Abstract- Distributed network-control-systems (D-NCS) are a multidisciplinary effort whose aim is to produce a network structure and components that are capable of integrating sensors, actuators, communication, and control algorithms in a manner to suit real-time applications. They have been gaining popularity due to their high potential in widespread applications and becoming more realizable due to the rapid advancements in wireless communication and data transfer technologies. This paper addresses the issue of D-NCS information security as well its real-time performance with respect to network security protocols and encryption schemes. We use a wireless network based, robot navigation path tracking system called Intelligent Space (iSpace) as a D-NCS test bed in this paper. The paper classifies the data from every NCS module (sensors, actuators and controllers) according to bandwidth requirement, time and information sensitivity. We define performance parameters for this NCS test bed. Various system factors affecting these performance parameters are recognized. Network security algorithms DES and 3DES are integrated with the application to secure the sensitive information flow. These wireless security features are considered as an added factor to the NCS. Standard statistical approach (2^k factorial experiment design) is used to study and estimate the effect of each factor on the system performance especially security additions. Thorough experimental results, tables of detailed characterization and effect estimate analysis is presented followed by the discussion on the performance comparison of NCS with and without wireless security.

Index terms – Distributed network-control-system, iSpace, data-sensitive wireless network, data security.

I. INTRODUCTION

A distributed network-controlled-system (D-NCS) effectively uses distributed sensors, actuators, computing processors and information technology over a communication network. Connecting the distributed control system components via a network can effectively reduce the complexity of the systems with nominal economical investments making the system scalable with efficient sharing of data. D-NCS makes it easier to fuse global information and to take intelligent decisions over a large physical space. Research in tele-operation, initiated with the concern of safety in hazardous environments such as spaces and nuclear reactor plants, gave rise to a wide variety of applications like monitoring and operating manufacturing plants, robot navigations, space projects, nursing homes, military applications and many more over the years. The synergistic collaboration between industry and academia has led to advanced NCS especially in the field of Industrial automation. This became possible because of the tremendous increase in the deployments of wireless systems. With the rapid development of wireless networks and information technology, many organizations and individuals are utilizing wireless systems for transmitting sensitive and critical data. As wireless medium is susceptible to easy intercepting, it is critical to protect transmitted data from unauthorized access and modifications in wireless systems. Wireless network security includes essential elements in Internet security devices that provide traffic filtering, integrity, confidentiality, and authentication. Therefore data sharing, and data/network security is of utmost concern in D-NCS considering the time and data sensitive applications. Most of these NCS that are not designed with security protection can be vulnerable to network attacks nowadays, so there is a growing demand of efficient and scalable intrusion detection systems (IDS) in the network infrastructure of industrial plants. However, information security and data sensitivity issues have not been addressed enough to be applied in a real-time D-NCS. Using security in an NCS gives rise to many topics like network architecture to support security for NCS, the performance assessment for NCS with security, studying NCS for adding network security etc. Chi-Ho and Kwong [9] propose an efficient and biologically inspired learning model for anomaly intrusion detection in the multi-agent IDS of industrial plants. Creery and Byres [9] present methods to determine and reduce the vulnerability of NCS to malicious intrusions through thorough assessment of process control networks and identifying the security issues to mitigate these risks for an industrial plant. Xu et al. [10] developed core architecture to address the collaborative control issues of distributed device networks under open and dynamic environments by adopting policy-based network security technologies. Gupta et al. [11] characterize the wireless NCS application on the basis of security effect on NCS performance to show the trade-off between security addition and real-time operation of D-NCS. It is highly required to study the performance effects in NCS due to different security additions. This gives rise to an optimization problem in NCS with an additional state of network security. To estimate the effect of addition of security in NCS, its properties based on data-sensitivity and network delay effect and the requirement for the network security, we developed a closed-loop network-controlled navigation system called intelligent space (iSpace) as a testbed. It consists of different modules like image processing, automation control, communication and networking for navigation of a differential drive unmanned ground vehicle (UGV) in an indoor 2-D environment. We illustrate the procedure of the characterization using factorial experiment design for time-sensitive applications on iSpace platform.

Section II describes the iSpace platform as a D-NCS and the different components involved in it. It also includes description of the wireless system and the encryption algorithms implemented in iSpace. Section III explains in detail how the experiments are designed in iSpace using statistical 2^k factorial design approach. Section IV elaborates on the different experimental observations and the analysis done to estimate the effect of security on the performance of the NCS. We conclude in and with Section V.

II. SYSTEM DESCRIPTION AND IMPLEMENTATION

In the wireless systems, several security protocols such as wired equivalent privacy (WEP) and 802.1x port access control with extensible authentication protocol (EAP) support are proposed to address security issues [3][2],

Information security with real-time operation: performance assessment for next generation wireless distributed networked-control-systems

Rachana A Gupta, Mo-Yuen Chow, Avesh Kumar Agarwal, Wenye Wang

rgupta@ncsu.edu, chow@ncsu.edu, akagarwa@ncsu.edu, wwang@ncsu.edu

Electrical and Computer Engineering Department, North Carolina State University, Raleigh, NC-27606, USA
Moreover, due to the strong security it provides in wired networks, the IP security protocol (IPsec) is considered as a good option for wireless systems as well. Some basic security features associated with these security protocols as defined by IETF for wireless systems are [7][1]: Authentication, Confidentiality, Integrity and non-repudiation. In this paper, we focus on confidentiality of security service. Other security features such as authentication, integrity and non-repudiation will be discussed in future work.

As described briefly in Section I, a distributed NCS consists of many different components such as sensors to collect data, controllers to process data in the desired manner and actuators to perform tasks asked by the controllers as shown in Figure 1.

![Figure 1: Describing iSpace concept as a D-NCS.](image)

The key point in building such a system is that the operational intelligence lie on one or many distributed controllers and all these components are connected over a network – these days Internet – to achieve almost limitless virtual space. In this paper, we are using one of the kinds of NCS - Intelligent Space (iSpace) to characterize for confidentiality aspect of security service in D-NCS. We have used iSpace as a network based intelligent control platform for our experiments for two different security protocols so that we can characterize the system behavior. The more detailed explanation of iSpace therefore follows in this section.

A. iSpace as a D-NCS testbed

iSpace is used in this paper as a networked-controlled integrated navigation system for a UGV. The destination point is chosen by the user via any remote computing interface in the world by accessing the iSpace GUI from the Internet as shown in Figure 2. The iSpace structure is explained in [4] and shown in the Figure 5.

![Figure 2: The graphical user interface for iSpace.](image)

The components are follows.

(i) An overhead network camera is used as a global sensor. The web camera collects the top-view image of the 2-D space of interest and sends it to the main controller for processing. The network camera used in the iSpace is the Axis network camera (Axis 206). The camera allows using CGI (common gateway interface) commands in order to control several capture properties under the web environment.

(ii) A differential drive UGV (Figure 3) is the navigator in iSpace named Betty. Eq(1) defines the motion of a differential drive UGV. The UGV receives commands from the main controller on the wireless channel to initiate the movement to reach the destination point specified by the user through the GUI. Betty carries a dell C800 Laptop with Linux OS (Figure 4).

\[
\begin{align*}
\dot{\theta}_r &= \frac{\sqrt{rW}}{2r} \\
\dot{\theta}_l &= \frac{\sqrt{rW}}{2r}
\end{align*}
\]

![Figure 3: Model of differential drive UGV.](image)

(iii) Main network controller has all the functional modules to operate the system. It also hosts the GUI. The main network controller (Linux Kernel 2.4.18-3 i686) is connected to the global sensors and the actuators in the space of interest through a wireless channel. The modules on the main controller are as follows.

- (i) A vision system to recognize the contents of the environment including obstacles, the UGV and their configurations. The obstacles and the UGV locations in iSpace are determined by processing the image captured by the network camera on top of the iSpace platform.
- (ii) A planning module for computing the optimal, obstacle-free, reference path for the UGV.
- (iii) A motion controller - Once the optimal path is generated, the UGV is commanded to follow this path to the destination by choosing reference points on the path at each loop interval \(\Delta t\). The whole control algorithm runs in a loop continuously until the UGV reaches its destination chosen by the user. This is explained by the flow diagram (Figure 6).

![Figure 4: The Betty UGV in iSpace](image)

![Figure 5: iSpace - prototype of the NCS used as a test-bed.](image)
UDG local controller: One of the major tasks of the local controller residing in the computing processor on the UGV is to receive linear speed (v) and angular speed (ω) information from the main controller via 802.11b wireless channels and drive the actuators (servo motors) for the UGV.

As the application can also be controlled over IP, we use TCP/IP or UDP/IP for the communication between the sensors, actuators and the main controller. TCP is slower than UDP because it also caters for the acknowledgement of the data received. In a time-sensitive application of iSpace, therefore, we send the sensor data with TCP protocol because the sensor data – image in case of iSpace – having all the needed information about the UGV workspace, will decide the course of processing and control commands generation by the main controller. However, the commands from the main controller to the actuator – the UGV in case of iSpace – are sent using the UDP protocol. The motion control commands are time sensitive. Once the interval time (Δt) is passed the old control command might have become obsolete and the new position of the UGV needs new motion commands to keep the UGV on path. It is one way communication. If the packet is dropped, the UGV will receive the next command packet corresponding to the next current position and act accordingly, all packets carrying the time sensitive information about the real-time motion of the UGV.

As the application can also be controlled over IP, we use TCP/IP or UDP/IP for the communication between the sensors, actuators and the main controller. TCP is slower than UDP because it also caters for the acknowledgement of the data received. In a time-sensitive application of iSpace, therefore, we send the sensor data with TCP protocol because the sensor data – image in case of iSpace – having all the needed information about the UGV workspace, will decide the course of processing and control commands generation by the main controller. However, the commands from the main controller to the actuator – the UGV in case of iSpace – are sent using the UDP protocol. The motion control commands are time sensitive. Once the interval time (Δt) is passed the old control command might have become obsolete and the new position of the UGV needs new motion commands to keep the UGV on path. It is one way communication. If the packet is dropped, the UGV will receive the next command packet corresponding to the next current position and act accordingly, all packets carrying the time sensitive information about the real-time motion of the UGV.

Encryption algorithms for wireless systems can be broadly classified into two categories: symmetric and asymmetric ciphers. Encryption algorithms, which require a single cryptographic key to encrypt and decrypt the data, are termed as symmetric ciphers. The Data Encryption Standard (DES) and the Advanced Encryption Standard (AES) are two symmetric ciphers which are considered secure for wireless systems [2][5]. However, the issue associated with symmetric ciphers is how to transmit cryptographic keys securely to the legitimate recipient. On the other hand, asymmetric ciphers are the ones which require different cryptographic keys for encryption and decryption of data. For example, public key encryptions, such as RSA, employ asymmetric ciphers. In public key encryption, each user has a pair of public and private keys. The public key is published while the private key is kept confidential. Messages are encrypted using the public key of the recipient, and can be decrypted only by the private key of the recipient. As private key is never known to others, this method eliminates the problem of transmitting secure keys to receivers. However, the critical issue associated with public key encryption is the high computational power required for encryption and decryption [1]. Thus public key encryption is used only for exchanging cryptographic keys securely in real time systems, whereas symmetric ciphers are used for actual data encryption and decryption.

In this work also, we implement DES and 3DES (a stronger variant of DES) for encrypting and decrypting the data securely over NCS. We do not consider AES in this work due to two reasons. First, AES requires wireless cards with new software drivers, which are not compatible with system software in UGVs. Moreover, due to limited processing and memory capabilities of UGVs in NCS, it was not feasible to have more number of security algorithms implemented in robotics node at the same time.

As discussed above, we use DES and 3DES encryption algorithms to implement confidentiality feature over NCS. Data Encryption Standard (DES) incorporates a 64-bit key to encrypt and decrypt the transmitted data. However, as the least significant bit of each byte in the key is a parity bit, that bit from each byte is ignored during actual encryption and decryption process, leading to reduction of 8 bits from 64-bit key. Therefore, DES uses 56-bit key effectively [2]. Besides, DES, being a block cipher, operates on 64-bit blocks of plaintext as input and outputs 64-bit blocks of encrypted blocks. 3DES is very similar to DES and requires three 64-bit keys to provide stronger security than DES. In 3DES, first each 64-bit block is encrypted by using first 64-bit key, then decrypted using second 64-bit key and then encrypted again by using third 64-bit key. For decryption at the receiver side, the entire process is reversed to obtain the original form. Consequently, 3DES is 3 times slower than DES [5][7]. DES incorporates different modes of operations such as ECB (Electronic Code Book), CBC (Cipher Block Chaining), CFB (Cipher Feedback) and OFB (Output Feedback). As ECB is considered the fastest mode of operation, it is used commonly in real time systems [7]. In this work, we use DES and 3DES with ECB mode to reduce computational processing.

III. FACTORIAL DESIGN FOR EXPERIMENT

First, we illustrate the different delays which have their effects on the functionality of the whole system. On a broader level, the delay classification can be done as
Transportation or Network delay component and Computational or Processing delay component. Their definitions and significance for a distributed NCS is explained as below.

<table>
<thead>
<tr>
<th><strong>Transportation / Network delay component ((\tau_{nc}^{(m)}))</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensor to controller delay ((\tau_{sc}^{ca}))</td>
</tr>
<tr>
<td>- (\tau_{k}^{imacacq}) - The time between the main controller requests the camera for the image</td>
</tr>
<tr>
<td>- (\tau_{k}^{R\cdot TT}) - Round trip network time.</td>
</tr>
<tr>
<td>m - Number of packets needed to send the complete image information.</td>
</tr>
<tr>
<td>n - Number of cameras to access to get the whole workspace image.</td>
</tr>
</tbody>
</table>

Controller to actuator delay (\(\tau_{ca}^{ca}\)) | \(\tau_{k}^{ca} = \tau_{k}^{R\cdot TT} / 2\)

<table>
<thead>
<tr>
<th><strong>Computational / Processing delay component ((\tau_{cp}^{(m)}))</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Non-real time computational delay ((\tau_{NrC}^{sc}))</td>
</tr>
<tr>
<td>- (\tau_{k}^{improc}) - Initial image processing time,</td>
</tr>
<tr>
<td>- (\tau_{k}^{pathgen}) - path generation.</td>
</tr>
<tr>
<td>- (\tau_{k}^{sysproc}) - system processing time</td>
</tr>
</tbody>
</table>

Real time Computational delay (\(\tau_{ReC}\)) | \(\tau_{k}^{ReC} = l \cdot \left(\tau_{k}^{improc} + \tau_{k}^{pathtrack} + \tau_{k}^{sysproc}\right)\) |
| l - Number of times the control loop runs before the UGV reaches the destination. |

<table>
<thead>
<tr>
<th><strong>Confidentiality delay</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>(\tau_{k}^{Encrypt}) Encryption of the command on the controller side.</td>
</tr>
<tr>
<td>(\tau_{k}^{Decrypt}) Decryption of the command on actuator side.</td>
</tr>
</tbody>
</table>

Therefore, stability and reliability are the main concerns for a D-NCS and under discussion in this paper. When a UGV is controlled over a network, its performance can be degraded by network-induced delays, and the system can even become unstable. Therefore the network delay compensation is necessary in a remote controlled, network based time sensitive application like iSpace. Similarly, because of the lack of security, the data can be maliciously used affecting the reliability of the operation. These concerns considered simultaneously will have different effects on the system described in the next section.

A. Defining the Factors

In this paper, as the distance between the UGV’s position and the reference path generated by the main controller is calculated at each \(\Delta t\), the UGV has to reach the destination as quickly as possible and avoiding the obstacles in the space. Two performance measure parameters are defined as follows.

\(Y_i\): \(\varepsilon_{Error}\) – Tracking error from the generated path
\(Y_2\): \(T_{Total}\) – Total time to reach the destination.

The following factors which will affect the iSpace system performance:

**NCS parameters**

(i) Network delay (\(N_{delay}\)) – \(N_{delay}\) is random in nature and can be represented by stochastic models.

(ii) Processing delay (\(P_{delay}\)) – \(P_{delay}\) depends upon the algorithm complexity.

(iii) System gain (\(\beta\))

(iv) Security in loop (\(U_{sec}\))

The \(N_{delay}\) factor on Internet is random in nature. Therefore to repeat the experimental environment and to have control over the factors affecting the systems, we setup the system as shown in Figure 7. Instead of wireless communication, the sensors to controller and the controller to the UGV are connected using the wiredhard connection. We connected the actuator to the controller with a delay generator in between. The delay generator is PC-based implementation to simulate network delay.

![Figure 7: Modified experimental setup to control the experiment factors.](attachment:image.png)

The delay generator receives the packets from the controller in a buffer format and it sends the packets to the UGV on the serial channel after applying a certain delay which is user defined. Considering negligible delay and no packet drops on the serial port, we can simulate the network delay \(\tau_{ca}^{ca}\) by giving different inputs of \(\Delta t\) to the network delay generator. The network camera is connected to the controller through Ethernet. The sensor to controller delay \(\tau_{sc}^{ca}\) in each loop is observed for around 40 experiment runs.

- Mean (\(\tau_{sc}^{ca}\)) = 0.2185 sec, Variance (\(\tau_{sc}^{ca}\)) = 2.5 e-4
- Mean (\(\tau_{ReC}\)) = 0.0223, Variance(\(\tau_{ReC}\)) = 1.5 e-4

Since the variances are small and insignificant in comparison with the mean, the sensor to controller delay as well as processing delay are considered to be constant for current operation of iSpace. The system gain and the security in the loop are taken as user inputs while running the experiment, therefore these factors are also controllable. Thus we are running the NCS experiment in a controlled environment where each significant factor affecting the system can be varied, repeatedly to study the security effect on the system performance.

**Application dependent factors**

In this paper, path tracking is the NCS application we are dealing with. Quadratic curve fitting path tracking controller is implemented as the path tracking controller. The basic principle of this control algorithm is that a reference point is moved along a desired path so that the length between the reference point and the UGV is kept in some distance (\(d_0\)). Control (velocity) commands - speed (\(v\), in cm/s) and turn rate (\(\omega\), in rad/s) - are generated for the UGV to reach that reference position from the current position. \(d_0\) is calculated by Eq(2). Therefore the factors which affect \(\varepsilon_{track}\) and \(T_{Total}\) the most are (i) Path curvature (\(\kappa\)) and (ii) Maximum lookahead distance for the path tracking (\(S_{max}\)).

\[
d_i = \frac{S_{max}}{1+\beta / k} \tag{2}
\]

As \(S_{max}\) and \(\kappa\) are path tracking parameters \([4]\), they are application dependent factors, which might be different in case of different NCS application. But all other factors are common for all NCS applications. Therefore to study the effect of network delay, security features in any common
NCS, we fix the values for the application dependent factors i.e. $S_{\text{max}}$ and $\kappa$ for the current system operation. We define a fixed path with predefined curvatures for all experiments as shown in Figure 8. The path consists of pieces of different path curvatures.

![Path to be tracked](image)

Figure 8: Path P to be tracked with curvature pieces as {0, 2, 4, 8, 0} m$^{-1}$

To find a suitable value of $S_{\text{max}}$ for the current operation of iSpace, we first define different levels of the $S_{\text{max}}$ for path tracking operation in iSpace. To decide the range of $S_{\text{max}}$ we intuitively decide the lower and higher value suitable for $S_{\text{max}}$. The look-ahead distance should be ideally chosen such that the interpolated path between the current and the reference point matches the actual path as close as possible. In the predefined path $P$, $\kappa_{\text{max}} = 4$ m$^{-1}$ as $r_{\text{max}} = 0.25$m. We have half circle with $\kappa_{\text{max}}$ in P. We assume that for practical purposes $S_{\text{max}}$ should not be greater than quarter the perimeter i.e. $0.5\pi r_{\text{max}}$ (as shown in Figure 8). Therefore,

$$\max(S_{\text{max}}) = 0.125\pi r = 0.3927 \approx 0.4 \text{ m}.$$  

We therefore vary $S_{\text{max}}$ through 5 levels {0.05, 0.1, 0.2, 0.3, 0.4}. The path tracking experiment is run repeatedly 5 times with each value of $S_{\text{max}}$ with other factors kept constant at the values $N_{\text{delay}} = 0$ sec, $U_{\text{sec}} = 600$ ms. After analyzing the $e_{\text{track}}$ and $T_{\text{total}}$ we decide that $S_{\text{max}} = 0.2$ m is a suitable value for tracking of the path P (Figure 9).

![Distance Error Vs Smax](image)

Figure 9: Mean of $T_{\text{total}}$ and $e_{\text{track}}$ Vs $S_{\text{max}}$ over 5 experiment runs. $S_{\text{max}} = 0.2$ m is suitable looking at the lowest $T_{\text{total}}$ and comparatively low $e_{\text{track}}$.

### B. 2$^k$ factorial design

After choosing suitable value for application dependent factors and setting up the experiment so that other NCS factors are controllable, we use statistical experiment design approach to estimate the effect of factors on the NCS performance. We have $k=3$ factors which are controllable. Let us design the $2^k$ factorial design to estimate the effect of these factors’ values on NCS performance measure. For $2^k$ factorial design, we have to define high and low level for all the factors [8]. That is done as follows:

1. $N_{\text{delay}}$ (Factor A) – Typical loop network delay observed when the iSpace is operated over wireless channel varies from 0.2 to 0.3 seconds. Generally speaking, the network delay is in the order of few hundreds of millisecond on the Internet. Typical delays observed in US continent are 0.1 s to 0.6 s. Therefore we define lower limit of $N_{\text{delay}}$ = 20 ms and higher limit of $N_{\text{delay}}$ = 600 ms.

2. System gain ($\beta$) (Factor B): $\beta = 1$ is considered to be the standard or typical value for the system gain. We take lower and higher limit for $\beta$ to be 0.5 and 1.5 respectively. (Symmetric about $\beta=1$).

3. Security in loop ($U_{\text{sec}}$) (Factor C): As mentioned before, we take into consideration 3 types of security features. (i) No security (ii) DES encryption (iii) 3DES encryption. Therefore we have two sets of factorial design experiments. (1) No security in comparison with DES and (2) No security in comparison with 3DES.

The experiments are replicated twice $n=2$. For detailed explanation of $2^k$ factorial design, you can refer to [8].

**Null Hypothesis:**
We define $H_0$: Security feature addition does not have significant main effect on NCS performance given 95% confidence interval. i.e. Effect($C$)$\times$e.e. /Effect$_{bysys}$ includes zero.

### IV. Observation and Analysis

In this section, we estimate the effect on $e_{\text{track}}$ and $T_{\text{total}}$ due to the three factors separately as well as the interaction between the factors if any. As $k=3$ therefore we will run the experiment for $2^3 = 8$. (1) $a$, $b$, $ab$, $c$, $ac$, $bc$, $abc$ represent the sum of $n$ replicates of 8 treatment - (1) is the treatment with all factors at their low levels, a is the treatment with factor A at high and factors B and C low level and so on (Eq(3)).

\[
(1) = \sum_{i=1}^{n} y_{ABC}, \text{ where } A=20 \text{ms}, B=0.5, C=\text{No security}
\]

\[
a = \sum_{i=1}^{n} y_{ABC}, \text{ where } A=600 \text{ms}, B=0.5, C=\text{No security}
\]

\[
ab = \sum_{i=1}^{n} y_{AB}, \text{ where } A=600 \text{ms}, B=1.5, C=\text{3DES}
\]

\[
abc = \sum_{i=1}^{n} y_{ABC}, \text{ where } A=600 \text{ms}, B=1.5, C=\text{3DES}
\]

Using the totals under the treatment combination shown in tables above, we estimate the factor effects as follows, given in [8]:

\[
A = \frac{1}{4n} \left( a - (1) + ab - b + ac - c - abc - bc \right)
\]

\[
B = \frac{1}{4n} \left( b + bc + ab + abc - (1) - c - a - ac \right)
\]

\[
C = \frac{1}{4n} \left( c + ac + bc + abc - (1) - a - b - ab \right)
\]

Using the totals under the treatment combination shown in tables above, we estimate the factor effects as follows, given in [8]:

\[
A = \frac{1}{4n} \left( a - (1) + ab - b + ac - c - abc - bc \right)
\]

\[
B = \frac{1}{4n} \left( b + bc + ab + abc - (1) - c - a - ac \right)
\]

\[
C = \frac{1}{4n} \left( c + ac + bc + abc - (1) - a - b - ab \right)
\]
ABC = 1/4n \sum_{abc} \left( \sum_{i} y_{ij} - \mu \right)^2 / \sigma^2 \quad \text{(4)}

We also find the standard error of effect estimates. \( S^2 \) is the sample variance estimate, \( \sigma \) is the sample mean of the \( n \) observations. \( s.e.(\text{Effect}) \) is the standard error of each estimated effect. (Eq 4)

\[
\sigma^2 = \frac{\sum_{i=1}^{n} y_{ij}^2}{n} - \frac{1}{n^2} \sum_{i=1}^{n} \left( \sum_{j=1}^{n} y_{ij} - \mu \right)^2
\]

Table 2: Main effect estimate on system performance with \( N_{\text{delay}} \), System gain and Security addition.

<table>
<thead>
<tr>
<th></th>
<th>DES_\text{track}</th>
<th>3DES_\text{track}</th>
<th>DES_T\text{total}</th>
<th>3DES_T\text{total}</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0.0145</td>
<td>0.0117</td>
<td>32.73</td>
<td>33.2413</td>
</tr>
<tr>
<td>B</td>
<td>0.0100</td>
<td>0.0125</td>
<td>-30.422</td>
<td>-31.2209</td>
</tr>
<tr>
<td>C</td>
<td>0.0109</td>
<td>0.0075</td>
<td>-1.4606</td>
<td>-2.3288</td>
</tr>
<tr>
<td>AB</td>
<td>0.0126</td>
<td>0.0111</td>
<td>-2.463</td>
<td>-4.9491</td>
</tr>
<tr>
<td>AC</td>
<td>0.0079</td>
<td>0.0072</td>
<td>-10.664</td>
<td>-9.3109</td>
</tr>
<tr>
<td>BC</td>
<td>0.0088</td>
<td>0.0113</td>
<td>0.4847</td>
<td>-0.3134</td>
</tr>
<tr>
<td>ABC</td>
<td>0.0112</td>
<td>0.0096</td>
<td>3.979</td>
<td>1.4934</td>
</tr>
<tr>
<td>( S^2 )</td>
<td>4.375e-7</td>
<td>2.087e-6</td>
<td>2.5736</td>
<td>1.8236</td>
</tr>
<tr>
<td>s.e.(Effect)</td>
<td>3.307e-4</td>
<td>7.223e-4</td>
<td>0.8021</td>
<td>0.6752</td>
</tr>
</tbody>
</table>

Figure 10: Comparison of performance parameters with respect to security addition and network delay for \( \beta = 1 \).

V. CONCLUSION AND FUTURE WORK

Through the experiments on the iSpace platform – which demonstrates a distributed networked-controlled-system – we characterized a data-sensitive wireless distributed NCS, discussed its properties based on data-sensitivity, network delay effect and the requirement for the network security. This paper puts forth a point that statically adding security algorithms in a time sensitive NCS would degrade the performance of the system to some extent, which is critical considering the stability of the NCS. Using standard statistical characterization approach we also observed that interaction between security features and network delay and system gain is also an important factor towards NCS performance. This trade-off between the security addition and real-time performance of the system can be solved by dynamically optimizing the part of the security in the system loop. Thus, future work will involve investigating the dynamic data protection method in addition to finding an optimum solution between the reliability offered by network security, stability and time-sensitivity required for an NCS. Dynamically choosing the data security algorithms will also be a function of data priority, bandwidth requirement and time sensitivity of NCS. Therefore the characterization of any data-sensitive wireless NCS over different parameters is a key.

ACKNOWLEDGEMENT

The authors acknowledge the National Science Foundation (NSF) for sponsoring this research (Grant No. IIS-0426852). The opinions expressed are those of the authors and do not necessarily reflect the views of the NSF.

REFERENCES