Optimal Control of DC Motor via Ethernet Using Ant Colony Optimization (ACO)

Sochima V. Egoigwe[1], Oghenevwaire Akpeghagha[2] and Chukwudi Chukwudozie[3]
[1] *Department of Mechatronic Engineering, University of Nigeria, Nsukka, Nigeria*
[2,3] *Department of Electronic Engineering, University of Nigeria, Nsukka, Nigeria*

Abstract- Internet of Things (IoT) technologies can be connected via wired or wireless technologies, nevertheless several wireless devices still make use of wired technological infrastructure. When it comes to accessing cloud services, Ethernet is the dominant wired network technology with expanding capabilities. Ethernet and Transmission Control Protocol/Internet Protocol (TCP/IP) communications are however not deterministic, so communication response time between control devices is variable. This paper proposes the design and implementation of ant colony optimization (ACO) algorithm for the control of a DC Servo via an Ethernet network. The paper demonstrates how the ant colony optimization can be used to improve the stability and control performance of the Ethernet network in control systems. The ant colony optimization algorithm is used for tuning the PID controller and finding its optimal parameters. The Truetime simulator and the Matlab/Simulink software was used for the design and simulation of the model. The proposed system was compared and evaluated against the existing traditional PID control system. The result reveals the potentials of the ACO algorithm in reducing the effect of delay introduced by the Ethernet network in the control system and the feasibility of the Ethernet for control applications.

Keywords: Internet of Things, Ethernet, Control, DC Motor, ACO.

I. INTRODUCTION

Lately, research trend has been on control through LAN/Ethernet network, which guarantees faster speed and easy configuration. The IEEE 802.3 Ethernet standards provide a low cost option for industrial applications with high level of reliability. Even though several IoT applications makes use of wireless networks, such as wearables. Ethernet has a lot of options that could provide IT professionals with the perfect-wired connectivity. Ethernet provides the possibility of combining power and data, which provides interconnect options that are not possible with wireless networks. Several IoT applications require lower data rate connectivity. Wired Ethernet can be used to guarantee the necessary throughput. Different ways industries use Ethernet for IoT applications include [1]:

1. Power over Ethernet (PoE) as at now powers wireless access points and telephones that uses IP. Nevertheless, it is gradually being incorporated into IoT devices, such as security cameras which requires power as well as data. PoE is also essential to wireless access points and supporting Wi-Fi technologies and will be very useful with smart LED lighting.
2. The recently developed IEEE 802.3cg standard transmits over a single twisted pair cable and also enables higher speed connection in process control networks, without the aid of gateways for actuators, sensors and field switches. The IEEE 802.3cg standard can be employed by organizations in equipment to ensure the intrinsic safety demand, which is a necessity for hazardous areas.
3. IoT devices which makes use of the Wi-Fi technology to link wireless access points though a wired infrastructure, based on physical layers which supports the operation of the Ethernet network over UTP cabling.

4. Since the last decade, series of Ethernet standards have been developed that will enable data connectivity within the range of 10 Mbps to 10 Gbps for deployment inside vehicles on a single twisted pair of conductors.

Ethernet connections are less susceptible to dropped connections, have more reliability and do not require constant debugging. Local factors such as interference from other electronic devices, walls, cabinets, floors, length of room etc. doesn't easily affect Ethernet connections [1].

II. RELATED WORKS

Devan et al. presents some of the benefits and risks of combining high-level distributed control system communications with low-level "field-bus" communications on the same network, and illustrates measures taken at SNS to improve compatibility between devices connected to the control system network. Nevertheless new initiatives may have to be employed to keep network traffic low to an acceptable level as the network grows in increases [2]. The authors in [3], experimented on the control of a networked brushless DC motor using the NetCon. From the experiment it was shown that the NetCon is reliable and both valid and can be trusted. Yong-Wang et al, [4] proposed a novel speed approach for the elevator driving system, which includes a speed loop and a current loop. The approach made use of the fuzzy logic controller with its parameters tuned using the ant colony optimization algorithm. The result gotten from the simulation demonstrated the feasibility of the proposed control approach. Prathyusha et al, [5], proposes the ant colony optimization algorithm for speed control of DC motor. The authors postulated that Ants could dynamically modulate next junction that can be referred in construction process, therefore it will guarantee accuracy and feasibility of solution.

Considering its exceptional speed control feature, the DC motor is widely used in factories and industrial applications even though it is quite expensive to maintain compared to the induction motor. This paper proposes the use of ant colony optimization (ACO) algorithm for the control of a DC Servo using an Ethernet connection. The ant colony optimization was used to improve the stability and control performance and as well reduce the effect of the non-deterministic characteristic of the Ethernet network for IoT applications. The paper evaluates the behaviour of a DC motor when it is controlled via an Ethernet connection. The Ethernet packet was discussed and also two major Ethernet protocols for control were discussed as well. The ant colony optimization algorithm was also discussed in this research. The model used for the simulation was given and designed using the Matlab/Simulink software and the Truetime toolbox. The results from the simulation were presented and discussed accordingly.

III. **THE ETHERNET/IEEE 802.3 PACKET**

Both the Ethernet and the IEEE 802.3 adopts similar network technology. They can both be classified as broadcast networks and also CSMA/CD Local Area Networks. Several distinct packet types may pass through an Ethernet network all will either make use of the Ethernet packet format or the IEEE 802.3 packet format. Fig. 1 below illustrates the format for both the IEEE 802.3 and Ethernet packet [1][6].

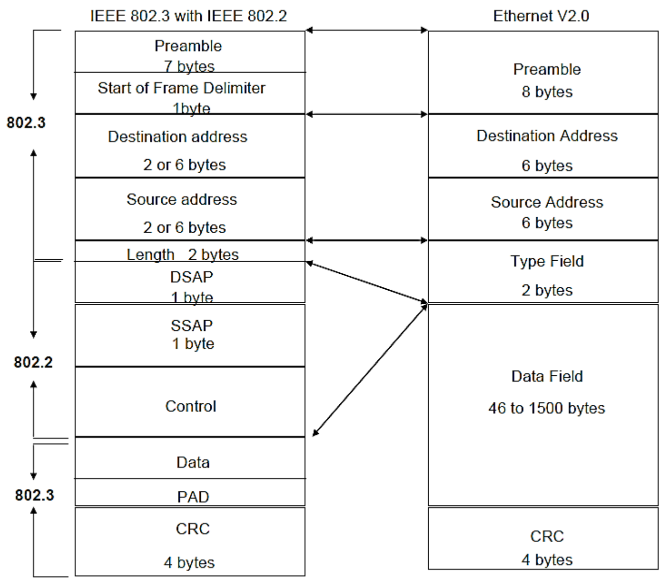

Fig 1. The Ethernet and IEEE 802.3 Packets

IV. ETHERNET OPTIONS FOR CONTROL SYSTEMS

Ethernet has two common protocols, which are the Transmission Control Protocol (TCP/IP), and User Datagram Protocol (UDP/IP). TCP/IP is most popular and it guarantees the arrival of data packets in the right order, which is a necessity for communications such as music and video streaming. It also makes sure that packets are not lost and provides congestion control, which is essential for open networks such as the Internet. Control systems using closed networks may lose packets as a result of an engineering fault such as electrical noise. For closed loop control systems, congestion is not a characteristic problem because the engineer has full control over the traffic in the network. The TCP/IP is very suitable for networks that are large and closed looped such as applications with heavy traffic, several unaffiliated devices and long cable lengths.

The UDP/IP is simpler than the TCP/IP; it doesn't require a connection to be established so there exist little handshaking between the slaves and the master. Also the device, which is transmitting, does not require an acknowledgement receipt from the receiver, hence reducing the packet count. This brings about high throughput and quick error recovery. Nevertheless, in the case when an error occurs, the transmitting device may not automatically recognize it. The UDP/IP is appropriate only for closed loop control systems. Fig. 2 illustrates the TCP/IP and UDP/IP Encapsulation [1, 6].

V. THE ANT COLONY OPTIMIZATION ALGORITHM

In the proposed system, Ant Colony Algorithm is used. The ACO is a new evolution algorithm, which can be used in finding optimal parameter for the PID controller. It searches for an optimal solution via the process of evolution of the group combined by candidate solution. The ACO algorithm uses positive feedback technique in order to perform the task of intelligent searching and global optimization; and at the same time possessing strong robustness. The ant colony optimization algorithm (ACO) is a probabilistic technique or mechanism for solving highly computational problems, which can be reduced to find good paths through graphs [5, 7].

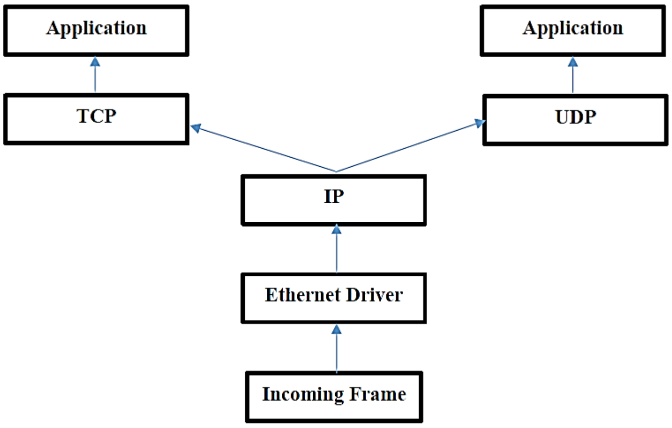

Fig. 2: TCP/IP and UDP/IP Encapsulation

The basic idea behind the ACO involves the movement of an ant colony across different paths or states of the problem. This movement is majorly influenced by two potential decisions, which are attractiveness and trail. Thus, each ant constructively proffers a solution to the problem. When a solution is completed by an ant, it analysis the solution and modifies the value of the trail on the components that are in its solution. This information known as pheromone information will in turn direct the search of the next ants [4, 7]. Fig. 3 illustrates the ACO algorithm using a flow chart.

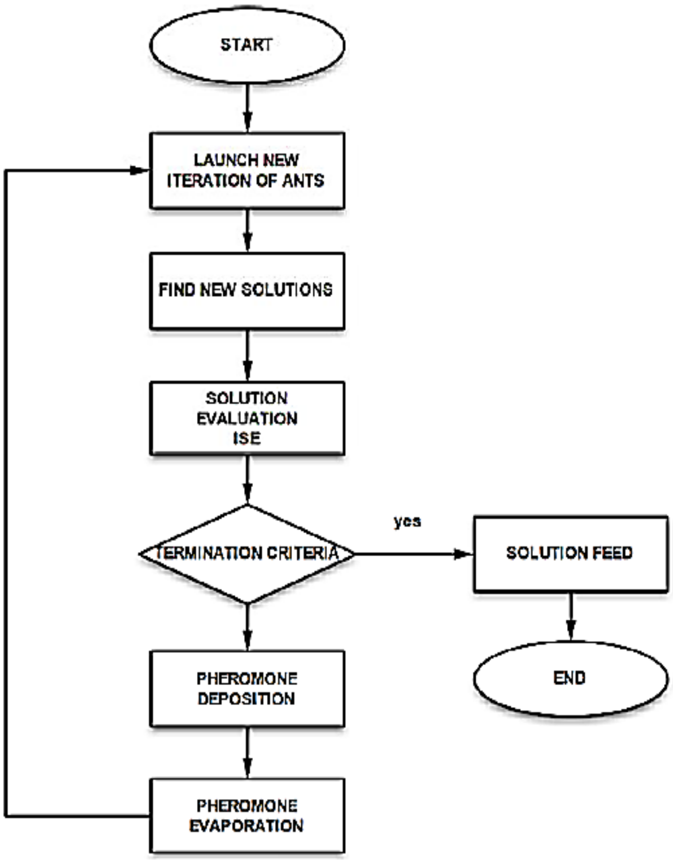

Figure 3: Flow Chart of ACO Algorithm

The parameters used for the simulation are shown below using Matlab code function:

```
%% ACO paramters
n_iter=1; %number of iteration
NA=3; % Number of Ants
alpha=0.8; % alpha
beta=0.2; % beta
roh=0.7; % Evaporation rate
n_param=3; % Number of paramters
LB=[0.1 0.5 0.1]; % lower bound
UB=[5 10 0.8]; % upper bound
n_node=1000; % number of nodes for each param
```

The Matlab function code below illustrate how the nodes were generated:

```
%% Generating Nodes
T=ones(n_node,n_param).*eps; % Phormone Matrix
dT=zeros(n_node,n_param); % Change of Phormone
for i=1:n_param
Nodes(:,i) =linspace(LB(i),UB(i),n_node); % Node generation at equal spaced points
end
```

The iteration loop was implemented using the Matlab function code shown below:

```
%% Iteration loop
for iter=1:n_iter

  for tour_i=1:n_param
    prob(:,tour_i)= (T(:,tour_i).^alpha) .* ((1./Nodes(:,tour_i)).^beta);
    prob(:,tour_i)=prob(:,tour_i)./sum(prob(:,tour_i));
  end

  for A=1:NA
    for tour_i=1:n_param
      node_sel=rand;
      node_ind=1;
      prob_sum=0;
      for j=1:n_node
        prob_sum=prob_sum+prob(j,tour_i);
        if prob_sum>=node_sel
          node_ind=j;
          break
        end
      end
      ant(A,tour_i)=node_ind;
      tour_selected_param(tour_i) = Nodes(node_ind, tour_i);
    end

    cost(A)=cost_func(tour_selected_param,0);
    clc
    disp(['Ant number: ' num2str(A)])
    disp(['Ant Cost: ' num2str(cost(A))])
    disp(['Ant Paramters: ' num2str(tour_selected_param)])
    if iter~=1
    disp(['iteration: ' num2str(iter)])
    disp('_________________________')
    disp(['Best cost: ' num2str(cost_best)])
    for i=1:n_param
      tour_selected_param(i) = Nodes(ant(cost_best_ind,i), i);
    end
    disp(['Best paramters: ' num2str(tour_selected_param)])
    end

  end
```

```
[cost_best,cost_best_ind]=min(cost);

% Elitsem
if (cost_best>cost_best_prev) && (iter~=1)
   [cost_worst,cost_worst_ind]=max(cost);
   ant(cost_worst_ind,:)=best_prev_ant;
   cost_best=cost_best_prev;
   cost_best_ind=cost_worst_ind;
else
   cost_best_prev=cost_best;
   best_prev_ant=ant(cost_best_ind,:)
end

dT=zeros(n_node,n_param); % Change of Phormone
for tour_i=1:n_param
   for A=1:NA
      dT(ant(A,tour_i),tour_i)=dT(ant(A,tour_i),tour_i)+cost_best/cost(A);
   end
end
```

VI. DESIGN OF THE CONTROL SYSTEM USING TRUETIME

The block model of the proposed design is shown in Fig. 4 below. The delay in the system is represented with τ. The total delay in the system is given by:

$$\tau = \tau_{sc} + \tau_{ca} + \tau_{s} + \tau_{c} + \tau_{a} \qquad (1)$$

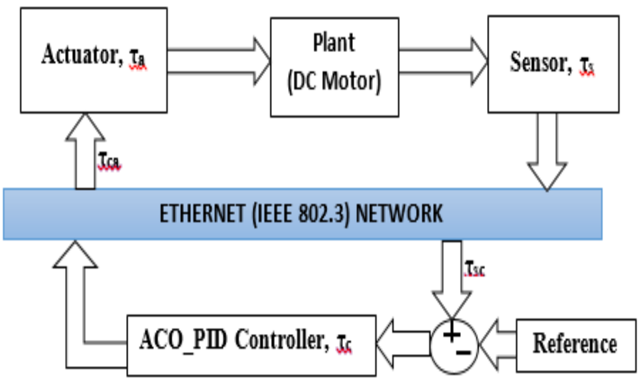

Fig. 4. Block Model of Proposed System

The delay in the control system can further be divided into two types: Computational delay (τ_{sc} *and* τ_{ca}) and Process delay (τ_s, τ_a, τ_c). This research evaluates the influence of the total delay on the system output. This shows the control performance of the Ethernet when used in the control of a DC Motor using the proposed model.
The proposed model was designed and simulated using the Matlab/Simulink Software and Truetime toolbox. The model was designed to simulate the control of a DC motor via an Ethernet network interface. The model implements the ant colony optimization algorithm for tuning the parameters of the PID controller used in the control system.

The transfer function of the DC motor is given by:

$$H(s) = \frac{1000}{s(s+1)} \qquad (2)$$

The simulation was done against a reference signal (a unit step function). The proposed model attempts to control the output of the DC motor to behave as a unit step function, which is the reference signal over the Ethernet network. Fig. 5 below shows the design of the proposed model.

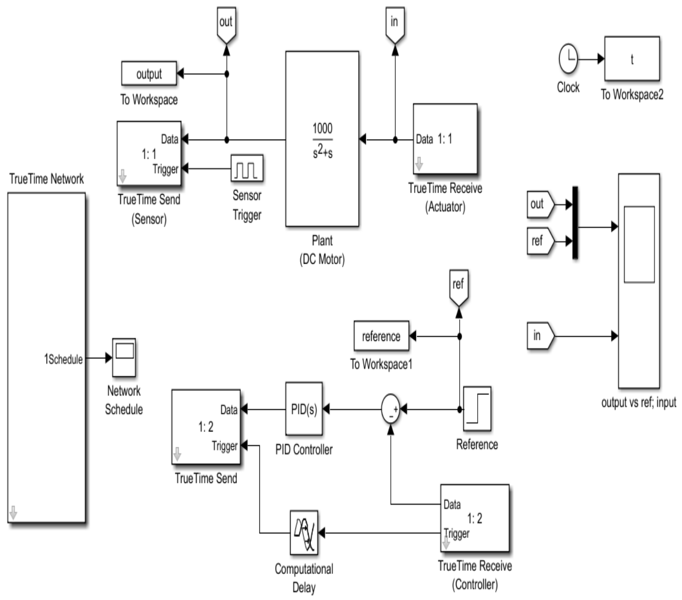

Fig. 5. Proposed Model in Matlab/Simulink

VII. ANALYSIS OF RESULTS

We implemented the ACO algorithm discussed in section V above using the Matlab//Simulink software and the result is shown in fig. 7 below. Fig. 6 displays the result for a control system interconnected via the Ethernet network using he traditional PID controller.

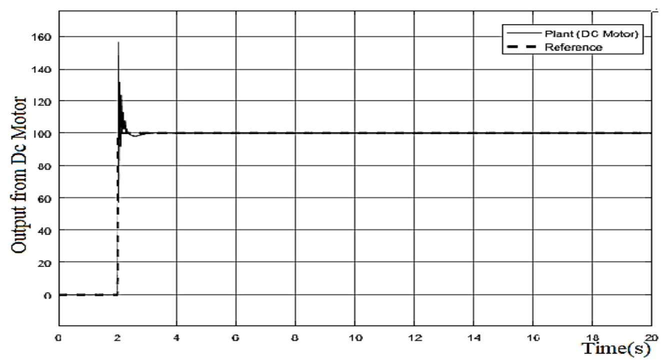

Fig. 6. Existing System (PID control of DC motor)

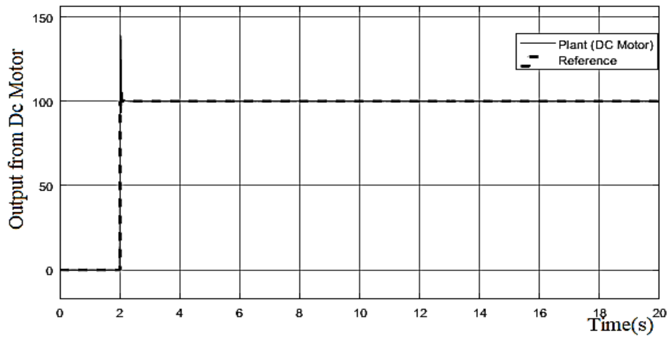

Fig. 7. Proposed ACO-PID Control of DC Motor

VIII. DISCUSSION

Fig. 6 shows the control output of a DC motor performance for a DC motor control using a manually tuned PID controller control system setup.

It can be seen from fig. 7 that the use of the ACO-PID attempts to nullify the effect of delay in the control system when it is connected. The table below further analysis the delay in both systems with an analysis of their settling time, rise time, peak overshoot and undershoot.

Table 1. Result Analysis

Parameters	**Existing System (PID controller)**	**Proposed System (ACO-PID controller)**
Peak overshoot	56.5 units	39.3 units
Undershoot	1.54 units	0.0950 units
Settling time	1.033 seconds	0.353 seconds
Rise time	0.001 seconds	0 seconds

Table 1 gives an analytical summary of the control output from the simulation. The difference of the peak shoot of the existing system to the proposed system was 17.2 units and for the peak undershoot it was 1.445 units. The Settling time of the existing system had a difference of 0.68 seconds to that of the proposed system. The rise time of the existing system to that of the proposed system was just 0.001 seconds.

These results demonstrate the effectiveness of the ant colony optimization has great potentials in mitigating the influence of delay introduced by the Ethernet network when used in a control system setup. This result will aid the application of Ethernet network in future control applications.

IX. CONCLUSION

This research was aimed at evaluating the feasibility of employing the Ethernet for control applications. From the result analysis, it can be seen that with the use of the Ant Colony Optimization algorithm the effect of the non-deterministic characteristics of the Ethernet network on control performance can be reduced. This result reveals the potentials of the ACO algorithm in reducing the effect of delay in Ethernet and the feasibility of the Ethernet for control applications. Though the system had a level of disorder due to the influence of the Ethernet network. Future research will focus on mitigating this disorder to the barest minimum.

REFERENCES

[1] Galil, (2014); Evolution of Ethernet in Control Systems: Manufacturing Automation Handbook; Annex North. Vol 1

[2] W. Devan et al, (2005); "Using a Control System Ethernet Network as a Field Bus"; Proceedings of 2005 Particle Accelerator Conference; pp. 3961-3963.

[3] H. Bian G.P Liu, Z. Dong,; "Structure Design and Application of Embedded Ethernet Based Control Systems"; Proceedings of the 2007 IEEE International Conference on Networking, Sensing and Control; London, UK; 20017; pp. 15-17

[4] F. Yong-Wang, F. Hong-Wei and X. Mu-Feng,; "Control of Brushless DC Motor with Ant Colony Optimization"; International Conf. on Manufacturing Sci. & Eng.; 2015; pp. 289-292

[5] P. Prathyusha, T. P. Babin, V. Suresh; "Speed Control of DC Motor Using Ant Colony Optimization Algorithm"; International Journal of Advanced Research in Biology Engineering Science and Technology"; India; Vol. 2, Issue 6; 2016; pp. 112-117

[6] C. Adrian; "The Extension: A Technical Supplement to Control Network"; Fiedlbus. pub.; Fall; Vol. 1, Issue 3;1999; pp. 1-6.

[7] C. L. Xia, H. W. Fang, W. Chen, and et al,; "Ant colony algorithm based fuzzy control for a brushless DC motor", The SixthWorld Congress on Intelligent Control and Automation; Dalian; 2006; pp. 6498–6502.