Turkish Journal of Electrical Engineering and Computer Sciences

Volume 21 | Number 1 Article 16

1-1-2013

Control in networked systems with fuzzy logic

AKHAN AKBULUT

OĞUZHAN ÖZTAŞ

Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2013) 21: 225 – 233

doi:10.3906/elk-1107-24

Control in networked systems with fuzzy logic

Akhan AKBULUT[1], **Oğuzhan ÖZTAŞ**[2*]
[1]Department of Computer Engineering, Faculty of Engineering and Architecture,
İstanbul Kültür University, 34156, İstanbul, Turkey
[2]Department of Computer Engineering, Faculty of Engineering, İstanbul University,
34320, İstanbul, Turkey

Received: 13.07.2011 • **Accepted:** 16.10.2011 • **Published Online:** 27.12.2012 • **Printed:** 28.01.2013

Abstract: Recently, the development of control systems based on network-based architecture is getting very high attention. Because of its fast data communication, network-based control is in high demand. However, there are some disadvantages, such as delays, data packet dropouts, and communication constraints. Network-based control systems are getting attention in the development of this architecture because of these disadvantages. Optimization of the system is supplied to improve the existing structure. The optimization is categorized into 2 main structures: software optimization and hardware optimization. The structure of the overall system is designed with these optimization strategies. A hierarchical and communicational structure has emerged as a result of constructing the overall structure.

After the optimization and design processes, the emerged system is simulated to implement and test the structure. Sensors and actuators are implemented as programs in different computers. Controllers are set into different computers and an observer program is set into another computer to observe the overall system. The components of the system are set in a sequential structure and give the expected performance due to their environment.

Key words: Networked control systems, data packet, sensor, controller, fuzzy, agent-based systems

1. Introduction

To implement a real-time control system, a project is designed for process control in a network-based architecture. In this project, 3 sensors, 2 actuators, 2 controllers, and an observer program are used to form the overall system. The aim of this project is processing the incoming material and packing the processed material. The final process of the system is labeling the processed material. The sensor program gets the information and sends a request to the corresponding controller. The controller takes the request and sends a response to the sensor. After the handshake of the sensor and the controller, the controller sends the related information to the corresponding actuator. The actuator processes the information and the next sensor takes the processed information. This processing continues until the final sensor produces the desired information. During these processes, an observer program observes the overall system, analyzes the performance, and reports the events. The system is improved using conceptual methods to reduce the network-induced delays and stabilize the system functionality.

This paper is organized into 8 sections: the introduction is given in Section 1; in Section 2, a time-line of the control systems is presented; reasons for the development of the control mechanisms are introduced in Section 3; in Section 4, the analysis stage is introduced; the optimizing strategies are shown in Section 5;

*Correspondence: oguzhan@istanbul.edu.tr

the structure of the system is presented in Section 6; after the design phase is shown, implementation of the simulated system is presented in Section 7; and, finally, the conclusion is presented in Section 8.

2. Time-line of control

J.C. Maxwell's 1868 publications on the steam engine regulation that emerged in the 19th century were the first steps in the design principles of specific feedback systems. With these developments, Nyquist's 1932 paper supplied principles that could be theoretically applied to any feedback system. A foundation for frequency domain methods was supplied by the work of Nyquist, Bode, Nichols, and Evans between 1930 and 1950 [1]. After 1950, digital computers started to implement control systems. The continuous-time state is applied to the discrete-time state.

In the past, information was brought by system components via hard-wired connections from a sensor to a centralized location for monitoring and making decisions for the control. Next, the actuators implement the control policies. Today, processing at remote locations is available via a network. Bosch GmbH began a study using networked devices to control passenger cars in 1983 [1]. In February 1986, the communication protocol of the control area network (CAN) was announced at the Congress of the Society of Automotive Engineers, in Detroit, MI, USA. CAN hardware with Intel's 82526 chip was introduced in 1987 [1]. The evolution process is illustrated in Figure 1.

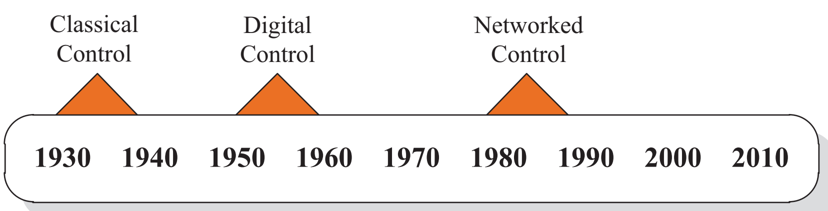

Figure 1. Timeline of the technological evolution, from the classical feedback control to the digital control to the networked control.

3. Reasons for development

3.1. Basic issues in networked control systems

Network-induced delay is the basic issue in this concept. The total delay is calculated by a delay in each part of the system. Another issue is data packet dropouts. This problem occurs because of the software that is used for the control mechanism. The processing mode (time-driven and event-driven) of the control mechanism states the behavior of the system [2]. If the processing mode is time-driven, the system runs in a periodic time range to execute its tasks. If the processing mode is event-driven, the system waits for the previous task to end before starting the next task. In the first mode, a task cannot be completed by a system component because of delays or data packet dropouts. In the second mode, the system may wait for a long time for the previous task to end. In both cases, there are some problems that make the system unstable.

3.2. Advantages and disadvantages of networked control systems

There are advantages such as the decentralization of control and productivity, and disadvantages such as delays of the networked control system and data packet dropouts, which are shown in below.

3.2.1. Advantages of networked control systems

Networked control systems have many advantages, such as low costs, decentralization of control, simple installation, quick and easy maintenance, reduced weight, and modularity. Network control systems support many topologies such as bus topology, star topology, and tree topology. Each component of the networked control system has individual functions for a global goal by communicating and transmitting information via a network. For large-scale, complex systems, integrating the networked communication structures guarantees the transmission of real-time control information in the total system. They are faster and cheaper than most proprietary systems. PC-to-Ethernet costs from US$20 to $50, but a proprietary system board costs from $300 to $900. Ethernet connectivity is available at 100 Mb/s, and there are studies to improve the data rates from 1 Gb/s to 10 Gb/s. The faster proprietary systems are generally available at 12 Mb/s [3].

3.2.2. Disadvantages of networked control systems

Information transmission delays may occur because of communication constraints. Delays may be fixed or random. When random delays emerge, the analysis and design of networked control systems can become very complex [2]. Because of delays, the system loses determinability, which means that the time of the arriving data is not determinable. The system loses integrability because the data may produce loss or create a mistake during the transmission process. The system loses the cause and effect because of the uncertainty of the network transmission time. Timing is a very important factor. Frames that are transmitted in the communication channel must be processed fast enough in order to not lose frames due to the controller's latency [4]. The system loses constancy because of the data arriving time. If the communication channel is noisy, randomly occurring bit errors decrease the system's performance [1]. If the bit errors are frequent, the system's performance is reduced and the system loses its functionality.

4. Analysis

4.1. Delay analysis

Network-induced delays are the sum of the delays, such as medium access delay, transmission delay, and congestion delay. Delay values depend on the network architectures, protocols, and operating conditions [2].

- τ_s is the latency delay of the sensor for network transmission.
- τ_{sc}^k is the transmission delay in the communication channel from the sensor to the controller.
- τ_{sq} is the latency delay of the information being sampled by the controller.

When the controller is in the event-driven mode:

$$\tau_{sq} = 0.$$

- τ_c is the computation delay of control algorithms due to unpacking, decoding, and execution of the feedback information, and coding and packing of the decision-making information.
- τ_{cq} is the latency delay of the decision-making information to queue up the network transmission.
- τ_{ca}^k is the transmission delay in the communication channel from the controller to the actuator.

- τ_a is the latency delay of the information being adopted by the actuator.

When the actuator is in the event-driven mode:

$$\tau_a = 0;$$

hence, the total delay is:

$$\tau = \tau_s + \tau_{sc}^k + \tau_{sq} + \tau_c + \tau_{cq} + \tau_{ca}^k + \tau_a. \tag{1}$$

4.2. Data packet dropouts

In communication between the components, there may be data packet dropouts because of transmission errors. Thus, the system transmits incomplete data or communicates with missing data. This causes incorrect processing and the system may be unstable because of this problem. This should be addressed by the software that is used by the system components in the control.

Using protocols to code stable communication blocks is a solution. Thus, data packets are guaranteed by some controls, and the receiver sends the information that the data is received and the transmitter gets the information that the data is transmitted. Applying this control in a noiseless communication channel gives good results.

5. Optimization

5.1. Software optimization

There are 2 modes for processing the data in the network by system components. The first is the time-driven mode and the second is the event-driven mode [2]. For optimization, the event-driven mode can be used to protect data integrity within the communication in the system. To set the time range of the data processing, the time-driven mode can be used. Therefore, using the hybrid mode makes the system more flexible. In networked control systems, network-induced delays and packet losses are 2 main factors that reduce the system stability and dynamic performance and decrease the functionality of the system [5]. To stabilize networked control systems and improve their dynamic performance, a dynamic feedback controller can be designed and the total network-induced delays, including the sensor-to-controller delays and the controller-to-actuator delays, can be presented with this controller.

The total delay is shown below.

$$\tau = \tau_s + \tau_{sc}^k + \tau_{sq} + \tau_c + \tau_{cq} + \tau_{ca}^k + \tau_a \tag{2}$$

To observe all of the system components, an observer is designed to watch the process environment. For minimum loss, integration with a high performance program for the distribution of networked-induced delays is designed. A delay analyzer is designed to support that integration. For a control system, with feedback values being communicated from a controller to the actuator over a noisy communication channel, randomly occurring bit errors will decrease the system's performance. If the channel is noisy and bit errors are frequent, the system's performance may decrease the stability [1]. In this stage, coding for robustness to time-varying data rates is very important to get optimal results.

The delay range is shown below.

$$T_{\min} < T < T_{\max}$$

To reduce the delay and prevent data packet losses, the setting of the timing has a very important role. Frames must be processed fast enough in order to prevent loss of frames due to the controller's lateness. Otherwise, unreliable functionality may occur due to packet losses. For this problem, a design using some arguments in the coding gives the solution. These arguments, such as data length, receiver ID, data bytes, command counter, and cyclic redundancy check (CRC) byte, can be embedded into the function that sets the connection between the system components. The CRC byte is used for the CRC and occurs from the sum of the receiver ID byte, the command counter byte, and the data bytes. The transmitter calculates the CRC on the transmission and sends it to the receiver, and the receiver calculates its own CRC. If the 2 CRCs do not match, an error has occurred [4].

5.2. Hardware optimization

Ethernet connectivity is faster than serial data connection protocols. On the Ethernet side, the ring topology is the preferred selection for the solution [6]. In ring topology, the broadcast packets can fall into an unlimited loop if they are not blocked or consumed by a network station. On the software side, the control with a logical token pass between the master and slave components prevents an unexpected packet storm in the network. Thus, the fault tolerance process that controls the data flows can be applied to the communication of the system. In a full duplex flow control system, packet transmission and reception are possible without any interruption. Using the common user datagram protocol (UDP) and transmission control protocol (TCP) over physical links gets the optimum result. Optimal sensor placement is another solution for the optimization of the system.

6. Structure of the system

6.1. Hierarchical structure

The industrial network is formed by system components and organizations in a specific industry. Industrial networks have a manager. To organize the network operations for its members, the industrial network manager is used. Its roles are: managing memberships, answering requests from the members, ensuring information flow among the members, distributing decision-supporting tools to the members, and developing new opportunities for the members [7]. In the implementation of the simulated system, the industrial network manager is referred to as the observer. The roles of the observer are similar to the roles of the industrial network manager that are defined above. The observer uses fuzzy logic [8] to operate the controllers. Fuzzy sets are constructed by the sensor's data. The other system components are sensors, controllers, and actuators. The sensor gets the information and sends a request to the corresponding controller. The controller takes the request and sends a response to the sensor. After the handshake of the sensor and the controller, the controller sends the related information to the corresponding actuator. The actuator processes the information and the next sensor takes the processed information. This processing continues until the final sensor produces the aimed information. Industrial networks are different from other data networks. For example, the shape of the traffic is different, available processing capabilities for data interpretation and validation are usually very limited, and the most important issue is the expected data error rate and the on-time delivery of the correct data.

The network links the system components. It supplies the transfer capabilities on the physical medium. The simulated system is implemented by a local area network (LAN). It links the sensors and actuators with the controller computers. In control applications for industrial automations, slave devices are organized around a master controller. Actuators and sensors are controlled by control software running on the master device [9]. All of the operations are illustrated in Figure 2.

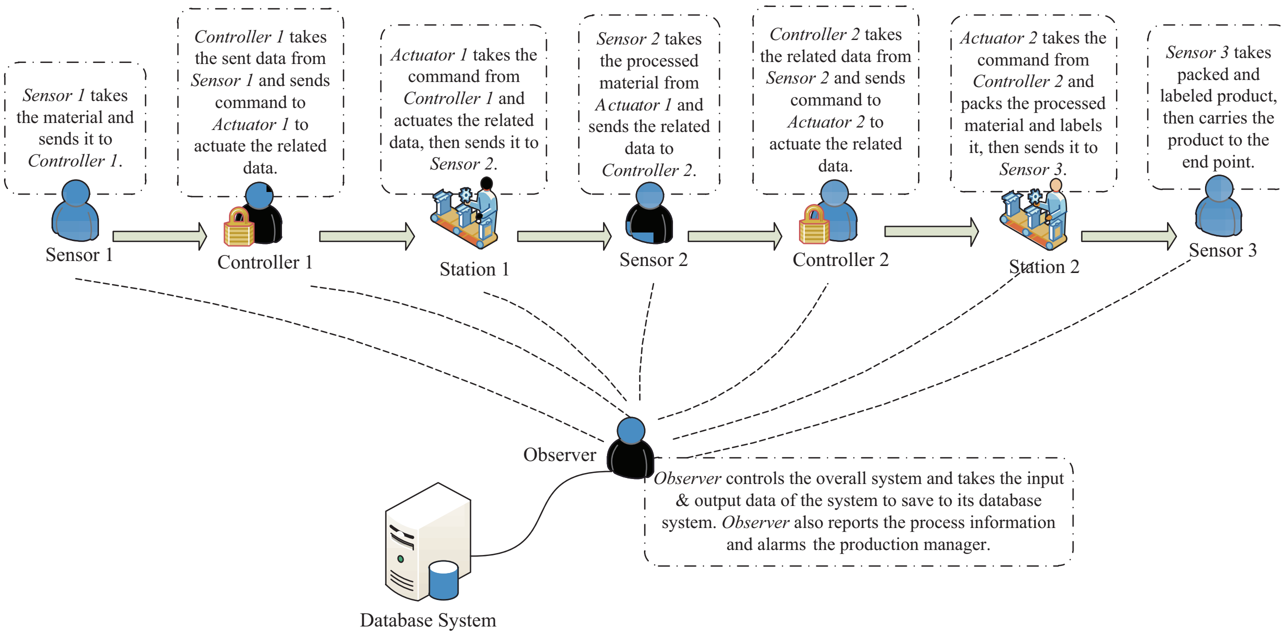

Figure 2. Structure of the system.

6.2. Communicational structure

The major role of LANs is data transmission. In the industrial LAN, reliability, responsiveness, and predictability are very important. With the development and use of smart sensors in digital controls, the development of these networks started in the 1980s [10]. Local industrial networks usually operate with twisted pair cables that have a low cost. Today, optical fiber is considered as a new medium for these networks.

Ethernet is a standardized concept and is widely introduced in automation domains. Ethernet has system limits regarding real-time behavior, because most of the solutions use the TCP (UDP)/Internet protocol (IP) functionality and a middle-ware above the TCP/IP stack that controls only the soft real-time and nonreal-time traffic. In the implementation of the simulated system, UDPClient class is used in the Visual C# windows application. The data packet transmission is supplied by UDP datagrams. By broadcasting, a program that is named Observer observes the overall system communication and reports the control process. In each component of the system, network interface cards are used for the communication between components using unshielded twisted pair (UTP) network cables.

7. Implementation of the simulated system

7.1. Overview

The aim of this work is to provide the most efficient product line for all work stations. Each station has different work capacities under divergent situations. Hence, we must consider each work stations' working optimization independently. Each station has an operating speed between α_{Min} and α_{Max}. All of the stations are required to produce the materials in the queues that are connected to them. All of the queues belonging to the working stations have an occupancy rate of F individually. Stations work sequentially in their direction of work. The station that has a queue with an increasing occupancy rate must be accelerated. Similarly, the station that has a low capacity queue should be slowed. All of these acceleration and slowing processes are done by autonomous mechanisms called agents. All of the occupancy rates of the queues are observed by an agent and all of the working stations report their operating speeds, α, to the observer instantly. The observer agent determines the speeds of the stations due to this sent data. The designated agent is notified about the change in speed. Hence, the requested production capacity is provided by changing the stations' operating speeds. A working methodology of the system is illustrated in Figure 3.

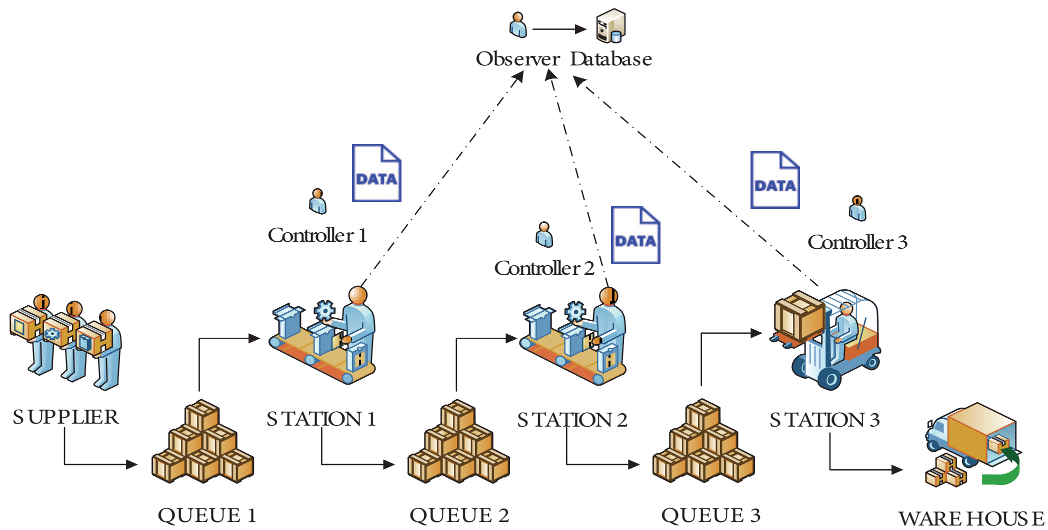

Figure 3. Overview of the system.

Fuzzy set values of the fuzzy variables F can be described as {VL, L, N, M, VM}belonging to the "very low", "low", "normal", "much", and "very much" levels shown in Figure 4. Moreover, operating speed α is described with numbers 0–9 such as {0,1,2,3,4,5,6,7,8,9}, in which 9 is the fastest and 0 is idle.

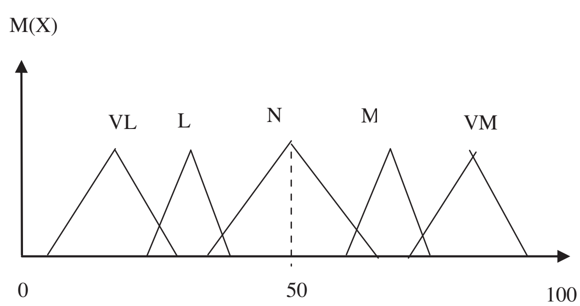

Figure 4. Fuzzy membership function for each linguistic fuzzy set value.

Next, we specify the fuzzy rule base. Fuzzy associations or rules associate the output fuzzy sets of the control values with input fuzzy sets of input-variable values. We can write fuzzy associations as antecedent–consequent pairs of IF-THEN statements, as shown in the Table.

Finally, we determine the output action given the input conditions.

7.2. Sensor simulation

In the sensor part, a program that is named Agent is coded to simulate a sensor. There are 3 sensor programs in the system. The first sensor is used for getting the material. It is used as the first product band. The second sensor is used for getting the processed material from the corresponding actuator and sending it to corresponding controller. The third sensor is used for getting the product from the last actuator and outputting the product.

Table. Fuzzy associative memory matrices for the working stations.

	VL	**L**	**N**	**M**	**VM**
VL	1	1	0	0	0
L	2	1	0	0	0
N	5	5	5	2	0
M	7	7	7	2	0
VM	9	9	9	3	0

	VL	**L**	**N**	**M**	**VM**
VL	1	1	1	0	0
L	2	2	1	0	0
N	6	5	5	2	0
M	8	6	5	2	0
VM	9	9	9	5	0

	A	**B**	**C**	**D**	**E**	**F**	**G**
VL	1	1	1	1	0	0	0
L	2	1	1	1	0	0	0
N	4	4	4	3	2	1	1
M	7	7	6	5	3	2	1
VM	9	8	7	6	4	1	0

7.3. Actuator simulation

In the actuator part, a program that is named Station is coded to simulate an actuator. There are 2 actuator programs in the system. The first actuator is used for getting the material from the corresponding sensor with the command of the corresponding controller. It processes the material and sends it to the related sensor. The second actuator is used for getting the processed material from the corresponding sensor with the command of the corresponding controller. It processes the input and outputs the product to the last sensor.

7.4. Controller

In the controller part, a program that is named Controller is coded for controlling the sensors. There are 2 controllers in the system. The first controller gets the material information from the related sensor and sends a command to the corresponding actuator for processing. The second controller gets the information from the corresponding sensor and sends a command to the corresponding actuator for processing.

7.5. Observer

In the observer part, a program that is named Observer is coded for observing the overall system. The observer program is set into one computer. It observes the system and gets the inputs and outputs of the system. Reporting of the system information is enabled by this program in a master computer. It shows the general information such as reports, alarms, and product amount.

8. Conclusion

New techniques have emerged for manufacturing places by the development of technology. Automation mechanisms were developed for production stages in factories. A controlling mechanism is used to process the systems. Thus, new control techniques have emerged using artificial intelligence with new generation algorithms.

Network-based connection has increased the speed of the communication between the system components in the system.

Finally, the incontestable value of the network-based structure in control systems is shown. The networked structure saves time for the overall system. Network-based control systems are still undergoing improvement. To supply the connection, Ethernet cards are used by connecting UTP cables. As a conclusion, the aim of this simulated system study was improving the performance of the control systems.

References

[1] J. Baillieul, P.J. Antsaklis, "Control and communication challenges in networked real-time systems", Proceedings of the IEEE, Vol. 95, pp. 9–28, 2007.

[2] L. Wei, D. Xue, D. E, "Some basic issues in networked control systems", 2nd IEEE Conference on Industrial Electronics and Applications, pp. 2098–2102, 2007.

[3] G. Kaplan, "Ethernet's winning ways", IEEE Spectrum, Vol. 38, pp. 113–115, 2001.

[4] V.K. Kongezos, C.R. Allen, "Wireless communication between A.G.V.'s (Autonomous Guided Vehicle) and the industrial network C.A.N (Controller Area Network)", Proceedings of the IEEE International Conference on Robotics and Automation, Vol. 1, pp. 434–437, 2002.

[5] Y. Zhang, Q. Zhong, L. Wei, "Stability of networked control systems with communication constraints", Chinese Control and Decision Conference, pp. 335–339, 2008.

[6] G. Yoon, D.H. Kwon, S.C. Kwon, Y.O. Park, Y.J. Lee, "Ring topology-based redundancy Ethernet for industrial network", SICE-ICASE International Joint Conference, pp. 1404–1407, 2006.

[7] S. D'Amours, P. Lefrançois, B. Montreuil, "Design of an information system for a networked industry", Proceedings of the International Conference on Engineering and Technology Management, pp. 128–133, 1996.

[8] Y.F. Lan, Y.K. Liu, "Fuzzy production planning problem with credibility service level constraints", 3rd International Conference on Innovative Computing Information and Control, p. 261, 2008.

[9] F. Fummi, S. Martini, M. Monguzzi, G. Perbellini, M. Poncino, "Software/network co-simulation of heterogeneous industrial networks architectures", Proceedings of the IEEE International Conference on Computer Design, pp. 496–501, 2004.

[10] Z. Cucej, D. Gleich, M. Kaiser, P. Planinsic, "Industrial networks", Proceedings of the 46th International Symposium Electronics in Marine, pp. 59–66, 2004.