IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 2, February 2017

Topological Conditions for Maintaining Stabilization of Dynamic System

Prof. Dilip Budhwant[1], **Bhagyashri Bhalekar**[1]

Department of Computer Science and Engineering, MGM's Jawaharlal Nehru College of Engineering, Aurangabad, Maharashtra, India[1]

Abstract: We study the problem of stabilizing a linear system over a wireless network using a simple in-network computation method. Specifically, we study an architecture called the "Wireless Control Network" (WCN), where each wireless node maintains a state, and periodically updates it as a linear combination of neighboring plant outputs and node states. This architecture has previously been shown to have low computational overhead and beneficial scheduling and compositionality properties. In this paper we characterize fundamental topological conditions to allow stabilization using such a scheme. To achieve this, we exploit the fact that the WCN scheme causes the network to act as a linear dynamical system, and analyze the coupling between the plant's dynamics and the dynamics of the network. We show that stabilizing control inputs can be computed in-network if the vertex connectivity of the network is larger than the geometric multiplicity of any unstable eigen value of the plant. This condition is analogous to the typical min-cut condition required in classical information dissemination problems. Furthermore, we specify equivalent topological conditions for stabilization over a wired (or point-to-point) network that employs network coding in a traditional way – as a communication mechanism between the plant's sensors and decentralized controllers at the actuators.

Keywords: Networked control systems, decentralized control, wireless sensor networks, structured systems, in-network control, network coding, cooperative control.

I. INTRODUCTION

With recent revolutions in sensor and actuator technologies, availability of powerful but inexpensive embedded computing and introduction of new multi-hop wireless network standards for industrial automation, control over wireless networks is becoming a disruptive technology. Traditional wired interconnections between the plant sensors, controllers and actuators can be replaced by wireless multi-hop mesh networks, yielding cost and space savings for the plant operator.

Despite this tremendous promise, the introduction of wireless communications into the feedback loop presents several challenges for real-time feedback control. For instance, delays may be introduced if a multi-hop wireless network is used to route information between the plant sensors, actuators and controllers. Furthermore, transmissions in the network must be scheduled carefully to avoid packet dropouts due to collisions between neighbouring nodes. These issues can be detrimental to the goal of maintaining stability of the closed loop system if not explicitly accounted for, and substantial research has been devoted to understanding the performance limitations in such settings.

II. AIM AND OBJECTIVE

To model resource constrained nodes, we assumed that each node is capable of maintaining only a limited internal state. We then presented a distributed algorithm in the form of a linear iterative strategy for each node to follow, where each node periodically updates its state to be a linear combination of the states of the nodes in its immediate neighbourhood. The actuators of the plant also apply linear combinations of the states of the nodes in their neighbourhood. Given a linear plant model and the network's topology, we devised a design-time procedure to derive the coefficients of the linear combinations for each node and actuator to apply in order to stabilize the plant. We showed that our method could also handle a sufficiently low rate of packet dropouts in the network to maintain mean square stability. We referred to this paradigm, where the computation of the control law is done in-network as a wireless control network (WCN).

The scheme has several benefits, including easy scheduling of wireless transmissions, compositional design, and the ability to handle geographically separated sensors and actuators.we illustrated the use of the WCN in industrial process control applications.

III. EXISTING SYSTEM

The introduction of wireless communications into the feedback loop presents several challenges for real-time feedback control. For instance, delays may be introduced if a multi-hop wireless network is used to route Furthermore, transmissions in the network must be scheduled carefully information between the plant sensors, actuators and controllers. To avoid packet dropouts due to collisions between neighboring nodes. These issues can be

DOI 10.17148/IJARCCE.2017.6298

detrimental to the goal of maintaining stability of the closed loop system if not explicitly accounted for, and substantial research has been devoted to understanding the performance limitations in such settings These works typically adopt the convention of having one or more dedicated controllers or state estimators located in the system, and study the stability of the closed loop system assuming that the sensor estimator and/or controller-actuator communication channels are unreliable (dropping packets with a certain probability, For this standard architecture the use of dedicated controllers imposes a routing requirement along one or more fixed paths through the network, along with strict end-to-end delay constraints to ensure stability.

IV. IMPLEMENTATION

This section describes the support software, materials, equipment, and facilities required for the implementation, as well as the personnel requirements and training necessary for the implementation.

The information provided in this section is not site-specific. If there arc additional support requirements not covered by the subsequent sections, others may be added as needed.

Hardware, Software, Facilities, and Materials

a. Hardware

This section provides a list of support equipment and includes all hardware used for testing time implementation. For example, if a client/server database is implemented on a LAN, a network monitor or “sniffer” might be used, along with test programs. to determine the performance of the database and LAN at high-utilization rates. If the equipment is site-specific, list it in Section 4, Implementation Requirements by Site.

b. Software

This section provides a list of software and databases required to support the implementation. Identify the software by name, code, or acronym. Identify which software is commercial off-the-shelf and which is State-specific. Identify any software used to facilitate the implementation process.

c. Facilities

In this section, identify the physical facilities and accommodations required during implementation. Examples include physical workspace for assembling and testing hardware components, desk space for software installers, and classroom space for training the implementation stall. Specify the hours per day needed, number of days, and anticipated dates.

d. Material

This section provides a list of required support materials, such as magnetic tapes and disk packs.

e. Personnel

This section describes personnel requirements and any known or proposed staffing requirements, if appropriate. Also describe the training, if any, to be provided for the implementation staff.

f. Personnel Requirements and Staffing

In this section, describe the number of personnel, length of time needed, types of skills, and skill levels for the staff required during the implementation period. If particular staff members have been selected or proposed for the implementation, identify them and their roles in the implementation.

g. Training of Implementation Staff

This section addresses the training, if any, necessary to prepare staff for implementing and maintaining the system; it does not address user training, which is the subject of the Training Plan. Describe the type and amount of training required for each of the following areas, if appropriate, for the system:

- System hardware/software installation
- System support
- System maintenance and modification

Present a training curriculum listing the courses that will be provided, a course sequence. and a proposed schedule. If appropriate, identify which courses particular types of staff should attend by job position description.

If training will be provided by one or more commercial vendors, identify them, the course name(s), and a brief description of the course content.

If the training will be provided by State staff, provide the course name(s) and an outline of the content of each course. Identify the resources, support materials, and proposed instructors required to teach the course(s).

h. Performance Monitoring

This section describes the performance monitoring tool and techniques and how it will be used to help decide if the implementation is successful.

i. Configuration Management Interface

This section describes the interactions required with the Configuration Management (CM) representative on CM-related issues, such as when software listings will be distributed, and how to confirm that libraries have been moved from the development to the production environment.

V. DETAILED DESIGN

This section provides the information needed for a system development team to actually build and integrate the hardware components, code and integrates the software modules, and interconnects the hardware and software segments into a functional product. Additionally, this

DOI 10.17148/IJARCCE.2017.6298

section addresses the detailed procedures for combining separate COTS packages into a single system. Every detailed requirement should map back to the FRD, and the mapping should be presented in an update to the RTM and include the RTM as an appendix to this design document.

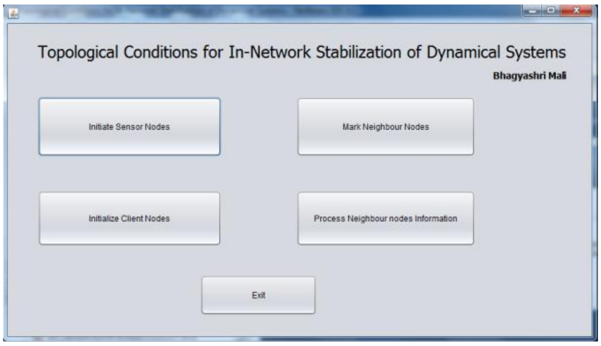


Fig. Server and node a started

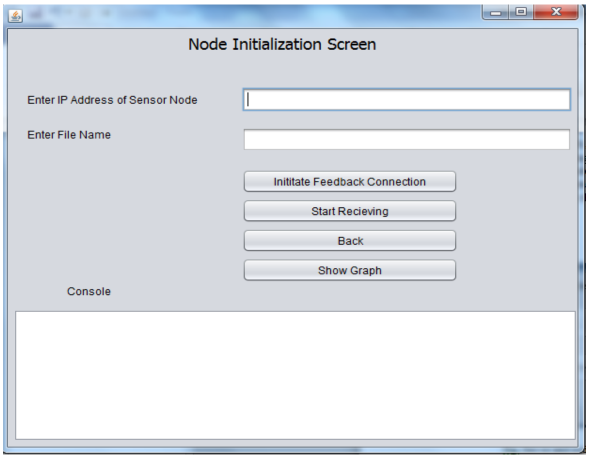
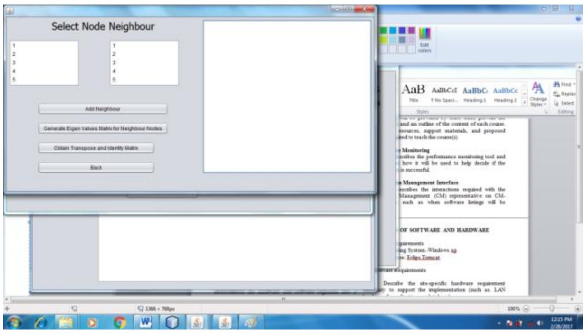
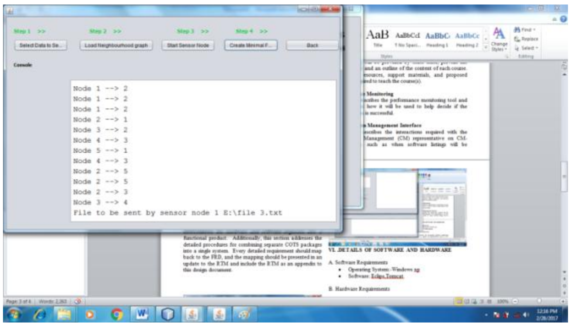


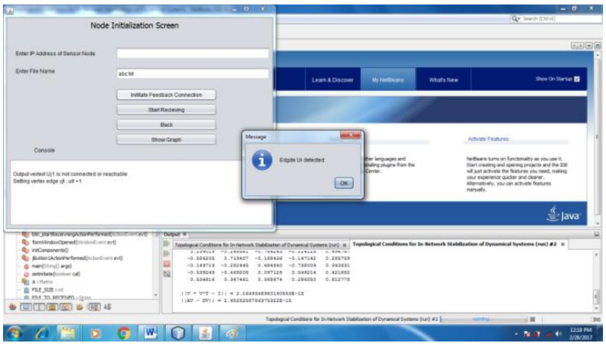
Fig. Node initialization screen





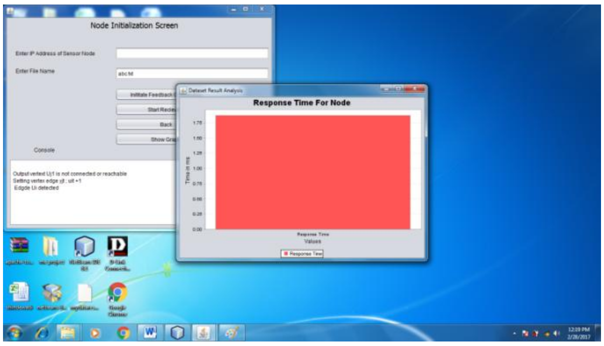
```
Node 1 --> 2
Node 1 --> 2
Node 1 --> 2
Node 2 --> 1
Node 3 --> 2
Node 4 --> 3
Node 5 --> 1
Node 4 --> 3
Node 2 --> 5
Node 2 --> 5
Node 2 --> 3
Node 3 --> 4
File to be sent by sensor node 1 E:\file 3.txt
```

Fig selecting data to send and loading neighboring file



```
Output verted Uj1 is not connected or reachable
Setting vertex edge yjt : uit +1
```

Fig: Receiving file from server



VI. EXPERIMENTAL RESULT

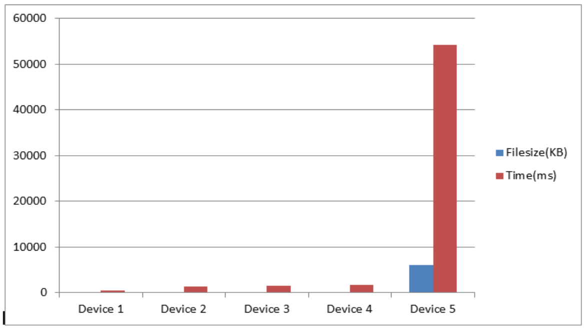


Fig: Base paper result

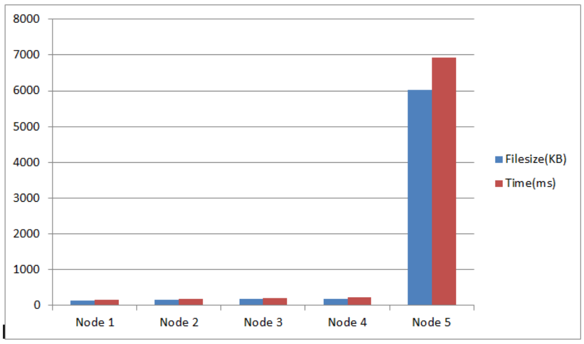


Fig: project result

VII. DETAILS OF SOFTWARE AND HARDWARE

A. Software Requirements

- Operating System:-Windowsxp
- Software: Eclips, Tomcat.

DOI 10.17148/IJARCCE.2017.6298

B. Hardware Requirements

Describe the site-specific hardware requirement necessary to support the implementation (such as. LAN hardware for a client/server database)

- System : Pentium IV 2.4 GHz.
- Hard Disk : 40 GB.
- Floppy Drive : 1.44 Mb.
- Monitor : 15 VGA Colour.
- Mouse : Logitech.
- RAM : 256 Mb.

C. Data Requirements

Describe specific data preparation requirements and data that must be available for the system implementation. An example would be the assignment of individual IDs associated with data preparation.

D. Facilities Requirements

Describe the site-specific physical facilities and accommodations required during the system implementation period. Some examples of this type of information are provided in Section 3.

VIII. ADVANTAGES

In decentralized control systems, a set of non-interacting local controllers is used to control a dynamical system (plant); each of the controllers generates the appropriate plant inputs by observing only a subset of the plant's outputs. Due to these limitations imposed on each of the local controllers, it is possible that even a controllable and observable system cannot be stabilized with the aforementioned setup.

There are two distinct reasons for a fixed mode. A fixed mode can either arise from a loss of rank due to a 'perfect cancellation' of the numerical parameters (which is a degenerate case), or it can be caused by deeper issues relating to the system structure. The latter set of fixed modes are called structural fixed modes.

IX. FUTURE WORK

The main difference between centralized and decentralized control is the communication. Controllers in a decentralized system can communicate with each other to achieve their common goal. In this paper, we argue that even linear time-invariant controllers in a decentralized linear system "communicate" via linear network coding to stabilize the plant. To justify this argument, we propose an algorithm to "externalize" the implicit communication between controllers that we believe must be occurring to stabilize the plant. Based on this, we show that the stabilizability condition for decentralized linear systems comes from an underlying communication limit, which can be described by an algebraic mincut-maxflow theorem.

X. CONCLUSION

In this paper, we have studied the problem of stabilizing a given dynamical system over a network. In contrast to traditional approaches that treat the network purely as a routing mechanism (delivering sensor measurements to controllers, and control inputs to actuators), we propose a fundamentally different approach that relies on inducing carefully chosen dynamics on the network (via the form of a simple distributed algorithm), and using those dynamics to stabilize the plant. This approach does away with end-to-end routing entirely, and only requires that nodes transmit information to their nearest neighbors at each time-step. We provided topological conditions on the network that allow the system to be stabilized in this manner. Specifically, we showed that if the network is sufficiently well connected, each node and actuator can use a linear iterative strategy with appropriately chosen weights to stabilize the plant; furthermore, the connectivity required is determined by the dynamics of the plant, rather than the number of source nodes (as in traditional information transmission scenarios). Our approach also extends in a straightforward manner to wired (point-to-point) networks via a standard graph transformation.

REFERENCES

[1] M. Pajic, S. Sundaram, G. J. Pappas, and R. Mangharam, "Topological Conditions for Wireless Control Networks," in Proc. 50th IEEE Conference on Decision and Control, 2011, pp. 2353–2360.

[2] J. P. Hespanha, P. Naghshtabrizi, and Y. Xu, "A survey of recent results in networked control systems," Proc. IEEE, vol. 95, no. 1, pp. 138–162, 2007.

[3] L. Schenato, B. Sinopoli, M. Franceschetti, K. Poolla, and S. S. Sastry, "Foundations of control and estimation over lossy networks," Proc. IEEE, vol. 95, pp. 163–187, 2007.

[4] C. L. Robinson and P. R. Kumar, "Optimizing controller location in networked control systems with packet drops," IEEE J. Sel. Areas Commun., vol. 26, no. 4, pp. 661–671, 2008.

[5] M. Pajic and R. Mangharam, "Embedded virtual machines for robust wireless control and actuation," in RTAS'10: 16th IEEE Real-Time and Embedded Technology and Applications Symposium, 2010, pp. 79–88.

[6] R. Alur, A.D'Innocenzo, K. H. Johansson, G. J. Pappas, and G. Weiss, "Compositional modeling and analysis of multi-hop control networks," IEEE Trans. Autom. Control, vol. 56, no. 10, pp. 2345–2357, 2011.

[7] M. Pajic, S. Sundaram, R. Mangharam, and G. J. Pappas, "The Wireless Control Network: A New Approach for Control over Networks," IEEE Trans. Autom. Control, vol. 56, no. 10, pp. 2305–2318, 2011.

[8] M. Pajic, S. Sundaram, J. Le Ny, G. J. Pappas, and R. Mangharam, "Closing the loop: A simple distributed method for control over wireless networks," in Proc. 11th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN), 2012, pp. 25–36.

[9] A. Giridhar and P. Kumar, "Computing and communicating functions over sensor networks," IEEE J. Sel. Areas Commun., vol. 23, no. 4, pp. 755 – 764, 2005.

[10] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," Systems & Control Letters, vol. 53, no. 1, pp. 65–78, 2004.

DOI 10.17148/IJARCCE.2017.6298