Transfer Function Identification by Using Self-Oscillations

N. Mišković, Z. Vukić and M. Barišić

University of Zagreb, Faculty of Electrical Engineering and Computing
Department of Control and Computer Engineering,
Laboratory for Underwater Systems and Technologies
Zagreb, Croatia

Abstract—**The work presented in this paper deals with the process of transfer function identification by using self-oscillation method (autotuning identification method). The algorithm is given in a general matrix form and some modifications are introduced. The modifications of the algorithm include augmentation of the initial algorithm for Type *k* systems, systems with delays and discrete-time systems. The paper also includes simulation examples which describe the introduced modifications. Apart from being rather simple, this method is applicable to real systems. Its greatest advantage is quick identification of a transfer function (depends on the system).**

Keywords: identification, self-oscillations, autotuning

I. INTRODUCTION

Identification of process' dynamics is usually the first step towards designing advanced controllers. A great number of methods have been developed for tuning controller parameters without knowing the process dynamics, [1], [2], etc. These methods are based upon numerous experiments conducted on a specific type of process, giving recommendations for specific controller structure and its parameters. The downside to these methods is that they were developed for a small number of process types and specific desired closed loop performance. Nevertheless, these same methods can be used for process' identification, leaving it up to the control engineer to design a controller according to desired performance.

Identification of process parameters in open-loop is often tedious and time consuming. If process' parameters change in time (due to time-variant payload, disturbances, environment), classical identification methods are simply not convenient. It was more then 20 years ago when Åström and Hägglund derived a so called ATV (autotuning variation) method used for system identification, [3], that is simple and appropriate for in situ identification. The method used a relay-feedback to bring the system to self-oscillations. Then Luyben used this method in chemical industry to determine a transfer function of extremely nonlinear systems (distillation columns), [4]. Since then, relay-feedback systems proved to be a great tool for controller tuning in processes and for process identification, [5], [6], especially in pharmaceutical industry. An application of this type of identification to underwater vehicles can be found in [7].

Many modifications of the original autotuning method have been made in order to apply it to different types of processes. The main objection to this method is that only one frequency is taken into consideration (the frequency of self-oscillations). That is why some authors used an additional time delay in series with a relay to insert a phase delay which will 'rotate' the Nyquist curve and therefore obtain different oscillation frequencies, [8], [9]. Another approach for determining process' characteristics at different frequencies was to place an integrator in a cascade to a relay element, [10]. As this area of research expanded, more complex methods have been developed using a similar concept. Great improvement was achieved in using transient response of the relay feedback system response to tune controllers and identify systems, [11], [12]. Wang et al. use a switching technique between a relay and a relay with an integrator to obtain sufficient information about the process, [13]. Another modification of a relay experiment is using a biased relay to identify system's parameters, [14], [15]. This method is very accurate when used to determine the open loop gain of the process.

All of these methods were used primarily for static processes (Type 0). However, a number of processes are not open-loop stable, and there are not many references regarding this problem. First publications that used this method for identification of Type 1 systems included a derivator in series with a relay element to compensate for the influence of the integrator. This method is more of a theoretical value than practical one, due to non-causality of an ideal derivator. Another, implementable, approach that uses a liner feedback which ensures the static behavior of the process was presented by Kwak et al. in [16]. We did not manage to find references that address the use of this method for Type *k* processes.

In this paper, we present a general autotuning identification algorithm that is based on inducing self-oscillations upon the system by introducing a nonlinear element in the closed-loop system. Section II describes the autotuning concept and gives a general matrix-based algorithm for identification. Section III presents the algorithm modification for Type *k* systems and systems with delays. Special attention is given to practical implementation of the method, which includes algorithm modification for discrete-time systems. It also describes filtering of self-oscillations in order to obtain more precise results. In section IV some simulation examples are presented in order to show the usefulness of the proposed method and improvements of the identification procedure when the modifications are introduced. The paper is concluded with Section V.

II. ALGORITHM FOR AUTOTUNING IDENTIFICATION

A. The Autotuning Concept

The autotuning identification is based upon introduction of a nonlinear element in the closed-loop, as shown in Figure 1.

Usually, the nonlinear element that is used is a relay with hysteresis. This is not the only nonlinear element that can induce self-oscillations, but is the simplest to implement and therefore most common in practice. The algorithm for autotuning identification will be developed for a general nonlinear element, whilst the simulation examples will be shown with a relay with hysteresis.

Autotuning identification method boils down to determining a set of process' parameters by using the data obtained from the induced self-oscillations. The grapho-analytical procedure can be described as: find the intersection points between frequency characteristic of the LTI process and an inverse negative describing function of the non-linear element. This grapho-analytical method is known as the Goldfarb method, [17], and is given with (1) where $G_N(X_m)$ is the describing function , X_m is the magnitude of oscillations at the input of the nonlinear element and $G_P(j\omega)$ is the process frequency characteristic.

$$G_P(j\omega) = -\frac{1}{G_N(X_m)} = -\frac{1}{P_N(X_m) + jQ_N(X_m)} \quad (1)$$

For the case of a relay with hysteresis and under the assumption that the self-oscillations are symmetric (for more details see [17] and [18]) the describing function is shown with (2) where x_a is half the width of the hysteresis, and C is the relay output.

$$P(X_m) = \frac{4C}{\pi X_m}\sqrt{1 - \left(\frac{x_a}{X_m}\right)^2}$$

$$Q(X_m) = -\frac{4C}{\pi X_m}x_a \quad (2)$$

Once the self-oscillations are established in a closed loop system, we can use the magnitude and frequency of the obtained self-oscillations at the input of the nonlinear element and determine one point in the Nyquist plane which is also one point on the Nyquist curve of the process, [17]. If the process consists of maximum two unknown parameters and its transfer function is known, a Nyquist curve of the process can be uniquely interpolated through that specific point. If the process is more complex, e.g. consists of 4 unknown parameters, it takes two points in the Nyquist plane and consequently two autotuning experiments with different nonlinear element parameters are required (see Figure 2. where P_{N1}', P_{N2}' are real and Q_{N1}', Q_{N2}' are imaginary parts of (1)).
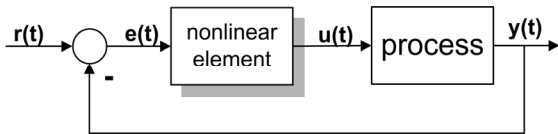

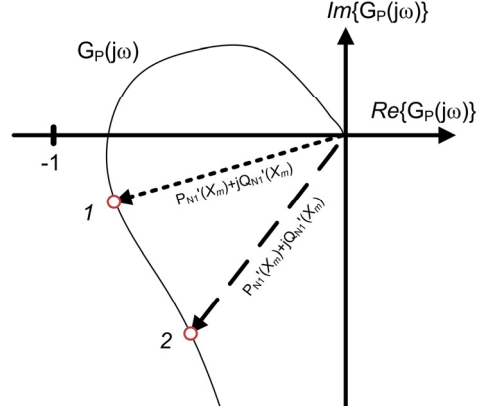Figure 1. Closed-loop scheme used for autotuning identification.

Figure 2.   Points of intersection in the Nyquist plane.

B. General Algorithm Matrix Formulation for Static Processes (Type 0 Processes)

A linear time invariant process can be described by a transfer function (3) where n is the degree of the denominator (number of non-zero poles), m degree of the nominator (number of finite zeros) and $n \geq m$.

$$G_P(s) = \frac{\sum_{i=0}^{m} b_i s^i}{\sum_{i=0}^{n} a_i s^i} \quad (3)$$

Let us suppose that the closed loop system is as in Figure 1. Using the Goldfarb method (1) we can obtain a general equation in the frequency domain that gives relation between oscillation parameters (magnitude X_m and frequency ω_u) and process' parameters. The process transfer function transferred to frequency domain has the following form:

$$G_P(j\omega) = \frac{\sum_{i=0}^{m} b_i(j\omega)^i}{\sum_{i=0}^{n} a_i(j\omega)^i} =$$

$$= \frac{\left[\sum_{i=0}^{\lfloor n/4 \rfloor} b_{4i}\omega^{4i} - \sum_{i=0}^{\lfloor n/4 \rfloor} b_{4i+2}\omega^{4i+2}\right] + j\left[\sum_{i=0}^{\lfloor n/4 \rfloor} b_{4i+1}\omega^{4i+1} - \sum_{i=0}^{\lfloor n/4 \rfloor} b_{4i+3}\omega^{4i+3}\right]}{\left[\sum_{i=0}^{\lfloor n/4 \rfloor} a_{4i}\omega^{4i} - \sum_{i=0}^{\lfloor n/4 \rfloor} a_{4i+2}\omega^{4i+2}\right] + j\left[\sum_{i=0}^{\lfloor n/4 \rfloor} a_{4i+1}\omega^{4i+1} - \sum_{i=0}^{\lfloor n/4 \rfloor} a_{4i+3}\omega^{4i+3}\right]} \quad (4)$$

After combining (1) and (4), and equating imaginary and real parts, the following equations are obtained.

$$\left[\sum_{i=0}^{\lfloor n/4 \rfloor} a_{4i}\omega^{4i} - \sum_{i=0}^{\lfloor n/4 \rfloor} a_{4i+2}\omega^{4i+2}\right] = -P\left[\sum_{i=0}^{\lfloor n/4 \rfloor} b_{4i}\omega^{4i} - \sum_{i=0}^{\lfloor n/4 \rfloor} b_{4i+2}\omega^{4i+2}\right] + Q\left[\sum_{i=0}^{\lfloor n/4 \rfloor} b_{4i+1}\omega^{4i+1} - \sum_{i=0}^{\lfloor n/4 \rfloor} b_{4i+3}\omega^{4i+3}\right]$$

$$\left[\sum_{i=0}^{\lfloor n/4 \rfloor} a_{4i+1}\omega^{4i+1} - \sum_{i=0}^{\lfloor n/4 \rfloor} a_{4i+3}\omega^{4i+3}\right] = -Q\left[\sum_{i=0}^{\lfloor n/4 \rfloor} b_{4i}\omega^{4i} - \sum_{i=0}^{\lfloor n/4 \rfloor} b_{4i+2}\omega^{4i+2}\right] - P\left[\sum_{i=0}^{\lfloor n/4 \rfloor} b_{4i+1}\omega^{4i+1} - \sum_{i=0}^{\lfloor n/4 \rfloor} b_{4i+3}\omega^{4i+3}\right]$$

From this we can conclude that one autotuning experiments gives two equations that can be used for identification.

If we run enough autotuning experiments so that all unknown parameters can be determined, the latter two equations can be expressed in a matrix form. In order to obtain a unique solution of the matrix equation, we fix the value of one parameter: $a_0 = 1$. The number of the experiments that need to be run in order to identify all the parameters of process (3) with $a_0 = 1$ is

$$\varepsilon = \left\lceil \frac{n + m + 1}{2} \right\rceil \quad (5)$$

Let's define three vectors of measurements $\boldsymbol{\omega}=\begin{bmatrix}\omega_1 & \cdots & \omega_\varepsilon\end{bmatrix}^T$; $\mathbf{P}=\begin{bmatrix}P_1 & \cdots & P_\varepsilon\end{bmatrix}^T$ and $\mathbf{Q}=\begin{bmatrix}Q_1 & \cdots & Q_\varepsilon\end{bmatrix}^T$, where elements P_i and Q_i are functions of the experimentally obtained magnitude of self-oscillations and nonlinear elements parameters, and ω_i frequency of the self-oscillations obtained in the ith experiment.

The vector of unknown parameters is defined as

$$\boldsymbol{\Theta}=\begin{bmatrix}\boldsymbol{\Theta}_a & \boldsymbol{\Theta}_b\end{bmatrix}^T=\begin{bmatrix}a_1 & \cdots & a_n | b_0 & \cdots & b_m\end{bmatrix}^T.$$

From the above mentioned we can write the following equation:

$$\underbrace{\begin{bmatrix}\boldsymbol{\Omega}_a | \boldsymbol{\Omega}_b\end{bmatrix}}_{\boldsymbol{\Omega}}\boldsymbol{\Theta}=\underbrace{\begin{bmatrix}-\mathbf{I}\\0\end{bmatrix}}_{\mathbf{Y}} \quad (6)$$

where

$$\boldsymbol{\Omega}_a=\begin{bmatrix}\mathbf{0}_\varepsilon & -\mathbf{I}_\varepsilon & \mathbf{0}_\varepsilon & \mathbf{I}_\varepsilon\\ \mathbf{I}_\varepsilon & \mathbf{0}_\varepsilon & -\mathbf{I}_\varepsilon & \mathbf{0}_\varepsilon\end{bmatrix}\begin{bmatrix}\boldsymbol{\omega}^{.1} & 0 & 0 & 0 & \cdots & \neg(n-1)\%4\cdot\boldsymbol{\omega}^{.n}\\ 0 & \boldsymbol{\omega}^{.2} & 0 & 0 & \cdots & \neg(n-2)\%4\cdot\boldsymbol{\omega}^{.n}\\ 0 & 0 & \boldsymbol{\omega}^{.3} & 0 & \cdots & \neg(n-3)\%4\cdot\boldsymbol{\omega}^{.n}\\ 0 & 0 & 0 & \boldsymbol{\omega}^{.4} & \cdots & \neg(n-4)\%4\cdot\boldsymbol{\omega}^{.n}\end{bmatrix}$$

$$\boldsymbol{\Omega}_b=\begin{bmatrix}\mathbf{P}^T & \mathbf{Q}^T\\ \mathbf{Q}^T & -\mathbf{P}^T\end{bmatrix}\begin{bmatrix}\mathbf{I}_\varepsilon & \mathbf{0}_\varepsilon & -\mathbf{I}_\varepsilon & \mathbf{0}_\varepsilon\\ \mathbf{0}_\varepsilon & -\mathbf{I}_\varepsilon & \mathbf{0}_\varepsilon & \mathbf{I}_\varepsilon\end{bmatrix}\begin{bmatrix}\boldsymbol{\omega}^{.0} & 0 & 0 & 0 & \cdots & \neg m\%4\cdot\boldsymbol{\omega}^{.m}\\ 0 & \boldsymbol{\omega}^{.1} & 0 & 0 & \cdots & \neg(m-1)\%4\cdot\boldsymbol{\omega}^{.m}\\ 0 & 0 & \boldsymbol{\omega}^{.2} & 0 & \cdots & \neg(m-2)\%4\cdot\boldsymbol{\omega}^{.m}\\ 0 & 0 & 0 & \boldsymbol{\omega}^{.3} & \cdots & \neg(m-3)\%4\cdot\boldsymbol{\omega}^{.m}\end{bmatrix}$$

and $\mathbf{I}_\varepsilon=\mathbf{I}_{\varepsilon\times\varepsilon}$, $\mathbf{0}_\varepsilon=\mathbf{0}_{\varepsilon\times\varepsilon}$, $0=\mathbf{0}_{\varepsilon\times 1}$, $\mathbf{I}=\mathbf{I}_{\varepsilon\times 1}$. The dot symbol $(.^k)$ denotes the element-wise exponent, % is the modulus operation and $\neg$ is the logical negation symbol. The parameter vector $\boldsymbol{\theta}$ can be found by using the formula $\boldsymbol{\Theta}=\boldsymbol{\Omega}^{-1}\mathbf{Y}$ only if there is an even number of unknown parameters. If there is an odd number of parameters, matrix $\boldsymbol{\Omega}$ will have one row more than there are parameters. In this case, the last row can simply be omitted, or the pseudo-inversion (7) can be used to determine the solution.

$$\boldsymbol{\Theta}=\left\{\boldsymbol{\Omega}^T\boldsymbol{\Omega}\right\}^{-1}\mathbf{Y} \quad (7)$$

Example:
The following example will describe the algorithm development for the second order process: $n=2$; $m=0$. According to (5), $\varepsilon=2$. Partial measurement matrices $\boldsymbol{\Omega}_a$ and $\boldsymbol{\Omega}_b$ are obtained as

$$\boldsymbol{\Omega}_a=\begin{bmatrix}\mathbf{0}_2 & -\mathbf{I}_2 & \mathbf{0}_2 & \mathbf{I}_2\\ \mathbf{I}_2 & \mathbf{0}_2 & -\mathbf{I}_2 & \mathbf{0}_2\end{bmatrix}\begin{bmatrix}\boldsymbol{\omega}^{.1} & 0\\ 0 & \boldsymbol{\omega}^{.2}\\ 0 & 0\\ 0 & 0\end{bmatrix}=\begin{bmatrix}0 & -\omega_1^2\\ 0 & -\omega_2^2\\ \omega_1 & 0\\ \omega_2 & 0\end{bmatrix}$$

$$\boldsymbol{\Omega}_b=\begin{bmatrix}\mathbf{P}^T\mathbf{I}_2 & -\mathbf{Q}^T\mathbf{I}_2 & -\mathbf{P}^T\mathbf{I}_2 & \mathbf{Q}^T\mathbf{I}_2\\ \mathbf{Q}^T\mathbf{I}_2 & \mathbf{P}^T\mathbf{I}_2 & -\mathbf{Q}^T\mathbf{I}_2 & -\mathbf{P}^T\mathbf{I}_2\end{bmatrix}\begin{bmatrix}\boldsymbol{\omega}^{.0} & 0\\ 0 & \boldsymbol{\omega}^{.1}\\ 0 & 0\\ 0 & 0\end{bmatrix}=$$

$$=\begin{bmatrix}\begin{bmatrix}P_1 & 0\\ 0 & P_2\end{bmatrix}\begin{bmatrix}\omega_1^0\\ \omega_2^0\end{bmatrix}\Big| -\begin{bmatrix}Q_1 & 0\\ 0 & Q_2\end{bmatrix}\begin{bmatrix}\omega_1^1\\ \omega_2^1\end{bmatrix}\\ \begin{bmatrix}Q_1 & 0\\ 0 & Q_2\end{bmatrix}\begin{bmatrix}\omega_1^0\\ \omega_2^0\end{bmatrix}\Big| \begin{bmatrix}P_1 & 0\\ 0 & P_2\end{bmatrix}\begin{bmatrix}\omega_1^1\\ \omega_2^1\end{bmatrix}\end{bmatrix}=\begin{bmatrix}P_1\omega_1^0 & -Q_1\omega_1^1\\ P_2\omega_2^0 & -Q_2\omega_2^1\\ Q_1\omega_1^0 & P_1\omega_1^1\\ Q_2\omega_2^0 & P_2\omega_2^1\end{bmatrix}$$

Full matrices in equation (6) are.

$$\boldsymbol{\Omega}=\begin{bmatrix}0 & -\omega_1^2 & P_1\omega_1^0 & -Q_1\omega_1^1\\ 0 & -\omega_2^2 & P_2\omega_2^0 & -Q_2\omega_2^1\\ \omega_1 & 0 & Q_1\omega_1^0 & P_1\omega_1^1\\ \omega_2 & 0 & Q_2\omega_2^0 & P_2\omega_2^1\end{bmatrix}\quad \boldsymbol{\Theta}=\begin{bmatrix}a_1 & a_2 & b_0\end{bmatrix}\quad \mathbf{Y}=\begin{bmatrix}-1\\ -1\\ 0\\ 0\end{bmatrix}$$

From here, equation (7) gives the process' parameters.

III. MODIFICATIONS OF THE GENERAL ALGORITHM

The algorithm described in the previous section was generalized for static processes of any order. The modifications presented in this section will enable the use of the general algorithm for astatic, discrete-time and processes with delay without changing its appropriate matrix formulation.

A. On Higher Harmonics and Signal Filtering

Higher harmonics in real systems are always present. The main assumption in the describing function method is that the input to the nonlinear element contains only one harmonic, i.e. that the process is a good low-pass filter, [17]. Type 1 processes are good low-pass filters due to high damping of the integrator at high frequencies; therefore higher harmonics are well-attenuated. Self-oscillations of systems with delays often contain higher harmonics, i.e. the oscillations are not monoharmonic. This implies that better results could be obtained if the obtained oscillations are filtered.

The procedure that should be followed when filtering out higher harmonics is well-known in the literature. One suggestion is as follows:

- Obtain the self-oscillations of frequency ω_u and magnitude X_m;
- Filter the acquired oscillations by using a filter of the form $G_f(s)=\left(\dfrac{\omega_u\sqrt{2}}{s+\omega_u}\right)^N$ where N is an arbitrary order of the filter. Note that the higher the order of the filter, the longer the set of self-oscillations will be needed to get to the filtered steady oscillations of magnitude X_m^*;
- use the general algorithm with ω_u and X_m^* parameters.

B. Modification for Type k systems

The algorithm described in the previous section can easily be augmented for Type k systems (systems with k integrators). Let's suppose that the process (3) has k integrators, $G_P^*(s)=G_P(s)\dfrac{1}{s^k}$. Using (1) for this type of systems we obtain

$$-\begin{bmatrix}P_i+jQ_i\end{bmatrix}=\frac{1}{G_P^*(j\omega_i)}$$

for the ith experiment, what leads us to

$$\frac{1}{G_P(j\omega_i)}=-\left(j\omega_i\right)^{-k}\left(P_i+jQ_i\right)=$$
$$=-\left(P_i^*+jQ_i^*\right)$$

The new describing function parameters can now be written in a matrix form as

$$\begin{bmatrix}P_i^*\\ Q_i^*\end{bmatrix}=\begin{bmatrix}0 & \omega_i^{-1}\\ -\omega_i^{-1} & 0\end{bmatrix}^k\begin{bmatrix}P_i\\ Q_i\end{bmatrix}$$

This approach is practical, because the original autotuning algorithm developed for Type 0 systems can easily be modified for Type k systems. The only parameters that have to be changed are P_i and Q_i, i.e. the describing function vector. This modification can be observed from

another point of view: we in fact joined the astatic part of the process to the nonlinear part; therefore converting the original nonlinear element into an inertial nonlinear element that depends upon the frequency of oscillations. If we look at it this way, then we can conclude that in fact the original describing function vector is rotated by 90° and multiplied by ω^{-1} for each integrator.

C. Algorithm Modification for Discrete-Time Systems

What type of modification is needed when dealing with discrete-time systems will be shown in the following example. When the autotuning identification is used in practice, the process is usually computer-controlled. Therefore a slight modification of the procedure has to be done. Let's say that the relay should switch when the input is 3 (i.e. x_a=3). It could happen that at some time step k the input is 2.15 and at k+1 it is 6.57, see Figure 3. Since at the moment k the input to the relay has not yet reached the switching value, it will switch at the moment k+1. In other words, this is equivalent to hysteresis parameter x_a having the value 6.57, which is more than double the value we supposed, therefore false results can be expected. It can also happen that the switching occurs exactly at the desired moment, resulting in accurate identification. Since it cannot be known a priori weather the chosen hysteresis parameters will give satisfactory results, the only way is to perform the correction of the hysteresis width, after the experiment has been performed.

To conclude we suggest the following procedure when dealing with discrete-time systems:

- perform the autotuning test with the relay width x_a;
- after acquiring the self-oscillations, determine the exact points in which the switching occurred (i.e. determine the new x_a^*);
- use the general algorithm with the corrected x_a^* parameter.

The simulation examples in the following section will confirm the improvement of the identification when this procedure is used.

D. Algorithm Modification for Systems with Delays

Delays in systems have great influence on quality of control. They are often present and seldom negligible. Some research has been done on autotuning of systems with time delays, but they were mostly based on inserting an additional time delay in order to shift the system in phase and therefore obtain different frequency points for identification. In this work, time delay is treated as a known part of the system, and its influence has to be compensated for.
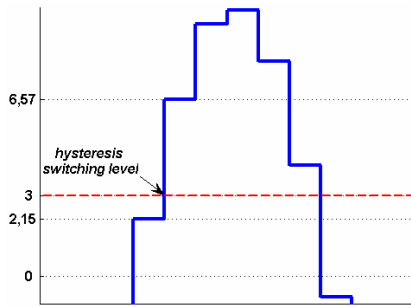

Figure 3.   Illustration of the relay switching error in discrete-time systems.

The influence of time delays is rather obvious in systems forced into self-oscillations. Systems with delays have greater magnitude and smaller frequency of oscillations. If the system with a delay is to be identified, the delay should be taken into account.

Let's suppose that the process (3) has a time delay, $G_P^*(s) = G_P(s)e^{-sT_d}$. Using the Goldfarb method (1) for closed-loop systems with nonlinear elements, we write the following equation.

$$\frac{1}{G_P(j\omega_i)} = -e^{-j\omega_i T_d}\left(P_i + jQ_i\right) =$$

$$= -\left(P_i^* + jQ_i^*\right)$$

The approach of combining the known time delay with the nonlinear element is elegant and does not require great intervention with the general algorithm. The case when time delay exists within the system is similar to the case with k integrators - it can be viewed as rotation of the nonlinear describing function vector, as shown in Figure 4. .

The new nonlinear element that should be used in the general algorithm, can be written in a matrix form as

$$\begin{bmatrix} P_i^* \\ Q_i^* \end{bmatrix} = \begin{bmatrix} \cos\omega_i T_d & \sin\omega_i T_d \\ -\sin\omega_i T_d & \cos\omega_i T_d \end{bmatrix}\begin{bmatrix} P_i \\ Q_i \end{bmatrix} = \mathbf{T}_d\begin{bmatrix} P_i \\ Q_i \end{bmatrix}.$$

Unlike the case with integrators, this transformation does not change the magnitude of describing function, but only its angle. In this case, the describing function can achieve all possible angles, based on the time delay. The following section will demonstrate the necessity of this modification on simulation examples.

IV.   Simulation Results

As it is shown, the general matrix representation of the algorithm is suitable for any order of a system. To demonstrate this, we use a transfer function of the following form:

$$G(s) = \frac{b_1 s + b_0}{\left(a_2 s^2 + a_1 s + 1\right)s},$$

where a_1=4, a_2=3, b_0=2 and b_1=1. Since the transfer function has 4 unknown parameters, two different experiments have to be performed. The identification results are shown in Table I.
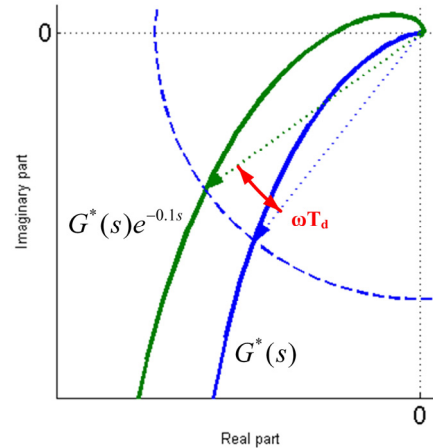

Figure 4.   Influence of the time delay – rotation of the describing function vector.

FILTERING	NO		YES	
x_a	1	1	1	1
C	1	2	1	2
ω_m	0.4889	0.5881	0.4889	0.5881
X_m	2.6904	3.7788	2.7181	3.8414
a_1 (error [%])	4.1201 (3)		4.0210 (0.53)	
a_2 (error [%])	3.8314 (27.7)		2.9628 (-1.24)	
b_0 (error [%])	1.9641 (-1.8)		2.0198 (1)	
b_1 (error [%])	1.4167 (41.7)		0.9646 (-3.54)	

As the function order gets bigger, greater precision in obtained self-oscillations parameters is needed. Therefore, signal filtering as described in section III-A is advised. Table I also gives results when the self-oscillations are filtered. The improvement is obvious.

A. Modification for Discrete-Time Systems

The example used here is $G(s) = \dfrac{50}{s(0.3s+1)}$ with sample time 0.1s.

When the hysteresis width x_a is chosen to be 16, the same oscillations are obtained as in the case when the width is e.g. 20 (see Table II.). In the first case the identification is far from exact values, while in the second case it is much better.

If algorithm modification proposed in section III-C is applied, we see that the relay switching occurs at the moments when hysteresis input is 20.28. When this hysteresis parameter is used in the identification algorithm, the identified parameters are the closest to real values.

Figure 5. shows results of simulations for hysteresis widths x_a from 5 to 20 with sample time 0.1. Void circles present percent parameter errors for the cases when hysteresis width was not modified. Full circles are positioned at the modified hysteresis widths and show percent parameter errors. It is self-evident that the results are much better when relay hysteresis is modified. For the hysteresis widths positioned between full circles, the magnitude and frequency of self-oscillations remain constant. It is obvious that with greater sampling time, one has more chance of choosing such hysteresis width so that real switching moment does not coincide with the assumed one.

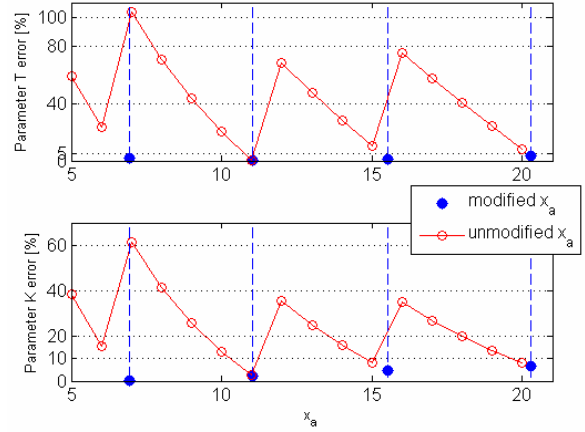| $X_m = 24.47;$ $T_u = 2.8$			
MODIFICATION	NO		YES
x_a	16	20	20.28
K (ERROR IN %)	67.43 (34.86)	53.94 (7.88)	53.2 (6.4)
T (ERROR IN %)	0.526 (75.33)	0.325 (8.33)	0.311 (3.67)

Figure 5. Comparison of identified parameter errors with and without algorithm modification for discrete-time systems (sampling time is 0.1s).
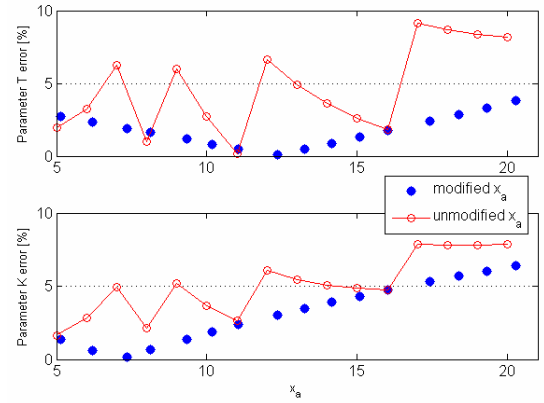

Figure 6. Comparison of identified parameter errors with and without algorithm modification (sampling time is 0.01s).

In the case when sampling time is much smaller, i.e. 0.01s (Figure 6.), we see that the cases where hysteresis width was modified also give better results. However, the results without modification give satisfactory errors (below 10%). This is expected, because as sampling time gets smaller, the chances of relay to switch at a significantly different input value are much smaller, hence the error is smaller.

B. Modification for Systems with Delays

Take a system given with $G(s) = \dfrac{50}{s(0.3s+1)}e^{-0.1s}$ as an example, and force it to self-oscillations. As it is seen from Table III, the errors that occurred when time delay was ignored are enormous. Once the algorithm was modified using the method described in section III-D, the results were much more accurate. Even though the improvement is obvious, the following part will show how filtering of these oscillations can improve the identification even more.

Now that the modification has been introduced, the same process can be even better identified if the oscillations are filtered. Figure 7. shows the percent error of two identified parameters for the system $G(s) = \dfrac{50}{s(0.3s+1)}e^{-T_d s}$, with time delays varying from 0.01 to 2.

TABLE III.   COMPARISON OF RESULTS WITH AND WITHOUT THE PROPOSED MODIFICATION FOR SYSTEMS WITH DELAYS

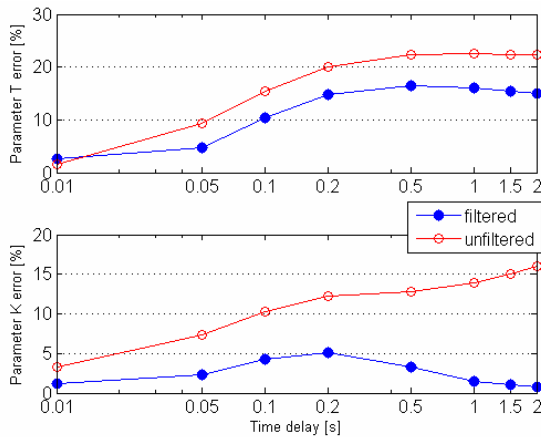| $x_a = 10;$ $C = 1;$ $X_m = 24.47;$ $T_u = 2.8;$		
MODIFICATION	NO	YES
rotation matrix	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0.964 & 0.266 \\ -0.266 & 0.964 \end{bmatrix}$
K (ERROR IN %)	76.98 (156.6)	55.14 (10.28)
T (ERROR IN %)	0.6037 (101.23)	0.346 (15.33)

Figure 7.   Comparison of identified parameter errors with and without filtering of the induced oscillations.

Full circles present the values obtained with filtering, while void circles give results without filtering. Obviously, filtering gives better identification results, especially in the case of the process gain.

V.   CONCLUSION

The paper gave insight to a general, matrix-formed, algorithm for the autotuning identification method. The method is based on finding enough frequency points on the Nyquist curve so that the supposed process transfer function parameters could be identified. Usually the downside to autotuning method is that only one point at the frequency curve is used to determine the process dynamics. With the proposed matrix method we can conduct many tests at different frequencies (by changing the relay parameters), and determine a more precise transfer function.

In addition to that, the algorithm is augmented in a simple manner for Type *k* systems, systems with delays and discrete-time systems. It is shown that more precise identification is achieved when modifications for discrete systems and systems with delays is introduced. In addition to that, filtering of self-oscillations is addressed, and improvements are shown on a simple example.

REFERENCES

[1] J. G. Ziegler and N. B. Nichols, "Optimum settings for automatic controllers," *Trans. ASME* 1942, *64*, 759,

[2] Z. Vukić and Lj. Kuljača, *Automatic control – analysis of linear systems*, 2005. (in Croatian)

[3] K. J. Åström and T. Hägglund, "Automatic tuning of simple regulators with specifications on phase and amplitude margins," *Automatica* 1984, *20*, 645.

[4] W. L. Luyben, "Derivation of transfer functions for highly nonlinear distillation columns," *Ind. Eng. Chem. Res.* 1987, *26*, 2490.

[5] W. Li, E. Eskinat and W. L. Luyben, "An improved autotune identification method," *Ind. Eng. Chem. Res.* 1991, *30*, 1530.

[6] R. S. Chang, and C. Y. Shen, "Derivation of transfer function from relay feedback systems," *Ind. Eng. Chem. Res.* 1992, *31*, 855.

[7] N. Mišković, Z. Vukić and M. Barišić, "ROV identification by use of self-oscillations" (submitted to IFAC CAMS 2007 conference)

[8] K. K. Tan, T. H. Lee and Q. G. Wang, "An enhanced automatic tuning procedure for PI/PID controllers for process control," *AIChE Journal* 1996, *42* (9), 2555.

[9] A. Besançon-Voda and H. Roux-Buisson, "Another version of the relay feedback experiment," *Journal of Process Control* 1997, *7* (4), 303.

[10] Y. H. Kim, "PI controller tuning using modified relay feedback method," *J. of Chem. Eng. of Japan* 1995, *28*, 118.

[11] Q. G. Wang, C. C. Hang and B. Zou, "Low-order modelling from relay feedback," *Ind. Eng. Chem. Res.* 1997, *36* (2), 375.

[12] C. C. Hang, Q. G. Wang and L. S. Cao, "Self-tuning Smith predictors for processes with long dead time. *Int. J. Adaptive Control and Signal Processing* 1995, *9* (3), 255.

[13] L. Wang, M. L. Desarmo and W.R. Cluett, "Real-time estimation of process frequency response and step response from relay feedback experiments," *Automatica* 1999, *35*, 1427.

[14] Q. G. Wang, C. C. Hang and Q. Bi, "Process frequency response estimation from relay feedback," *Control Engineering Practice* 1997, *5* (9), 1293.

[15] I. Kaya and D. P. Atherton, "Parameter estimation from relay autotuning with asymmetric limit cycle data," *Journal of Process Control* 2001, *11*, 429.

[16] H. J. Kwak, W. S. Sung and I.-B. Lee, "On-line process identification and autotuning for integrating processes," *Ind. Eng. Chem. Res.* 1997, *36*, 5329-5338

[17] Z. Vukić, Lj. Kuljača, D. Đonlagić and S. Tešnjak, *Nonlinear Control Systems,* Marcel Dekker Inc., 2003.

[18] N. Mišković, Z. Vukić, M. Barišić and B. Tovornik, "Autotuning autopilots for micro-ROVs," MED'2006, Ancona 28-30. June 2006.

[19] Y. Z. Tsypkin *Relay Control Systems,* Cambridge University Press: Cambridge, UK, 1984.