Integrating Security in Process Control Loop: A Case for Proportional-Integral-Derivative Algorithm

Omagbemi Oghogho Weyinmi¹ and Mbonu Ekene Samuel²*

¹Department of Physics, Federal University of Petroleum Resources, Effurun, Nigeria.
²Department of Mechatronics, Federal University of Technology, Owerri, Nigeria.

Authors’ contributions

This work was carried out in collaboration between both authors. Author OOW managed the literature searches and used Simulink to analyze the existing proportional-integral-derivative (PID) control algorithm, thus discovering the security gap inherent in the algorithm. Author MES developed the mathematical model for embedding security in the existing PID algorithm. Both authors read and approved the final manuscript.

ABSTRACT

This work presented mathematical models and flow charts for implementing secure Proportional-Integral-Derivative (PID) algorithm in a process control loop. A number of security solutions have been recommended and some deployed in a process control system. Majority of these solutions are network based while others leverage on good security policy. A security solution based on network can be effective for securing control system from external threat agents who have to first of all, gain access to the control network. But for an internal threat agent or a disgruntled insider who does not only have the right privilege but also has a good understanding of the control system’s operation, a network security is definitely not going to be effective. This work used system analysis to identify the possible things an internal threat agent can do to manipulate a control system using PID control algorithm as a case study. A secured PID mathematical model is proposed as a proactive mitigation technique to embedding security in a process control loop. As ongoing research, future work will concentrate on simulation and prototyping of the secured algorithm.
presented in this work. The proposed secured algorithm will not only serve as an additional security layer in industrial control system (ICS) but will also be relevant in the control domain of Internet of Things.

**Keywords:** Disgruntled insiders; internal threat agents; process control loop; Proportional-Integral-Derivative algorithm secure mathematical models.

### 1. INTRODUCTION

Control systems were originally designed to be isolated entities that had nothing to do with internet networks [1]. The risk of cyber-attacks on control systems was almost zero then, but the isolated systems did not make for optimal supervision, efficient data mining and intelligent business decisions among other benefits that supervisory and distributed control offer [1,2]. To overcome these challenges, Information, Communication and Technology (ICT) based solutions were deployed in control system especially in industrial domain [2]. These solutions brought undeniable benefits to control system practice in industries but not without its attending problems, the major one being exposure of industrial control system to threat agents inherent in the ICT solutions [3]. A number of efforts have been made to secure industrial control system. International bodies like National Institute of Standard and Technology (NIST), International Society of Automation (ISA), International Electrotechnical Commission (IEC), United States Cyber Emergency Response Team (US-CERT) among others have made a number of recommendations for control system security [4,5,6]. These recommendations however seem to be geared towards protecting industrial control systems (ICS) from external threat agents. Thus the recommendations usually center on network security, developing and maintaining good security policies supposing that internal agents will always have good intentions.

One of the lessons learnt from stuxnet attack is that the major threat to ICS security is an insider who has a good understanding of the working principles of a control system [7]. Such individuals when disgruntled can manipulate a control system that does not have embedded security. In this work, control loop, the basic building block of a control system was examined considering the algorithm of control. The vulnerable parameters of the algorithm were identified, and secure models to mitigate the identified vulnerabilities were proposed as a solution. This secure solution promises to protect ICS from both internal and external threat agents, and in future will be relevant to control system security in the domain of Internet of Things (IoT). The rest of the work is organized as follows. Section 2 reviewed network segmentation as an ICS security solution pointing out that it cannot prevent a disgruntled insider from attacking a control system. Section 3 analyzed process control system loop pointing out the parameters that could be manipulated to cause abnormal operations in control system. The consequences of such manipulations were also stated. Section 4 dealt with the mathematical modeling of the proposed security solution detailing the flow charts for implementation of the proposed solution in section 5 while section 6 is the conclusion.

### 2. REVIEW OF RELATED WORKS

Industrial control system security is not actually new. Some of the major security solutions that have been deployed in industry are based on network segmentation leveraging on defense-in-depth architectures [1,4,8,9,10,11,12]. Fig. 1 shows a typical implementation of network segmentation in industrial control system, an excerpt from Design and Engineering Practice (DEP) of Shell Nigeria Exploration and Production Company, SNEPCo [13].

The system is made up of the work stations or the hosts, the network security elements represented by Virtual Local Area Networks (VLAN) and firewall, the instrumented protective system (IPS) representing a control loop. It is clear from Fig. 1 that the IPS does not have any form of security inherent in it. The hosts comprising the instrument asset management system (IAMS), distributed control system (DCS) controller work station, operator’s work station and engineering work station (EWS) have application level security [13]. To have access to EWS for example, the operator has to enter his password correctly and can only access limited applications depending on the level of privilege given to him. Let us consider a case of an operator who has the right to change control system parameters. These changes are usually
done and sent to control system without further security checks on the integrity of the action. This is in line with ICS security implementation policy given the real time requirements of ICS [14]. The operator is usually given the necessary trainings that will make for competent decisions and actions, and so is trusted to do the right thing [14]. Even though his actions are logged, the operator is a potential threat agent and can decide to sabotage the system and face the disciplinary consequences. He may even do the wrong thing inadvertently. The VLAN together with the firewall makes sure that it is only the right host and applications that talk to the IPS. While VLAN uses internet protocol (IP) subnetting to actualize its purpose [15], firewall confirms the right application by checking the port numbers of the applications before granting the user access to IPS [16]. A pictorial representation of a typical control system data frame and security checks done on it is given in Fig. 2.

Fig. 1. Typical implementation of network segmentation in industrial control system

Fig. 2. The security logic for a typical control system data frame
From Fig. 2, it is clear that VLAN and firewall cannot detect variations in process data whether such variations are legitimate or not. There are versions of firewalls that carry out in-depth check on a frame but they are not recommended for process control system due to false alarm usually associated with such firewalls [14,16].

3. ANALYSIS OF CONTROL SYSTEM LOOP

3.1 The Simulink Model of a Feedback Control Loop

A typical feedback control loop is made up of five major elements namely the reference point, controller, final control elements, process plant and the feedback elements [17,18,19]. This is shown in Fig. 3. The reference point, is the control objective of the loop while the controller contains the algorithm that achieves the control objective through the final control elements. The process plant is the system under control; the controller ascertains the real time state of the process through the feedback elements. In this analysis, acid gas removal from natural gas is considered. The process as shown in Fig. 4, involves passing the natural gas (sour gas) through an absorber at a controlled temperature of 40ºC [20,21]. The forward reactions in the absorber chamber are shown in (1) and (2) [22]. The control algorithm is shown in (3) while the open loop transfer function of the system is given by (4) [23,24,25,26].
\[ \text{C}_2\text{H}_7\text{NO} \text{ (liquid)} + \text{H}_2\text{S} \text{ (aq)} \rightarrow 38-43\degree \text{C} \rightarrow \text{C}_2\text{H}_7\text{NO} \cdot \text{H}_2\text{S} \quad (1) \]

\[ \text{C}_2\text{H}_7\text{NO} \text{ (liquid)} + \text{CO}_2 \text{ (aq)} \rightarrow 38-43\degree \text{C} \rightarrow \text{C}_2\text{H}_7\text{NO} \cdot \text{CO}_2 \quad (2) \]

The tuned model of the system is generated in Simulink as shown in Fig. 6 with \( K_p = 1.85; \ K_i = 0.06 \) and \( K_d = 2.51 \)

The response of the system (Fig. 5) is shown in Fig. 7. It shows that the control system was able to achieve the control objective in 75\(^{th}\) minute.

\[ G(s) = \frac{K}{(\tau s + 1)} e^{-Ls} \quad (4) \]

Fig. 5 showed how a temperature control system can be characterized in Simulink. The tuned model of the system was generated by tuning the controller using a software tool in Simulink. In practice, such a control system is tuned by varying the controller’s gains while keeping the control objective (reference point) constant until an optimal result is achieved \([28,29]\). This is usually done at the workstation connected to the controller in a distributed control system (DCS).
Fig. 6. The tuned model of the temperature control system

Fig. 7. The response characteristics of the Simulink model of the temperature control system for the acid gas removal
Table 1. The implications of changing controller’s parameters in a process control

<table>
<thead>
<tr>
<th>Action</th>
<th>Observations</th>
<th>Implications or Consequences on a Typical Company’s Business</th>
</tr>
</thead>
<tbody>
<tr>
<td>The operating point was changed to 10°C</td>
<td>The control system could not converge at 40°C but rather the system temperature was found to stabilize at 10°C as shown in Fig. 9</td>
<td>The absorption of H₂S and CO₂ from the produced natural gas will not be optimal, so the natural gas will still contain these acid gases. The acid gases will in turn lead to fast corrosion of production equipment and transmission pipelines. This will lead to more frequent maintenance of production facilities, and the production loss which is usually associated with maintenance shut down will increase. Of more concern is the quality of the product which has been compromised. This will impact negatively on the company’s integrity and may attract sanctions due to regulatory violations. Besides, poor quality product will lead to poor customer satisfaction and the eventual loss of competitive advantage.</td>
</tr>
<tr>
<td>The operating point was changed to 200°C</td>
<td>As shown in Fig. 10, the control system did not regulate until the temperature rose to 200°C</td>
<td>This is dangerous! If the Safety Shut Down System (SSDS) is not also compromised, this will lead to shutdown of process operations leading to production loss. In a situation whereby the attack is from intelligent hackers, it could be repeated after a calculated period of time. This in deed would have escalated consequences. Now if the SSDS is also compromised such that it did not detect such abnormal temperature, then there would be explosion leading to major safety event. Consequences of such event range from loss of life, properties, security event, and oil spillage to withdrawal of license to operate due regulatory violations.</td>
</tr>
<tr>
<td>The PID parameters of the controller were varied without changing the operating point.</td>
<td>As shown in Fig. 8, the combination of ( K_c = 1.1015 ), ( K_i = 0.0607 ), and ( K_d = 2.5120 ) cause the control system to oscillate. With ( K_c = 1.8453 ), ( K_i = 0.0001 ), and ( K_d = 2.5120 ), the control system converged at temperature of 27°C. With ( K_c = 1.8453 ), ( K_i = 0.0607 ), and ( K_d = 5.1251 ), the control system was able to converge at temperature of 40°C. With ( K_c = 1.8453 ), ( K_i = 0.0607 ), and ( K_d = 0.0020 ), the control system was also able to converge at temperature of 40°C.</td>
<td>This scenario is revealing. It shows that by varying the PID parameters, the control system performance can be varied in a subtle manner. The system could be made to oscillate or misbehave for some time and then go back to normal operation after a while. If this kind of parameter variation is emulated by an attacker, then the control system could be under attacks for months without anybody knowing it. The effect would be variation in product quality that might be difficult to explain. It can lead to frequent changing of actuators or even the gas absorber with the notion that those components might be faulty. This indeed can cause the company to spend much money in maintenance without success. Ultimately it will lead to the sabotage of the company’s business efforts.</td>
</tr>
</tbody>
</table>
3.2 Vulnerabilities Inherent in the Control System Loop

Once the optimal result is achieved, the tuned parameters are passed on to the controller and the controller accepts the tuned parameters without questioning. “In this section, the effects the parameters will have on the control system performance were examined by changing the parameters’ values and then plotting the response of the system in each case”. Fig. 8 shows the responses of the system at various selected gain parameters with constant reference point. Fig. 9 shows the response of the system with the optimal tuned controller’s parameters but with reference point changed to 10°C. Fig. 10 is a case where the reference point was changed to 200°C without changing the controller’s parameters.

Since there is no security in the system’s control loop being implemented by the controller, it means an internal threat agent or external threat agent who gains access to the control system’s work station can actually manipulate a control system at will. Figs. 8, 9 and 10 represent a few of the several things the threat agent may want to do to a control system. Table 1 is a summary of the actions depicted in Figs. 8, 9 and 10, and their implications to a typical crude oil production scenario.

4. MODELING SECURE PID ALGORITHM

The existing equation for implementing PID control is given by (3) [27,29].

A close observation of (3) reveals that there are four things that can affect the output \( m \) of the controller: the proportional gain, integral gain, derivative gain and the error variable. The mathematical expression of the error variable is given by (5).

\[
e = s - b
\]  

Where \( s \) is the reference or set point of the control system and \( b \) is the feedback signal.

The set point and the PID gains are the parameters accessible from the workstation, thus securing PID algorithm shall entail securing these parameters against unauthorized
manipulation at the controller’s level. In order to incorporate security in (3), the following recommendations are made.

1) Equation (3) should be modified such that it will be self-diagnostic without impacting on system’s availability and integrity.
2) The parameters should be bounded within defined limits at the controller level.
3) The equation should be able to resist unauthorized modification without impacting on system’s availability and integrity.

4.1 Security Model for Implementing Reference Point in Process Control

Control system usually operates within a control envelop defined by minimum and maximum values of the reference point. A secure control system should not operate outside the valid control envelope under any condition. This can be achieved by defining the boundaries of operation within the nonvolatile memory of the controller and then incorporating the defined boundaries in control loop logic of the system. The model that defines the reference boundaries in a typical control system is given by (6).

\[
s_{\text{new}} = s_{\text{new}} \quad \text{for} \quad S_{\text{min, setpoint}} \leq s_{\text{new}} \leq S_{\text{max, setpoint}} \quad \text{at} \quad t = t_i
\]

\[
= S_{\text{default}} \quad \text{for} \quad S_{\text{new}} < S_{\text{min, setpoint}} \quad \text{or} \quad S_{\text{new}} > S_{\text{max, setpoint}} \quad \text{at} \quad t = t_i
\]

Where \( i = 1, 2, 3, \ldots, n \)

Essentially (6) states that the set point \( s_{\text{new}} \), of a feedback control system at any time \( t = t_i \), must fall within a defined bounded range otherwise the system will assume a predefined default value. \( n \) is the total number of changes made to the set point parameter throughout the controller’s operating time.

4.2 Security Model for Implementing PID Gains in Process Control

It was noted in section 3 that changing the PID gains or parameters for a process controller can prevent the control system from achieving its control objective. So developing a secured PID algorithm entails specifying a secure way of changing these parameters and building intelligence into the controller so that it can detect when unauthorized changes are made and then take proactive measures to mitigate the changes. Equation (6) will take care of invalid changes in the reference point. One of the ways the controller can prevent unauthorized changes is by having a memory of its last legitimate output \( m_{\text{new}} \) so that it can fall back to it when unauthorized changes are detected. Mathematically, this can be represented as shown in (7).

\[
m = m_{\text{new}} \quad \text{For (legitimate parameters) at} \quad t = t_i
\]

\[
= m_{\text{new}}-1 \quad \text{For (illegitimate parameters) at} \quad t = t_i
\]

where \( i = 1, 2, 3, \ldots, n \)

\( m_{\text{new}} \) is the present value of the controller’s output at time \( t \).

Equation (7) is a high level definition of the secured PID algorithm. The equation can be detailed further by formulating what should constitute legitimate parameters. It is therefore reasonable to define domains for acceptable gains so that the controller can always check for the integrity of any gain parameter before utilizing it in its control loop. The domains for the PID parameters are defined below in (8) to (10).

\[
D_{\text{pid, kp}} = \{K_{p1}, K_{p2}, K_{p3}, \ldots, K_{pnp}\}
\]

(8)

Where \( np \) is the total number of permissible \( K_p \) parameter.

\[
D_{\text{pid, ki}} = \{K_{i1}, K_{i2}, K_{i3}, \ldots, K_{ini}\}
\]

(9)

Where \( ni \) is the total number of permissible \( K_i \) parameter.

\[
D_{\text{pid, kd}} = \{K_{d1}, K_{d2}, K_{d3}, \ldots, K_{dnd}\}
\]

(10)

Where \( nd \) is the total number of permissible \( K_d \) parameter.

Equations (8) to (10) are integrity equations that will help the controller to make sure that the changed parameter values are within the confines of the domain families.

Having defined the domains of the PID parameters, a security operator \( \# \) is introduced in (7). That is,
\[ \#m = \#m_{\text{new}} \quad \text{For \ } t = t_i \]
\[ = \#m_{\text{new-1}} \quad \text{For \ } t = t_i \]  
(11)

where \( i = 1, 2, 3, \ldots, n \)

The security operator \( \phi \) will ensure that the changes made to PID parameters are within the defined domains. This is done by leveraging on a function \( R_{\text{pidrange}} \), that operates on the PID parameters and returns a positive or negative value depending on whether all the parameters’ ranges satisfy the required condition. Thus,

\[ \phi = f(R_{\text{pidrange}}) \]  
(12)

From (3), PID algorithm is a summation of three different entities namely: proportional, integral and derivative entities with individual parameters. Thus \( R_{\text{pidrange}} \) can be represented mathematically as

\[ R_{\text{pidrange}} = \theta (K_p + K_i + K_d) \]
\[ = \theta K_p + \theta K_i + \theta K_d \]

\( \theta \) is an operator that checks whether the parameter value is a member of the parameter domain, \( D_{\text{pid}} \) as stated in (8) to (10). The value of \( R_{\text{pidrange}} \) can either be a YES or a NO. In logical terms a 1 or a 0.

Thus logically,

\[ R_{\text{pidrange, v}} = \theta K_p \cdot \theta K_i \cdot \theta K_d \]  
(13)

(. ) is AND operator.

In order to maintain system availability, there is need to also tell the security operator, \( \phi \) what to do if \( R_{\text{pidrange, v}} \) becomes zero. Thus availability operator \( \Delta \) is introduced. \( \Delta \) is such that when it acts on \( \phi \) it will cause the controller to revert to \( K_{\text{pid, new-1}} \) when \( R_{\text{pidrange, v}} \) becomes zero. Thus mathematically,

\[ \Delta \rightarrow f(R_{\text{pidrange}}); \quad K_{\text{pid, new}} = K_{\text{pid, new}} \]

(14)

\[ \text{(for } D_{\text{min, pid}} \leq y_{\text{pid}} \leq D_{\text{max, pid}}, \text{i.e. } (R_{\text{pidrange, v}}) = 1 \) \]
[\[ = K_{\text{pid, new-1}} \quad \text{(for } R_{\text{pidrange, v}} = 0) \]  
(15)

\( y_{\text{pid}} \) is a scalar that contains the chosen values of the new PID parameters to be changed;

\( K_{\text{pid, new-1}} \) is the last valid values of the PID parameters.

Putting everything together in one equation, the output of a secure PID controller can thus be represented mathematically as

\[ M_{\text{pid}} = \Delta \#m \]  
(16)

Where \( \Delta \) is the availability operator that makes sure the security solution does not impact negatively on the system availability; \( \phi \) is a security operator that acts on \( m \) according to the previous defined equations.

Integrating (16) into (3), it implies

\[ M_{\text{pid}} = \Delta \phi (K_p e + K_i \int e \, dt + K_d \frac{de}{dt}) \]

\[ M_{\text{pid}} = \Delta \phi K_p e + \Delta \phi K_i \int e \, dt + \Delta \phi K_d \frac{de}{dt} \]  
(17)

Equation (17) is the secure equation for implementing PID algorithm at the controller’s level in a process control domain. It is a mathematical security equation that specifies how to implement secured PID algorithm at the controller’s level in a process control loop.

5. FLOW CHARTS FOR IMPLEMENTING THE SECURE PID ALGORITHM IN A CONTROL LOOP

In section 4, the mathematical models for implementing secure PID algorithm in a control loop were developed. Equation (6) is the model for implementing secure reference point while (17) is the model for implementing secure PID algorithm. Fig. 11 is a high level flow chart for implementing secure PID algorithm in a process control loop. Essentially it is made up of four major functions which call the subroutines that independently calculate the appropriate error, proportional, integral and derivative terms. The security equations are integrated in the functions as shown in Figs. 12 to 15. Fig. 12 is a high level flow chart for calculation of secure reference point in a control loop while Fig. 13 is a flow chart that shows how to calculate secure integral and derivative terms respectively. Figs. 14 and 15 show how to calculate secure integral and derivative terms respectively.
Fig. 11. High level flow chart for implementing secure PID algorithm in a process control loop
Fig. 12. High level flow chart for calculation of secure reference point in a control loop
Fig. 13. High level flow chart for calculation of secure proportional term in a PID control loop
Fig. 14. High level flow chart for calculation of secure integral term in a PID control loop
Fig. 15. High level flow chart for calculation of secure derivative term in a PID control loop
6. CONCLUSION

Mathematical models and flow charts for implementing secure PID algorithm in a process control loop have been presented in this work. The key requirement for deploying any security solution in process control domain (PCD) is that the solution must not impact meaningfully on the control system’s availability and integrity. The security models presented in this work has not been tested. An ongoing research focuses on developing a prototype control system that will be used to validate the concepts presented in this work.

COMPETING INTERESTS

Authors have declared that no competing interests exist.

REFERENCES

2. Bianca Scholten. MES guide for executives: Why and how to select, implement and maintain a manufacturing executive system. ISA; 2009.


© 2019 Weyinmi and Samuel; This is an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Peer-review history:
The peer review history for this paper can be accessed here:
http://www.sdiarticle3.com/review-history/45577