Int. J. Appl. Math. Comput. Sci., , *Vol.* , *No.* , –
DOI:

IMPLEMENTATION AND EVALUATION OF FINE-TUNING METHODS OF PI CONTROLLER WITH PLC-PID TUNER

JAKUB MOŻARYN [a], DAMIAN SUSKI [a], ARKADIUSZ WINNICKI [a]

[a] Institute of Automatic Control and Robotics
Warsaw University of Technology, Address ul. Sw. A. Boboli 8, 02-525, Warsaw, Poland
e-mail: j.mozaryn@mchtr.pw.edu.pl

In the paper, the method of an iterative modification of tuning parameters of a PI controller is presented. Proposed algorithms base on the off-line evaluation of quality indices. They were tested using PLC-PID Tuner - the application developed for tuning PID controller in S7 Simatic PLCs. It was shown that for some cases proposed iterative method of modifying tuning parameters might increase the control quality. Furthermore, the gain margin and phase margin achieved through this method are close to values recommended for practical applications.

Keywords: keyword 1, keyword 2, . . . , keyword 5. Please provide a few keywords (3–5) and keep them specific.

1. Introduction

Despite all the progress in advanced control, the PID (proportional-integral-derivative) algorithm remains the most popular. Its gains are often chosen independently by an engineer, based on his experience or through some simple selection methods such as *Ziegler-Nichols* (J.G. Ziegler, 1942) or *Cohen-Coon* (G.H. Cohen, 1953). However, regardless of the type of controlled process, there is usually requirement of exact response of the system. Without proper methodology of controller parameters tunig the quality of the control system may be unsatisfactory. Therefore many scientists and engineers develop tuning rules for different processes, where they consider different objectives. The survey presented in (O'Dwyer, 2006) gives the total of 1134 separate rules for PI and PID controllers. Such statistics were done in 2006, and one can expect that until now this number has increased significantly.

Nowadays, most of small and medium size production lines are controlled by Programmable Logic Controllers (PLC) which allow user to create the simple single loop control system as well as more complex structures. The main issue that ensures the correct functionality of the automation system is a proper adjustment of controller parameters. It should allow to minimize the deviation of the process variables from the desired output response, and therefore reduce the cost of the process due to the energy and material losses.

In many practical situations, it is impossible to develop one's own tuning rules. Popularity of PLC controllers result in the need to implement appropriate tools that support user during controller tuning. Unlike automation systems based on dedicated controllers such as Distributed Control System (DCS), solutions based on PLC don't have such wide support in terms of integration and operation of control algorithms.

Recently, manufacturers of automation systems develop advanced software tools that allow autotuning of PID controllers implemented in PLC's, using proprietary algorithms based on relay or area auto-tuning methods (Astrom and T., 1995), (Yu, 2006), (P. Tatjewski, 2011). Such modules which facilitate PID tuning are integrated within e.g. TIA Portal (Siemens), Automation Studio (B&R) or TwinCAT (Beckhoff) software. However, implemented methods don't allow designing control system in terms of specific performance indexes (R.H. Bishop, 2010), (J. Mozaryn, 2013) e.g. gain and phase margin, integral of the squared error (ISE) or the integral of the absolute value of error (IAE). Therefore, it has limited use in processes which apart from minimizing standard indices, must meet additional criteria like sensitivity or robustness.

Most applications that allow identification (e.g. IDCAD (Janiszowski, 2002), MATLAB/Simulink, LabVIEW) don't have automatic procedures for tuning the specific controller, especially in industrial environment where the additional software is required to communicate with PLC to collect and archive values of the process variables. After process identification, controller parameters can be calculated (O'Dwyer, 2009) and then implemented in PLC. Based on these premises the choice of proper method in industrial environment can be very complicated and time consuming. Therefore, the possibility to test, compare and use of wide variety of methods is very reduced.

These facts were the inspiration to create PLC-PIDTuner application which combines above mentioned functionalities: real process data collection using Simatic S7 1200/1500 controllers, SISO system identification, tuning of PID controller parameters with different methods, verification of the control system in time and frequency domains using various performance indices. PLC-PIDTuner simplifies the tuning rule selection process and can be used by the field engineers with basic knowledge of control engineering.

The article is organized as follows. In section II, the implementation details of PLC-PIDTuner are described. Then, in section III, the PLC controller tuning methodology is presented. Section IV describes results obtained during exemplary verification of developed application, based on a model of the 3rd order lag system with delay. Finally, in section V further development ideas are presented and in section VI concluding remarks are given..

2. Implementation details

PLC-PIDTuner is based on the open and free standards. Application allows tuning of `PID_Compact` function block implemented in SIMATIC S7-1200 and S7-1500 controllers which are relatively new products in Siemens SIMATIC S7 products line (, n.d.).

In PLC-PIDTuner there are implemented numerous methods, supporting different stages of the control system design:

2.1. PID controller structure. PLC-PIDTuner application considers PID algorithm implemented as follows:

$$y(s) = K_p \left[1 + \frac{1}{T_i s} + \frac{T_d s}{\frac{T_d}{k_d} s + 1} \right] e(s) \tag{1}$$

where: K_p, T_d, T_i are proportional, derivative and integral gains, k_d dynamic gain (during tests $k_d = 6$).

In `PID_Compact` function block in S7-1200 PLC there is implemented PID controller in 2-DOF form (Siemens, 2013):

$$\begin{aligned} y(s) = \quad & K_p \Big[(b y_{\mathrm{sp}}(s) - y(s)) + \tfrac{1}{T_i \cdot s}(e_{\mathrm{sp}}(s)) + \\ & + \tfrac{T_d s}{a T_d s + 1}(c y_{\mathrm{sp}}(s) - y(s)) \Big] \end{aligned} \tag{2}$$

Where a, b, c, are scaling parameters, and

$$e_{(sp)} = y_{\mathrm{sp}} - y(k) \tag{3}$$

Equation (2) can be converted into (1) by setting parameters $a = 1$, $b = 1$, $c = 1$.

2.2. Procedure of the controller parameters tuning with PLC-PIDTuner. Presented application divides controller tuning process into three main steps:

1. STEP 1: Process identification.
2. STEP 2: Calculation of the controller parameters.
3. STEP 3: Verification of the control system performance in time and frequency domains.

Thorough procedure divided into different sub-stages is presented in Fig. 1. Application contains simulation tools to verify the performance of the system in each sub-stage. Consequently, problems with the tuning process or decrease of an accuracy can be detected early and corrected.

.

There is possibility to extend PLC-PIDTuner functionality by implementing identification methods, tuning methods and performance indices in standardized, separate DLL libraries. Such approach allows adapting the application to user needs based on expert knowledge of controlled industrial process.

2.3. Process identification. The identification procedure provides mathematical model of process calculated on the basis of collected data of the control and process variables. It usually requires generation of proper excitation signal which affects industrial process and results in financial losses. Therefore, signal shape and parameters are often limited. PLC-PIDTuner allows coosing the suitable, least invasive excitation signal and set it's shape and parameters e.g. range of values or start/stop times of the step function.

In PLC-PIDTuner there are implemented following identification methods:

1. secant method (R. Haber, 1999),
2. tangent method (R. Haber, 1999),
3. *Strejc* method (Strejc, 1959),
4. extended least squares (ELS) method (, n.d.).

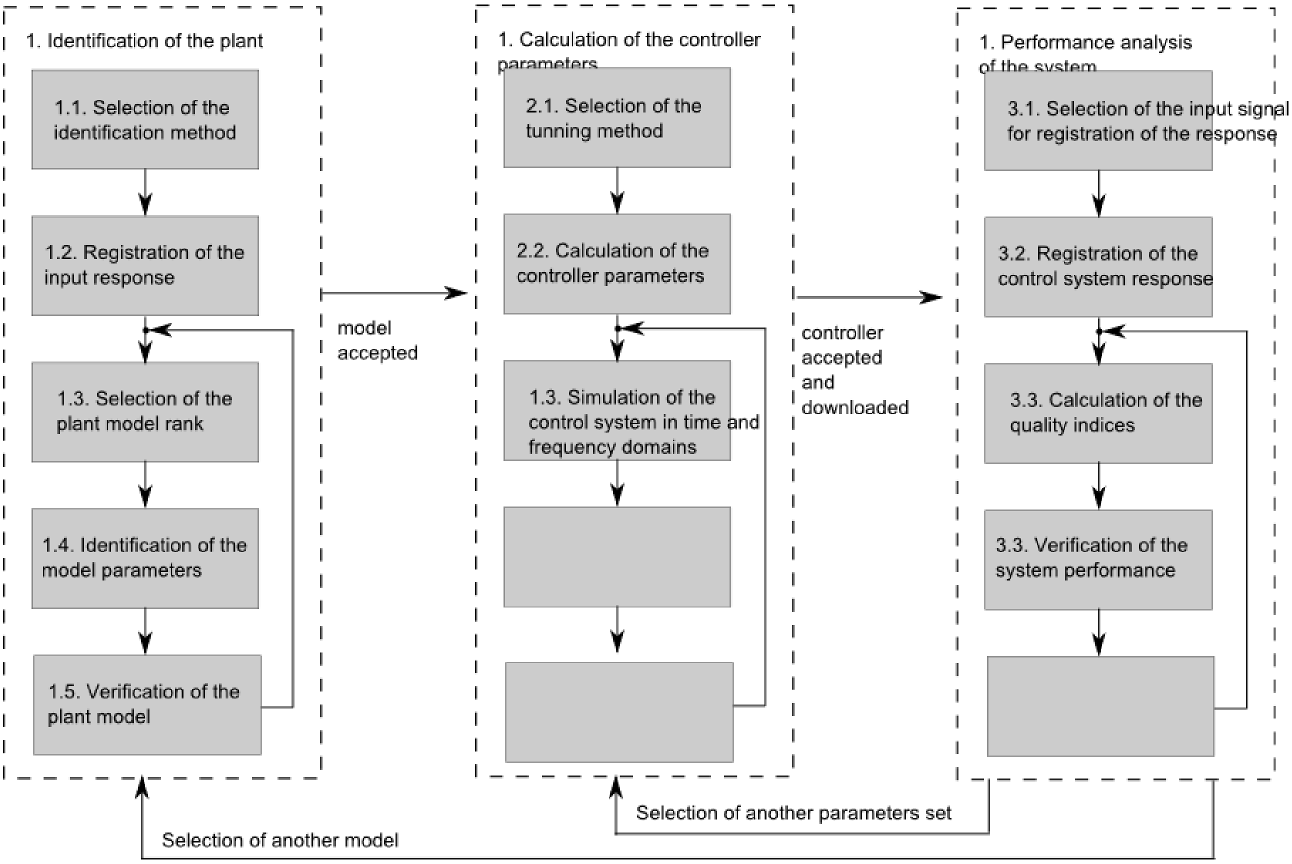
1. Identification of the plant
1.1. Selection of the identification method
1.2. Registration of the input response
1.3. Selection of the plant model rank
1.4. Identification of the model parameters
1.5. Verification of the plant model
model accepted
1. Calculation of the controller parameters
2.1. Selection of the tunning method
2.2. Calculation of the controller parameters
1.3. Simulation of the control system in time and frequency domains
controller accepted and downloaded
1. Performance analysis of the system
3.1. Selection of the input signal for registration of the response
3.2. Registration of the control system response
3.3. Calculation of the quality indices
3.3. Verification of the system performance
Selection of another parameters set
Selection of another model

Fig. 1. Procedure of PLC tuning process using PLC-PIDTuner.

Each method can be invoked in the application, and identified model parameters are saved for further controller tuning and process simulation. Moreover, user is able to select and compare several models identified by means of different methods.

Secant, tangent and *Strejc* methods are based on step response and specified mathematical model of the plant. Therefore they are mainly used for rough approximation of plant dynamics, which is often insufficient for the purpose of computer simulation or controller tuning. Therefore, ELS method is implemented, which is more accurate and easy in numerical implementation (, n.d.).

PLC-PIDTuner allows user to compare determined model with the real plant by automatically calculating the response of the model. It allows verifying the quality of identified model using several indexes e.g. Integral Squared Error (ISE), Integral Absolute Error (IAE), Integral Time-weighted Absolute Error (ITAE) and select the best of several models.

2.4. Tuning of the controller parameters.

Tuning of the controller parameters is performed on the basis of the process model identified in previous stage. There are impemented over 750 tuning methods, classified into groups based on the model type (O'Dwyer, 2009). Furthermore, user has the possibility to select proper tuning method in terms of:

- control algorithm (P, PI, PD, PID),
- type of tuning method (O'Dwyer, 2009),
- minimization of specified quality index,
- additional options e.g. required overshoot.

The set of the methods and calculated parameters, can be further examined using built-in time and frequency domains analysis modules. It should be emphasized, that all simulations are carried out with real PLC controller that communicates with PLC-PIDTuner.

Such functionality enables to choose the set of controller parameters more precisely. By modifying controller parameters and subsequently verifying control system response, user can adjust its properties to meet the control objectives. Furthermore, build-in simulation helps to verify designed control system without the need of costly experiments performed with the real process.

2.5. Analysis of the control system performance. After adjusting and downloading controller parameters, PLC-PIDTuner application allows user to verify operation of real control system. It is possible to plot and archive process variable values, change controller mode (Auto/Manual) or change set point value. Therefore, different quality indices in time domain can be calculated for the designed system. There were implemented 7 basic quality indices in time and frequency domain, described in detail in (Kolaj, 2015).

Build-in PLC-PIDTuner module for analysis in frequency domain allows operator to automatically draw *Bode* and *Nyquist* plots using simulation of the identified model. PLC-PIDTuner determines stability margins of the designed control system by calculating phase ($\Delta\phi$) and gain (ΔL) margins (R.H. Bishop, 2010) (D. Holejko, 2012). These values characterize the robustness of the control system to changes of the process parameters caused e.g. by aging of the actuators.

On the basis of collected data PLC-PIDTuner automatically calculates values of selected quality indexes. It allows user to rate real control systems operation in terms of certain properties. If real control system does not meet the requirements, the steps of controller parameters calculation or process mathematical model identification should be repeated.

2.5.1. Time domain analisys. The quality of the control system was analyzed in the time domain using following indexes:

- Maximum error

$$e_{\mathrm{sp\,max}} = \max_k [e_{\mathrm{sp}}(k)] \tag{4}$$

- Overshoot

$$\gamma = \left|\frac{e_2}{e_1}\right| 100\% \tag{5}$$

  where e_1 and e_2 are the first 2 consecutive biggest errors with opposite signs, assuming as the zero level (baseline) the steady state value of the output signal $y(t)$ after transient response.

- Transient response time T_R - it is the time between the moment of change of the set point $y_{\mathrm{sp}}(k)$, or introduction of disturbances $d(k)$ and the moment when error (2.5.1) reaches a fixed value inside a boundary $|0.05e_{\mathrm{sp\,max}}(k)|$.

- Integral Squared Error

$$\mathrm{ISE}_{\mathrm{sp}} = \sum_k [e_{\mathrm{sp}}(k)]^2 \tag{6}$$

- Integral of Absolute Value of Error

$$\mathrm{IAE}_{\mathrm{sp}} = \sum_k |e_{\mathrm{sp}}(k)| \tag{7}$$

2.5.2. Frequency domain analysis. In the next step, based on the simulation with model characterized by lowest indices, stability margins of the designed control systems are calculated. Gain margin determines how much the magnitude of the control system should increase, to make it unstable. Control systems with sufficiently large gain margin and phase margin are robust to changes in the parameters of the object and external distortions. Knowledge of the stability margins values allow also to adjust controller parameters in order to improve quality in the time domain (J. Mozaryn, 2013).

3. Fine-tuning of PID controller

Desired control system, was characterized by following properties:

- **Property 1:** Short T_R of the step response.

- **Property 2:** Short T_R after abrupt occurrence of disturbances.

- **Property 3:** Output signal should have no visible high-frequency oscillations.

To specify the robustness of the system there should be performed analysis in the frequency domain. In PLC-PID Tuner are calculated gain margin and phase margin of the control system (, n.d.). Gain margin determines how much the magnitude of the control system should increase, to make it unstable. Control systems with sufficiently large gain margin and phase margin are robust to changes in the pa-rameters of the object and external distortions. If ΔM and $\Delta\phi$ decrease, the control system has transient oscillatory character and the overshoot, but also has short transient response time.

In the proposed heuristic iterative modification procedures we propose to reduced chosen controller settings in order to comply with the following assumptions:

Assumption 1: Step response overshoot should be less than 5 %.

Assumption 2: The change of ΔM should be less than 25% of the original value and $\Delta\phi$ should be less than 25% of the original value.

Proposed PID fine-tuning iterative procedure can be presented in the form of algorithm as follows:

Algorithm 1. FT1 - Iterative modification of the proportional gain

Step 1. Set the iteration step $k = 0$. Enter the pre-tuned set of PID settings: k_{p0}, T_{I0}, T_{D0}, perform the step response experiment. Calculate and save values: γ_0, ΔM_0, and $\Delta\phi_0$. Set $k_{p,0} = k_{p0}$. If $\gamma_0 < 5\%$ go to step 2, else go to step 6.

Step 2. Set $k = k + 1$, and $k_{p,k} = 1,05 k_{p,k-1}$.

Step 3. Perform step response experiment, calculate values γ_k, ΔM_k, and $\Delta\phi_k$.

Step 4. If $\gamma_k < 5\%$, go to step 5, else go to step 6.

Step 5. If $\Delta M_k < (1 \pm 0,3) M_0$ and $\Delta\phi_k < (1 \pm 0,3)\phi_0$ go to step 2, else go to step 6.

Step 6. Set $k_p = k_{p,k-1}$.

Step 7. Stop the algorithm.

Algorithm 2. FT 2 - Iterative modification of the integral gain

Step 1. Set the iteration step $k = 0$. Enter the pre-tuned set of PID settings: k_{p0}, T_{I0}, T_{D0}, perform the step response experiment. Calculate and save values: γ_0, ΔM_0, and $\Delta\phi_0$. Set $T_{I,0} = T_{I0}$. If $\gamma_0 < 5\%$ go to step 2, else go to step 6.

Step 2. Set $k = k + 1$, and $T_{i,k} = 0,95 T_{I,k-1}$.

Step 3. Perform step response experiment, calculate values γ_k, ΔM_k, and $\Delta\phi_k$.

Step 4. If $\gamma_k < 5\%$, go to step 5, else go to step 6.

Step 5. If $\Delta M_k < (1 \pm 0,3) M_0$ and $\Delta\phi_k < (1 \pm 0,3)\phi_0$ go to step 2, else go to step 6.

Step 6. Set $T_I = T_{I,k-1}$.

Step 7. Stop the algorithm.

4. Expemental verification

4.1. Control system structure. The process was controlled in closed-loop control system, presented in Fig. 2.

In the control system (Fig. 2), an output of the process is the liquid level in the tank Z1 (H1). It is compared with the desired output response (setpoint) and an output error is given as:

$$e(t) = y(t) - w(t) \tag{8}$$

The error (1) is used by the controller to adjust the control signal which, through an actuator (pump) affects the regulated process. In the system there are disturbances (leakage) which can be introduced, as it is described in chapter 2.

The control algorithm was implemented on a Siemens S7-1200 PLC controller.

4.2. Identification of the control plant.

4.3. Selection of PID controller parameters.

4.4. Pre-tuning of PID controller. In (O'Dwyer, 2009) there are described 1731 different types of PID tuning rules, so the proper criteria should be used to choose the most suitable settings. Tuning rules selection (pre-tuning) criteria were as follows:

- a) Type of a process (stationary or nonstationary).
- b) Model of the process.
- c) PID controller structure.
- d) The ratio of transport delay and the time constant of the process:
$$R = \frac{T_d}{T_z} \tag{9}$$

4.5. Fine-tuning of PID controller. Obtained quality indices after the iterative modifications (FT1 and FT 2), are gathered in Table 2.

5. Conclusions

Acknowledgment

Please acknowledge here all and any support, institutional or individual, which you have received for your work.

References

(n.d.).

Astrom, K. and T., H. (1995). *PID Controllers: Theory, Design, and Tuning, 2nd Edition*, Instrument Society of America, Triangle Park, NC.

D. Holejko, W. K. (2012). *Automation of Continuous Processes (in polish)*, OWPW, Warsaw.

G.H. Cohen, G. C. (1953). Theoretical consideration of retarded control, *Transactions of the ASME* **75**(1): 827–834. DOI:.

J. Mozaryn, K. M. (2013). Tuning rules selection and iterative modification of pid control system parameters, *Proceedings of Mechatronics 2013 Conference, Brno, Czech Republic*, pp. 677–684.

Janiszowski, K. (2002). *Identification of Parametric Models in Examples (in polish)*, Exit, Warsaw, Poland.

J.G. Ziegler, N. N. (1942). Optimum settings for automatic controllers, *Transactions of the ASME* **64**(1): 759–768. DOI:.

Kolaj, W. (2015). *Design and Implementation of Application Supporting the Tuning of PID Controller in Simatic S7 Devices*, PhD thesis, Warsaw University of Technology, Warsaw.

O'Dwyer, A. (2006). Pi and pid controller tuning rules: an overview and personal perspective, *Proceedings of Irish Signals and Systems Conference*, pp. 161–166.

O'Dwyer, A. (2009). *Handbook of PI and PID Controller Tuning Rules*, Imperial College Press, London.

P. Tatjewski, L. Trybus, M. L. P. M. Z. S. A. S. (2011). *Supervisory Control and Optimization, Modeling, Diagnostics and Process Control. Implementation in the DiaSter System*, Springer, London.

R. Haber, L. K. (1999). *Nonlinear System Identification : Input-output Modeling Approach*, Kluwer Academic Publishers, Boston.

R.H. Bishop, R. D. (2010). *Modern Control Systems (12th ed)*, Prentice Hall.

Siemens (2013). Tia portal v13 user guide.

Strejc, V. (1959). Approximation aperiodisscher ubertragungscharakteristiken, *Regelungstechnik* **7**(1): 124–128. DOI:.

Yu, C. (2006). *Autotuning of PID Controllers: A Relay Feedback Approach*, Springer, London.

Jakub Możaryn. Place a brief biography here xxx xxx xxxxx xx xxx xxx xxx xx xxxx xx xxxxxxx xxx xxx xx xx xx xxxxx xx xxxx xxx xxx xxxxx xx xxx xxx xxx xx xxxx xx xxxxxxx xxx xxx xx xx xx xxxxx xx xxxx xxx xxx xxxxx xx xxx xxx xxx xx xxxx xx xxxxxxx xxx xxx xx xx xx xxxxx xx xxxx xxx xxx xxxxx xx xxx xxx xxx xx xxxx xx xxxxxxx xxx xxx xx xx xx xxxxx xx xxxx xxx xxx xxxxx xx xxx xxx xxx xx xxxx xx xxxxxxx xxx xxx xx xx xx xxxxx xx xxxx xxx xxx xxxxx xx xxx xxx xxx xx xxxx xx xxxxxxx xxx xxx xx xx xx xxxxx xx xxxx xxx xxx xxxxx xx xxx xxx xxx xx xxxx xx xxxxxxx xxx xxx xx xx xx xxxxx xx xxxx.

Damian Suski. Place a brief biography here xxx xxx xxxxx xx xxx xxx xxx xx xxxx xx xxxxxxx xxx xxx xx xx xx xxxxx xx xxxx xxx xxx xxxxx xx xxx xxx xxx xx xxxx xx xxxxxxx xxx xxx xx xx xx xxxxx xx xxxx xxx xxx xxxxx xx xxx xxx xxx xx xxxx xx xxxxxxx xxx xxx xx xx xx xxxxx xx xxxx xxx xxx xxxxx xx xxx xxx xxx xx xxxx xx xxxxxxx xxx xxx xx xx xx xxxxx xx xxxx xxx xxx xxxxx xx xxx xxx xxx xx xxxx xx xxxxxxx xxx xxx xx xx xx xxxxx xx xxxx xxx xxx xxxxx xx xxx xxx xxx xx xxxx xx xxxxxxx xxx xxx xx xx xx xxxxx xx xxxx xxx xxx xxxxx xx xxx xxx xxx xx xxxx xx xxxxxxx xxx xxx xx xx xx xxxxx xx xxxx.

Arkadiusz Winnicki. Place a brief biography here xxx xxx xxxxx xx xxx xxx xxx xx xxxx xx xxxxxxx xxx xxx xx xx xx xxxxx xx xxxx xxx xxx xxxxx xx xxx xxx xxx xx xxxx xx xxxxxxx xxx xxx xx xx xx xxxxx xx xxxx xxx xxx xxxxx xx xxx xxx xxx xx xxxx xx xxxxxxx xxx xxx xx xx xx xxxxx xx xxxx xxx xxx xxxxx xx xxx xxx xxx xx xxxx xx xxxxxxx xxx xxx xx xx xx xxxxx xx xxxx xxx xxx xxxxx xx xxx xxx xxx xx xxxx xx xxxxxxx xxx xxx xx xx xx xxxxx xx xxxx xxx xxx xxxxx xx xxx xxx xxx xx xxxx xx xxxxxxx xxx xxx xx xx xx xxxxx xx xxxx xxx xxx xxxxx xx xxx xxx xxx xx xxxx xx xxxxxxx xxx xxx xx xx xx xxxxx xx xxxx.