefficients $[\mathbf{C}_{\bar{P}}]$. $[\mathbf{S}_L]$ is rarely square therefore a pseudoinverse is used to find the least square solution of the

set of linear equations. Obtaining the least squares solution of (17) may also be expressed as finding the best fit of controller parameters such that the least squares error between the desired loop gain and the loop gain resulting from the controller design is minimized. The Moore-Penrose pseudoinverse algorithm is a suitable algorithm for this application and is described in (Matlab Reference Guide 1992). A second consideration is that $[\mathbf{S}_L]$ and $[\mathbf{B}_L]$ are complex matrices. To avoid the necessity of writing code for finding a pseudoinverse for a matrix with complex elements, (Gao Tabachnik, and Savescu) provides a method whereby the coefficients can be found by dividing $[\mathbf{S}_L]$ and $[\mathbf{B}_L]$ into real and imaginary parts as shown in (19). The pseudoinverse may now be obtained from a matrix with all real numbers, making the coding considerably simpler.

$$[a_0, \dots, a_n, -b_0, \dots, -b_{m-1}] = [\text{Re}(\mathbf{B}_L), \text{Im}(\mathbf{B}_L)][\text{Re}(\mathbf{S}_L), \text{Im}(\mathbf{S}_L)]^{-1} \quad (19)$$

Equation (19) forms the heart of the self-tuning algorithm.

The algorithm may be summarized as follows:

1. Provide a vector of desired loop gain frequency response data ($L(j\omega_l)$) at l frequencies.
2. From a set of measured input output data, taken from the plant on-line, compute the plant frequency response vector ($P(j\omega_l)$), using an FFT algorithm, at the same frequencies as $L(j\omega_l)$.
3. Substitute $L(j\omega_l)$ and $P(j\omega_l)$ into (13) and obtain the $[\mathbf{S}_L]$ and $[\mathbf{B}_L]$ matrices using (14).
4. Substitute the $[\mathbf{S}_L]$ and $[\mathbf{B}_L]$ matrices into (19) and solve the linear system of equations using a suitable pseudoinverse algorithm. The solution yields the controller parameters.
5. If the implementation is in a discrete environment, calculate the z- domain controller coefficients using a suitable transformation technique and update the controller.

A Short Discussion Regarding Controller Structure Selection

The choice of the controller structure is application dependent. Care must be taken to select a controller with an order high enough to satisfy the given regulation requirements. As stated previously, the proposed STR algorithm achieves the best possible fit between the desired loop gain and the actual loop gain. This may result in approximate pole zero cancellation. Approximate pole zero cancellation has been used for many years in industrial control systems. For example the motor electrical time constant is canceled with the zero of a PI controller in motor current loops, many other examples could be given. This approach is entirely acceptable when the values of the plant poles and zeroes do not change significantly (as is the case for a given product in web transport systems). If the plant poles and zeroes do change, an STR algorithm will be employed to adjust the controller parameters to compensate for the changes.

When dealing with pole zero cancellation of second order pole pairs the effectiveness of the cancellation is very much dependent on the damping of the pole pair. If the pole pair is extremely underdamped (i.e. $\zeta < 0.1$) the zeroes must be placed very close to

the poles to effectively cancel them. The transient step response of the system will show a discernible ringing if the poles are not canceled adequately. However web transport systems rarely, if ever, encounter open loop underdamped plant pole pairs with damping factors less than 0.3 and the use of 2nd order approximate cancellation techniques can be considered acceptable. In addition the bandwidth of the closed tension loop is always chosen to be less than the frequency of any underdamped pole pairs, resulting in attenuation of the resonance. Simulation has shown that in the presence of typical web transport system open loop underdamped pole pairs the proposed STR algorithm generates regulator designs that are more than adequate for acceptable tension regulation.

For this application a fourth order controller structure, as shown in Figure 6, provided adequate regulator designs. This choice was made with the a-priori knowledge that there are three dominant poles in the plant transfer function, as will be shown in the following section, and that making the regulator an order higher than the plant order provides on-line regulator tunings that meet low frequency gain specifications. In the following section it will be shown that if a PI regulator is used the system modeled in this paper is type 2 at stall, a condition where tension is applied to the web but the line speed is zero. This is due to an inherent integration in the plant, it will not however be type 2 when the web material is moving through the tension zone.

Of particular interest is the choice of domain to be used to derive the controller parameters. The Laplace domain applies to continuous systems but the real-time implementation will typically be on a computer which is discrete and any analysis relative to the derivation of the discrete controller parameters should be performed in the z-domain. However, given that the sample times of modern DCS control systems are of the order of 1–2 [msec] (which can be approximated as a pole at $2/T_s$ [rad/sec] or 1000–2000 [rad/sec]) and problematic natural frequencies encountered in web transport systems are less than 100 [rad/sec], the effect of sampling can be ignored and the controller parameters obtained in the Laplace domain. To find the final regulator coefficients in the z-domain a suitable transformation of the s-domain controller may be used.

3. Modeling of a Web Transport System

A physical representation of the plant shown in Figures 5 and 6 is shown in Figure 3. The motor and gear-box have been omitted. The shaft torque of the motor drives the gear-box which in turn drives Bridle 2, producing V_2 . Bridle 1 is regulated by a speed regulator only and is regulated to rotate at V_1 , the desired operational line speed. Assuming that,

1. Unstretched web is introduced into the tension zone.
2. Bridle No. 1 is an ideal speed regulator.

(Carter 1965; Parant, Coeffler and Lung 1992) show that for small signal analysis, change in tension in the tension zone (the tension zone is the length of web between the speed regulated bridle and the tension regulated bridle) can be modeled in the Laplace domain by (20) a commonly used tension model in web analysis.

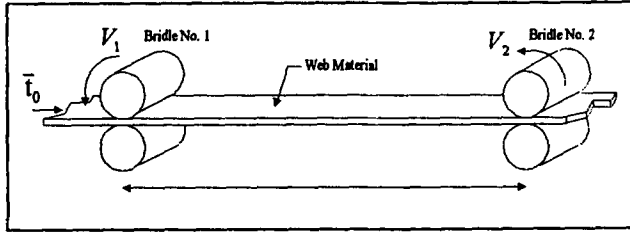


Figure 3. The physical plant.

$$\Delta T = \frac{E \cdot A}{V_1} \cdot \left(\frac{(V_2 - V_1)}{1 + \frac{L}{V_1} s} \right)_{t_0=0} \quad (20)$$

ΔT = The change in web tension in the tension zone [N]

E = Web modulus of elasticity [N/cm²]

A = Web cross sectional area [cm²]

L = Web length [m]

V_1 = Velocity of feed roll [m/sec]

V_2 = Velocity of exit roll [m/sec]

t_0 = tension in the web entering the tension zone.

Modeling the motor inertia and driven roll inertia (reflected through the gear-box to the motor shaft) as a lumped inertia J , a block diagram of the plant (Figure 4) can be constructed using (20). K_1 is a constant that converts motor rotational speed to linear line speed. K_2 is a constant that converts web tension to a torque reflected through the gear box and felt at the motor shaft. τ [N m] represents the torque produced by the motor. B_v is the coefficient of viscous friction. B_w is the web-damping modulus.

It can be shown that the transfer function from the output of the tension regulator ω_{TEN} to the tension feedback T_{FDBK} in Figure 6 is:

$$\frac{T_{FDBK}}{\omega_{TEN}} = \frac{\frac{K \cdot L}{V_1 \cdot K_2} (s/\omega_s + 1)}{\frac{J \cdot L}{K_s \cdot V_1 \cdot \omega_s} s^3 + \frac{J \cdot V_1 + K_s \cdot L}{K_s \cdot V_1 \cdot \omega_s} s^2 + \frac{K \cdot L + K_s (L \cdot \omega_s + V_1)}{K_s \cdot V_1 \cdot \omega_s} s + 1} \quad (21)$$

where:

J : Lumped Bridle/Motor inertia [Kg m²]

K_s : The inner speed loop PI proportional gain term

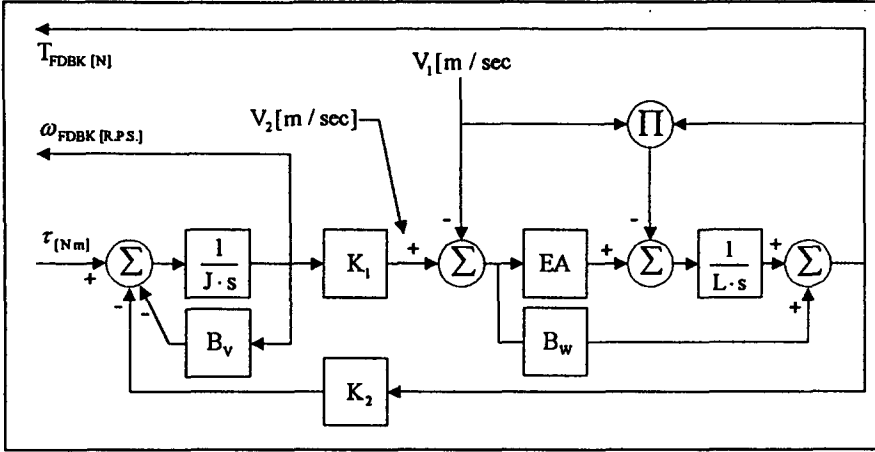


Figure 4. Plant block diagram (Including web-damping and viscous friction).

ω_s : The inner speed loop PI lead frequency [rad/sec]

D: The driven roll diameter [m]

GR: The gear ratio

$$K_1 = \frac{\pi \cdot D}{2 \cdot GR}$$

$$K_2 = \frac{D}{GR \cdot 2}$$

$$K = K_1 \cdot K_2 \cdot \frac{E \cdot A}{L} \cdot \frac{1}{\pi}$$

Note that (21) does not include web-damping (B_w) or viscous friction (B_v). If we include the transducer filter (see Figures 5 & 6), (21) becomes:

$$TF_{OL} = (21) \cdot \frac{1}{\frac{s^2}{\omega_f^2} + \frac{2s}{\omega_f} + 1} \quad (21.a)$$

Let: $K_v = \frac{V_1}{L}$; $\omega_{CO} = \frac{K_s}{J}$; $R^2 = \frac{K}{J}$

K_v : The inverse of the line velocity time constant

ω_{CO} : The closed speed loop bandwidth (w/o web)

R: The web/inertia natural frequency

It can be shown that the third order denominator in (21) can be approximately factored into the form:

$$\left(\frac{s}{\omega_1} + 1 \right) \left(\frac{s^2}{(\omega_2)^2} + \frac{2\zeta_r \cdot s}{\omega_2} + 1 \right) \quad (22)$$

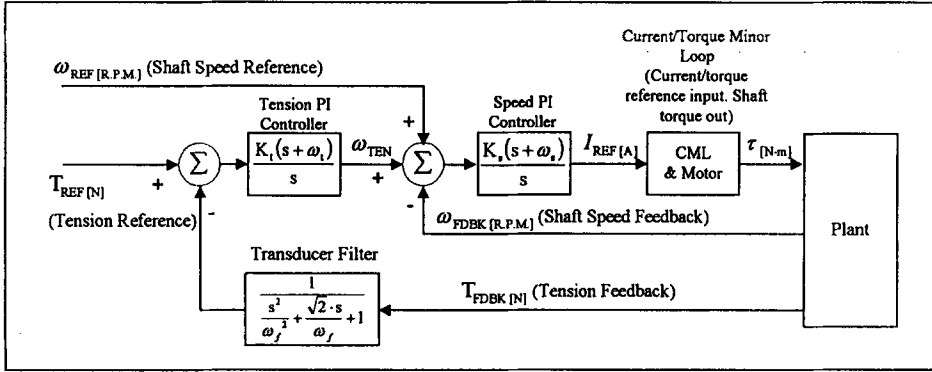


Figure 5. A typical non STR web tension regulation scheme as used in industry today.

where:

$$\left. \begin{aligned} \omega_2 &= \sqrt{R^2 + \omega_{CO} \cdot \omega_s} \\ \omega_1 &= \frac{K_v \cdot \omega_{CO} \cdot \omega_s}{\omega_2^2} \\ \zeta_T &= \frac{K_v \cdot R^2 + \omega_{CO} \cdot \omega_s^2}{2 \cdot \omega_2^3} \end{aligned} \right\} \quad (23)$$

The equations in (23) can be shown to be satisfactory approximations of the exact values providing $K_v \leq 10 \cdot \omega_{CO}$ which is reasonable for all practical web transport systems. Some interesting observations can be made from (23) namely;

1. The center frequency (ω_2) of the open loop 2nd order pole pair increases with a corresponding increase in the speed loop bandwidth.
2. Damping of the open loop 2nd order pole pair (ζ_T) increases with speed loop bandwidth.

In addition to the need for zero steady state error to a ramped speed reference, the above two observations provide additional impetus for the inclusion of a speed minor loop in any tension regulation scheme.

From the tension loop control system design perspective, the value of ζ_T in (23) is critical since the final design will depend on the installed system having a real or complex pole pair. The worst case system operating condition is at stall ($V_1 = 0$, $\omega_1 = 0$, $\zeta_T = \omega_{CO} / 2 \cdot \omega_2$) and it is in this condition that a web tension loop is typically tuned. Unfortunately in real applications the value of ζ_T is affected by non-modeled parameters such as web-damping, mechanical friction, roll slippage, and in the case of films and paper, temperature and humidity. It can also be shown that the bandwidth (or stiffness) of upstream speed regulators (Bridle No. 1 in this case) affects ζ_T . To summarize, the inability to analytically identify ζ_T in (23) is the main contributing factor to the need for heuristically tuning web tension regulators in the field.

It is also clear from (23) that the frequency ω_2 varies as a function of web parameters E, A, L and the lumped inertia J. In general it is obvious that a system with no STR mechanism

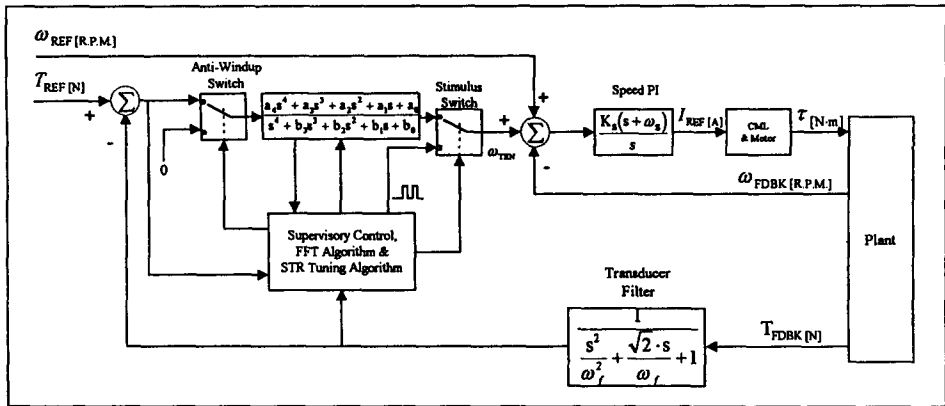


Figure 6. Proposed STR regulation scheme (As implemented in simulation).

needs to be tuned for stable operation with the worst case operating parameter set (quite often in and of itself hard to identify) and that for other operating parameter sets the system is running in a stable but less than desirable condition.

4. Self Tuning Web Tension Scheme & Simulation

4.1. Self Tuning Web Tension Regulator

To clarify the difference between the traditional web tension regulation approach and the proposed approach detailed block diagrams of the two schemes are presented below (Figures 5 & 6).

Figure 5 represents a typical tension regulation scheme as used in industry today. ω_{TEN} is the outer tension loop vernier contribution to line speed reference. Small changes in speed result in corresponding changes in web tension, this is the basic idea behind the use of an inner speed loop around which a tension regulator is strapped. In addition it can be shown that the use of a PI regulator for speed regulation results in a type 2 inner speed loop, a necessity for web transport systems where there must be zero steady state error to a ramped speed reference (for acceleration and deceleration of the process line). Both PI regulators are composed of a proportional gain term K_p and a zero frequency ω_{Lead} , the zero frequency is equal to the ratio of the integral and proportional gains K_i/K_p . This representation of a PI regulator is common in industrial control systems. A transducer filter is typically employed to filter unwanted high frequency noise. For this application the corner frequency of the transducer filter was set at 20 [rad/sec], a typical value.

Figure 6 is a representation of the proposed STR algorithm as implemented in the simulation described in this paper. The difference between this scheme and the scheme in Figure 5 is the replacement of the Tension Loop PI regulator with an STR regulator, an anti-windup switch, a stimulus switch and an autonomous supervisory controller. The function of the

anti-windup switch and the stimulus switch is explained in the logic sequence at the end of this section. The use of a PI controller for the inner speed loop in the STR regulation scheme is desirable due to the need to ramp up to an operational line speed with zero error to a speed ramp, as described previously, as well as the need to jog the bridle for threading new web material into the process line.

From Figure 1 it can be seen that a supervisory controller is implemented to provide the sequencing logic required to execute the STR algorithm. The logical sequence used in the simulation performed in this paper is as follows:

1. Set the anti-windup switch. (set the input of $C(s)$ to zero effectively disabling $C(s)$).
2. Provide a step stimulus to the plant input.
3. Collect a set of input/output plant data and perform an FFT to obtain $P(j\omega_i)$
4. Initiate the STR loop shaping algorithm, obtain a set of controller parameters, and update the controller.
5. Continue stimulus and monitor the feedback and tension reference. Turn off the stimulus when the feedback matches the reference (error = 0). Release the anti-windup switch and re-enable the re-tuned controller.

In other real-time applications this algorithm need not be adhered to. The more traditional STR approach may be implemented. That is, a continuous FFT may be generated from normal plant operational stimulus/response data and the controller updated periodically without the need for the anti-windup switch and an autonomous stimulus. However this is only feasible if the harmonic content of the normal operational stimulus is adequate enough to obtain a meaningful FFT. In applied tension regulation schemes the tension reference rarely changes and the feedback is always constant. This results in the need to provide an autonomous stimulus, as described above, when self-tuning is desired.

4.2. Simulation Set-Up

A simulation was performed using SIMULINK and MATLAB and computation times for algorithm execution logged. The purpose of this experiment was to determine the feasibility of implementing the algorithm in hardware. Based on the results of the described simulation a hardware implementation using the Reliance Electric "Automax" hardware/software DCS platform is currently being developed. To provide realistic unmodeled dynamics viscous friction, web-damping, tension feedback white noise and line speed white noise were included. The objective of the simulation was three-fold;

- 1) To demonstrate an algorithm execution times that are reasonable for the described, and similar, applications.
- 2) To demonstrate the ability of the self-tuning algorithm to satisfactorily re-tune the controller "on-line" for a wide range of product.

Table 1. Simulation set-up parameters.

| | | |
|---------------------------------------|---|--------------------------------------|
| (Oriented Polypropylene) E | = | 2×10^5 [N/cm ²] |
| A | = | 10 → 0.1 [cm ²] |
| L | = | 10 [m] |
| Roll diameter | = | 1 [m] |
| Gear ratio | = | π |
| Motor power | = | 10 [kW] |
| Gear-in speed | = | 1000 [rpm] |
| Inertia | = | 10 [Kg m ²] |
| Max. Line speed noise | = | 0.02% Max. line speed |
| Max. Transducer feedback noise | = | 0.1% Max. Tension feedback |
| Web damping modulus (B _w) | = | 100 [N sec/cm ²] |
| Viscous friction (B _v) | = | 0.03 [N m / rpm] |
| ω_f | = | 20 [rad/sec] |
| ω_{CML} | = | 200 [rad/sec] |
| ω_{CO} | = | 15 [rad/sec] |

- 3) To demonstrate the advantage of on-line self-tuning based on the measured frequency response of the plant as opposed to off-line tuning with the STR algorithm using the transfer function model (21.a) to obtain a theoretical plant frequency response.

The parameters used in the simulation are presented in Table No. 1. The plant frequency response is obtained with the use of an FFT implemented with a Hamming window with 50% overlap. For Figures 8 and 9 the following FFT set-up was used;

$$\begin{aligned}T_s &= 0.04 \text{ [sec]} \text{ (256 samples);} \\M &= 128 \text{ samples.} \\ \omega_i &= 0.2 \rightarrow 20 \text{ [rad/sec]} \text{ (200 points)}\end{aligned}$$

For Figures 9 and 10 the following FFT set-up was used;

$$\begin{aligned}T_s &= 0.01 \text{ [sec]} \text{ (1024 samples);} \\M &= 512 \text{ samples.} \\ \omega_i &= 0.5 \rightarrow 100 \text{ [rad/sec]} \text{ (200 points)}\end{aligned}$$

M specifies the size of the Hamming Window as described in (Ljung 1992) pp. 2-95–2-96.

A SIMULINK masked s-function performs the on-line FFT and executes the self-tuning algorithm to provide the new controller parameters. During the simulation the algorithm execution times were tabulated and are presented in Table 3 below. To accomplish self-tuning using the described algorithm the system is stimulated with a small speed reference step as described in section 2. Care was taken to ensure that web tension was reasonably bounded during this period. To prevent regulator windup during the plant stimulus process, the input to the tension regulator is forced to zero by setting the anti-windup switch.

For the first simulation set (Figures 7 through 12) the self-tuning system was initialized with a set of poorly tuned controller parameters. Two tension steps of 5 [sec] duration (from 0 to 10 [sec] in Figures 9 and 12) were followed by a 10 [sec] data collection and regulator tuning period. After which the re-tuned STR response to three tension steps of

Table 2. STR regulator parameters.

| coeff. | Figure 7 | | Figure 10 | |
|----------------|----------|--------|-----------|--------|
| | Before | After | Before | After |
| a ₀ | 1.9992 | 0.8720 | 0.2890 | 1.9992 |
| a ₁ | 1.6204 | 0.9376 | 0.4724 | 1.6204 |
| a ₂ | 0.0998 | 0.3554 | 0.3382 | 0.0998 |
| a ₃ | 0.0017 | 0.0438 | 0.0441 | 0.0017 |
| a ₄ | 0.0000 | 0.0015 | 0.0015 | 0.0000 |
| b ₀ | 2.2025 | 0.6615 | 0.4561 | 2.2025 |
| b ₁ | 3.3890 | 3.8308 | 3.1744 | 3.3890 |
| b ₂ | 3.2350 | 2.2528 | 2.3997 | 3.2350 |
| b ₃ | 0.1136 | 0.2929 | 0.3027 | 0.1136 |
| b ₄ | 0.0010 | 0.0010 | 0.0010 | 0.0010 |

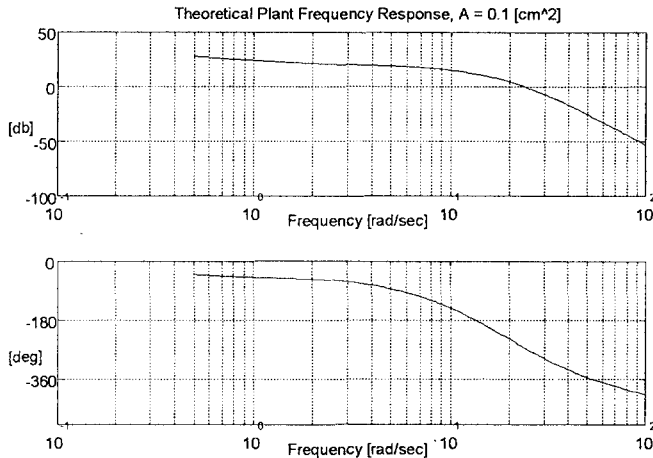


Figure 7. Theoretical freq. resp. ($A = 0.1 \text{ [cm}^2\text{]}$).

5 [sec] duration was obtained. Figures 7 and 10 represent the theoretical plant frequency responses obtained with the use of (21.a). The plant frequency responses obtained on-line with the FFT are shown in Figures 8 and 11. These plots are presented with probabilistic bounds of 1 standard deviation as described in (Ljung 1992) pp 1–21, 22. This presentation technique is similar to that described in (Astrom 1993) section 2. The regulator coefficients before and after self tuning for Figures 9 and 12 are presented in Table No. 2.

A second simulation (Figures 13 and 14) was performed to obtain a comparison between a regulator tuned off-line with the STR algorithm using transfer function (21.a) to obtain the plant frequency response (from 0 to 10 [sec]) and a regulator tuned on-line with the STR algorithm (from 20 to 35 [sec]). The on-line tuning process was the same as that used above.

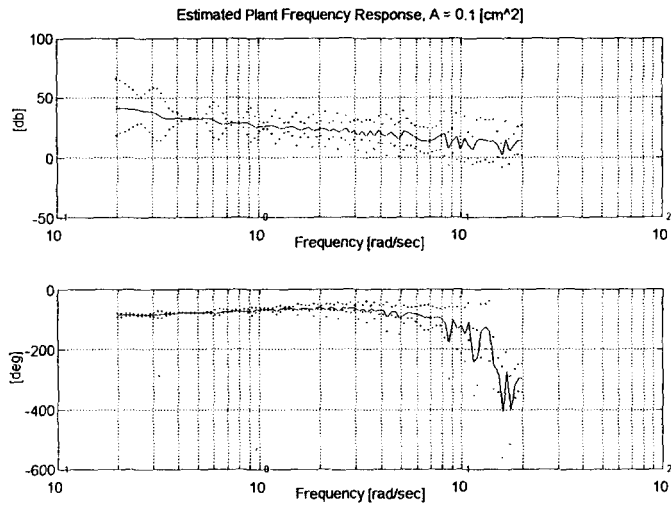


Figure 8. Estimated freq. resp. ($A = 0.1 \text{ [cm}^2\text{]}$).

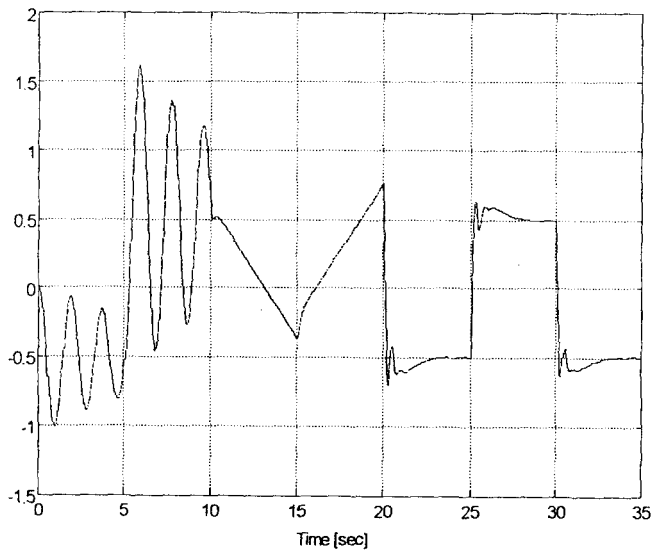


Figure 9. STR tuning ($A = 0.1 \text{ [cm}^2\text{]}$).

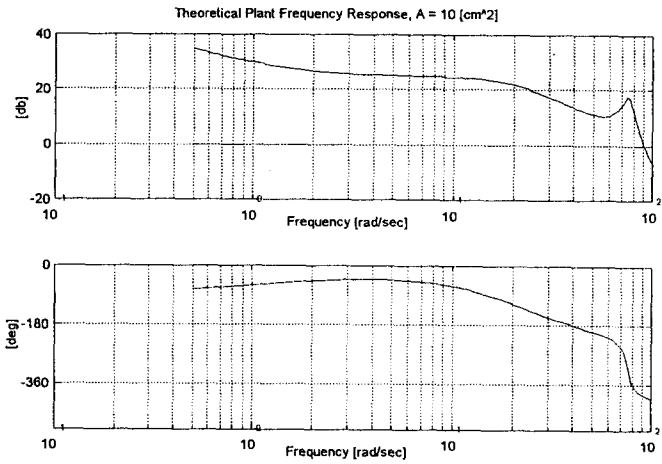


Figure 10. Theoretical freq. resp. ($A = 10 \text{ [cm}^2\text{]}$).

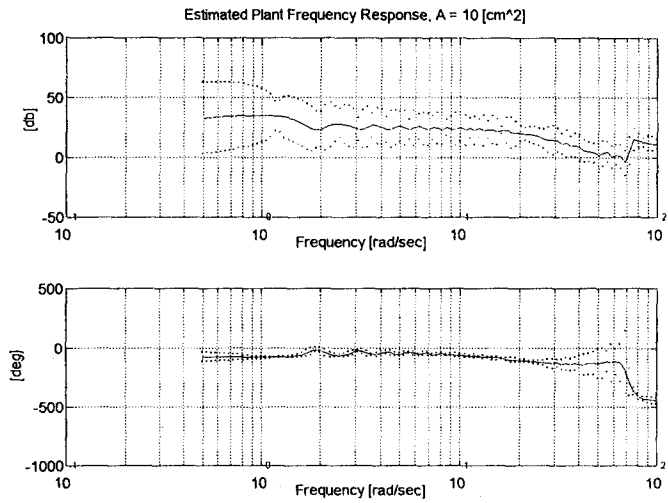


Figure 11. Estimated freq. resp. ($A = 10 \text{ [cm}^2\text{]}$).

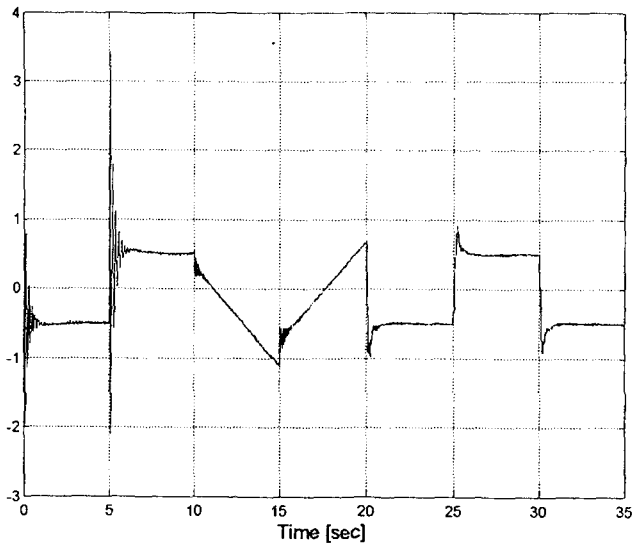


Figure 12. STR tuning ($A = 10 \text{ [cm}^2\text{]}$).

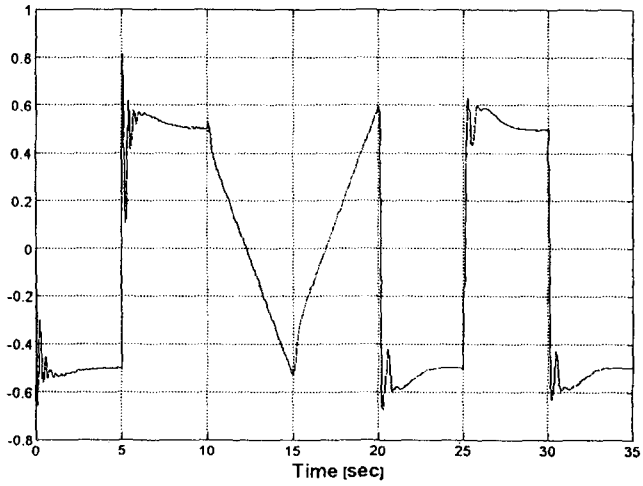


Figure 13. Theoretical v 's on-line tuning ($A = 0.1 \text{ [cm}^2\text{]}$).

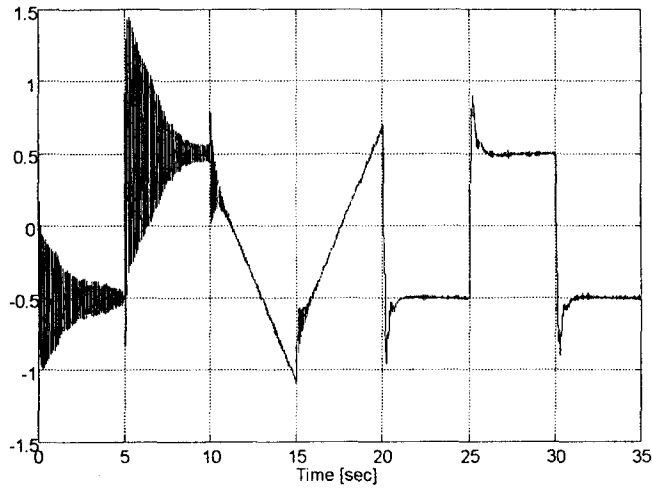


Figure 14. Theoretical v 's on-line tuning ($A = 10 \text{ [cm}^2\text{]}$).

Table 3. STR algorithm execution times.

| Length of l [points] | Controller order | Pseudoinverse execution time [sec] | Algorithm execution time [sec] |
|------------------------|------------------|------------------------------------|--------------------------------|
| 50 | 2 | 0.050 | 0.061 |
| 50 | 4 | 0.050 | 0.063 |
| 50 | 8 | 0.050 | 0.066 |
| 100 | 2 | 0.060 | 0.072 |
| 100 | 4 | 0.060 | 0.075 |
| 100 | 8 | 0.110 | 0.128 |

4.3. Simulation Observations

The execution times for the STR algorithm were obtained using a 486 66 [mHz] PC compatible computer running SIMULINK V1.2c. It was noted that the algorithm execution time was consumed mostly by the pseudoinverse algorithm. For the simulation discussed in this paper 50 frequency points and a fourth order controller were used.

An interesting observation from Table 3 is that the execution time for the pseudoinverse did not vary greatly for the range of matrix sizes that can be expected using the described algorithm.

The FFT execution times will vary greatly, depending on the choice of smoothing and windowing algorithms in addition to the number of points used in the FFT. Typically, however, in real-time applications where the STR algorithm is to be used in an adaptive regulation scheme the time-varying parameters in the controlled plant will be an order of magnitude or more slower than the execution time for the FFT. In a given application the

FFT will most probably be obtained with an off-the-shelf hardware plug-in and will run independent of the STR algorithm, providing $P(j\omega_i)$ on an as-needed basis.

The following set of figures summarize the results of the simulation. Figures 9 and 12 demonstrate the STR algorithms ability to tune on-line in the presence of unmodeled dynamics for two extreme product cross-sectional areas. The higher than desired overshoot can be attributed to inaccuracies in the on-line measured FFT plant frequency response. It was observed that relatively accurate plant frequency response data, as shown in Figures 8 and 11, is critical to the success of the application of the described self-tuning algorithm. Factors impacting the quality of the estimated plant frequency response are: 1) Sample Time (high frequency response). 2) The sample length (low frequency response). 3) The choice of FFT windowing algorithm (transient disturbance and noise rejection). 4) The choice of smoothing algorithm (noise rejection). 5) Harmonic content of the input stimulus (overall frequency response). These topics have been covered in (Goderdhansingh, Wang and Cluett 1992; Cunningham 1992; Whitfield 1986; and Jenkins and Watts 1968). Adequate performance of the STR algorithm on-line will depend on making appropriate choices in 1) through 5) based on the given application and the desired loop gain.

Simulation results also indicate that the choice of frequencies used in the generation of $L(j\omega)$ and $P(j\omega)$ govern the quality of the final self-tuned regulator design. Care should be taken to provide frequencies low enough to generate a meaningful estimation of the plants low frequency response yet high enough to provide for adequate compensation of high frequency underdamped modes.

Figures 13 and 14 show clearly that off-line tuning using the approximate linear plant transfer function (21.a) to derive the estimated plant frequency response resulted in unsatisfactory loop response. This can be attributed to the unmodeled dynamics present in the simulated system but not accounted for in (21). However by generating the frequency response of the plant on-line (including the unmodeled dynamics) the STR algorithm was able to provide a regulator tuning which adequately compensated for the included unmodeled dynamics.

5. Implementation Issues/Conclusions

The object of good tension regulation in a web transport system is to produce good product, any excessive deviation in tension typically results in product that must be scrapped. Since the stimulus required to obtain a meaningful plant frequency response with an FFT will for all practical purposes result in material that must be scrapped it follows that the self-tuning be performed seldom and generate a minimum of waste material (Quite often a process's profit margin is tightly linked to the amount of scrap material produced in a given time period). With this in mind a real-time implementation of the described STR in an industrial environment requires some thought and planning. If it is possible to excite the web in a stall condition, waste can be minimized. Applications that would meet this criteria for example, include rolling mills and tension levelers. However, it would not be practical in the furnace sections of continuous annealing lines. Mill management systems provide a medium through which controller configurations for particular set-ups may be stored. Using such a management system would alleviate the need for initiating self-tuning for

previously run product set-ups. In addition the self tuning algorithm need not be initiated for every product set-up, but rather for those where there is a significant change in product parameters or a predetermined error measure is exceeded.

The step stimulus used in the simulation may not be practical for most systems. For example, materials such as steel do not stretch much and a very little change in speed results in a significant change in tension. For these systems significantly smaller random speed steps would be required.

In applications where a real-time system requires the implementation of an adaptive scheme that continuously up-dates the regulator parameters, the update time will depend less on the algorithm execution time and more on the FFT execution time. In these applications care must be taken to ensure that FFT executes at least on order of magnitude faster than the fastest time-varying time constant of the controlled variable. In addition, as a rule of thumb, the FFT windowing overlap should be set at 20%–50% to eliminate transient distortion of the FFT. Stimulus schemes may be used in the same fashion as those currently implemented in MRAS based rolling mill gauge control regulation schemes. The main advantage of the described STR algorithm over the MRAS approach in current use is that the controller parameters are obtained "instantly" relative to the time taken in current MRAS schemes to converge to a solution set of controller parameters.

This paper did not investigate the use of frequency weighting in $L(j\omega)$ as described in (Anstaklis and Gao 1993) and (Gao and Anstaklis 1994). Use of this technique may prove useful in reducing the number of points required in the pseudoinverse.

Given that the time taken to find the least squares solution to the system of linear equations of a particular application has been identified, system computational resources may be allocated in a predictable manner. This feature is extremely desirable in large systems where computational resources are constrained and emphasis is on ever increasing sample rates.

In conclusion, the evident elegance and straightforward implementation of the STR algorithm presented in this paper shows great promise for its application in contemporary industrial control engineering practice. It is in no way limited to the tension problem presented, but may be applied to any of a number of SISO control applications.

References

- Anstaklis, P. J., and Gao, Z. 1993. Polynomial and rational matrix interpolation: Theory and control applications. *International Journal of Control* 58(2):349–404.
- Astrom, K. J. 1993. Autonomous controllers. *Control Engineering Practice*, 1(2):227–320.
- Astrom, K. J., Hagglund, T., Hang, C. C., Ho, W. K. 1993. Automatic tuning and adaptation for PID controllers—a Survey. *Control Engineering Practice* 1(4):699–714.
- Astrom, K. J., and Wittenmark, B. 1989. *Adaptive Control*. Allison Wesley.
- Carter, W. C. 1965. Reducing transient strains in elastic-strip processes. *Control Engineering*:84–87.
- Cunningham, E. 1992. *Digital Filtering*. Ch. 6,8 and 9, Houghton Mifflin.
- Jenkins, G. M., and Watts, D. G. 1968. *Spectral analysis and its applications*. Holden-Day.
- Gao, Z. 1993. An algorithmic approach to loop shaping with applications to self-tuning control systems. *Proceedings of the 8th. IEEE Symposium on Intelligent Control*.
- Gao, Z., and Anstaklis, P. J. 1994. New methods for control system design using matrix interpolation. *Proceedings of the 33rd IEEE Conference on Decision and Control*, 2506–2511.

- Gao, Z., Tabachnik, B., and Savescu, R. Transfer function matrix identification from input-output measurements. to appear in the *Journal of the Franklin Institute*.
- Goderdhansingh, Wang, L., and Cluett, W. R. 1992. Robust control system design using direct frequency response. *Proceedings of the American Control Conference*, 3026–3030.
- Landau, I. D. 1993. Evolution of adaptive control. *Journal of Dynamic Systems, Measurement and Control, Transactions of the ASME* 115(2B):381–391.
- Lee, C. C. 1990. Fuzzy logic in control systems-Part I and II. *IEEE Transactions on Systems, Man and Cybernetics* 20(2):404–435.
- Ljung, L. 1992. MATLAB System Identification Toolbox. The Mathworks.
- Matlab Reference Guide*. 1992. The Mathworks, 369–370.
- Parant, F., Coeffler, C., and Lung, C. 1992. Modeling of web tension in a continuous annealing line. *Iron and Steel Engineer*:46–49.
- Shin, K. G., and Cui, X. 1991. Design of a knowledge-based controller for intelligent control systems. *IEEE Transactions on Systems, Man, and Cybernetics* 21(2).
- Shirley, R. S. 1987. Some lessons learned using expert systems for process control, *IEEE Control Systems Magazine*:11–15.
- Tzes, A. P., and Yurkovich, S. 1992. On frequency domain loop shaping for self-tuning control. *Proceedings of the American Control Conference*, 61–65.
- Whitfield, A. H. 1986. Transfer function synthesis using frequency response data. *International Journal of Control* 43(5):1413–1426.