**STABILIZING A NETWORK IN LINEAR**

**NETWORK SYSTEMS**

**A MAJOR PROJECT REPORT**

Submitted in partial fulfillment of the Requirement

for the award of the degree of

**Bachelor of Engineering**

in

**Computer Science and Engineering**

By

G. SHREYA (H.T. No: 1608-19-733-023)

K. RAJA SHEKER (H.T. No: 1608-19-733-022)

M. VENKATESH (H.T. No: 1608-19-733-037)

Under the guidance of

A.V. MURALI KRISHNA

Assistant.Professor



**Department of Computer Science and Engineering**

**mATRUSRI ENGINEERINg collegE**

**JANURARY, 2022**

(i)

**mATRUSRI ENGINEERING COLLEGE**

**SAIDABAD - 500059**



**Department of Computer Science and Engineering**

Certificate from PROJECT GUIDE

This is to certify that the project report entitled **“STABILIZING A NETWORK IN LINEAR NETWORK SYSTEMS”** submitted by **Ms. G. SHREYA** bearing **H.T. No: 1608-19-733-023, Mr. K. RAJA SHEKER** bearing **H.T. No: 1608-19-733-022, Mr. M.VENKATESH** bearing **H.T.No: 1608-19-733-037** in the partial fulfillment of the requirement for the award of the degree of **Bachelor of Engineering** in **Computer Science and Engineering** is a bonafide work carried by them**.**

The results of the investigations enclosed in this report have been verified and found satisfactory.

**A.V. MURALI KRISHNA DR. P. VIJAYAPAL REDDY**

**(Assistant.Professor) HOD**

**Dept. of CSE Dept. of CSE**

**MECS MECS**

**(**ii)

**mATRUSRI ENGINEERING COLLEGE**

**SAIDABAD - 500059**



**Department of Computer Science and Engineering**

dECLARATION BY THE CANDIDATE

we, **G. SHREYA** bearing **H.T. No: 1608-19-733-023, Mr. K. RAJA SHEKER** bearing **H.T.No: 1608-19-733-022, Mr. M.VENKATESH** bearing **H.T.No: 1608-19-733-037** hereby certify that the mini project report entitled **“RATING AN AIRPORT MAPPING DATABASE”** is submitted in the partial fulfillment of the requirement for the award of the degree of **Bachelor of Engineering** in **Computer Science and Engineering.**

This is a Record of bonafide work carried out by us under the guidance of **Mr. A.V. MURALI KRISHNA**, Assistant Professor, Matrusri Engineering College, Saidabad. The results embodied in this report have not been reproduced/copied from any source. The results embodied in this report have not been submitted to any other university or institute for the award of any other degree or diploma.

G. SHREYA (H.T. No: 1608-19-733-023)

K. RAJA SHEKER (H.T. No: 1608-19-733-022)

M. VENKATESH (H.T. No: 1608-19-733-037)

**Dept. of CSE**

**(**iii) **MECS.**

**ACKNOWLEDGEMENT**

We wish to take this opportunity to express our deep gratitude to all the people who have extended their cooperation in various ways during our mini project work. It’s our pleasure to acknowledge the help of all those individuals.

Firstly, we would like to thank Dr.P. Vijayapal Reddy (HOD, CSE DEPT) for his encouragement and valuable guidance in bringing shape to dissertation.

We would like to thank our project guide, Mr. A.V. MURALI KRISHNA (Assistant Professor) for his guidance and help throughout the development of this project work by providing us with required information and support. Without his guidance, cooperation and encouragement, we couldn’t have learnt many new things during our mini project tenure.

G. SHREYA (1608-19-733-023)

K. RAJA SHEKER (1608-19-733-022)

M. VENKATESH (1608-19-733-037)

(iv)

**ABSTRACT**

Linear network systems in which nodes transmit data from source nodes to sink nodes by means of linear combinations. The project intents to stabilize the network with the data control in the network among the nodes. Many systems are working by use of networks and agents are used to make prediction to control network system.

Stabilized linear network system should be established for the access of the nodes in the network and control the system. This is to improve the network throughput, efficiency, scalability and get the optimization path to the network. This process used to attain the maximum possible information flow in the network. By analyzing all the paths of the network, the optimized network is found in the network among the nodes.

(v)

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| S.No | Contents | Page.No. |
| 1 | Introduction |  |
| 2 | Literature Survey |  |
| 3 | Methodology of the project |  |
| 4 | Technical Architecture |  |
| 5 | Research review write up |  |
| 6 | References |  |

(vi)

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| Figure No. | Figure | Page No. |
| 1 | Technical Architecture |  |
| 2 | Network Topology |  |

(vii)

**1.INTRODUCTION**

Network systems arise in many applications. In the system, agents must use sampled data to characterize the network must coordinate with one another to predict the network response and decide their control actions. It computes the control in a distributed manner and pose the control synthesis as a network optimization in the system. The network system is established in graphical manner with the nodes and optimized path selection by the distributed data is predicted with least distance among the nodes.

Stabilizing the network in the system by considering the distributed data of the nodes. The node will sense the data and check the neighboring node data to get the distributed data. Sensor nodes are generally equipped with a radio transceiver, a microcontroller, a memory unit, and a set of transducers with which they can acquire and process data. To reduce each node’s size and energy requirements this method is efficient. The path to the actuator node and the base station without interruption of the other networks. The path should be identified among the nodes and resultant optimal path is recorded.

Data based predictive control method of the node should be used to analyze an efficient path to control system. Networks can broadly classify its benefits as resulting in improvement in a network's throughput, efficiency and scalability. Nodes monitor and record the physical conditions of the environment and forward the collected data to a central location.

* 1. **OBJECTIVES**

The main objective of the project is to stabilize the network among the nodes and efficient network link among the nodes with the use of actuator nodes and the neighboring nodes of the source node. The required data is obtained from the neighboring nodes and the node itself and then check the least distance, less energy resources usage in the network. The Data based predictive control algorithm is used to predict the network link to the nodes and optimization technique is used to get the optimized path among the nodes.

1

* 1. **EXISTING SYSTEM**

Network system stabilizing is the complex activity. The network is stabilized by time series data, reinforcement learning and by behavioral approach of the network. It evaluates

performance and systematic patterns over time and determine controller directly from data and also control the agents of the network.

**1.2.1 DISADVANTAGES:**

* There is more utilization of the energy resources.
* The model-based learning give only the trajectory values as input.
* Problem with accurately identifying correct model to represent data.
* Complexity in accurate prediction of network behavior.
  1. **PROPOSED SYSTEM**

Stabilizing the network in data-based model is one of the possible optimized approaches. The network system given with the distributed data as input and the actuator node to find the stable network and control the system. Nodes use sampled data to characterize the network and coordinate with one another to predict the network response and decide their control actions.

**1.3.1 ADVANTAGES:**

* The energy consumption of resources and load taken is less.
* The network is stabilized with the node and its neighboring node data also.
* Predict path to control system in optimized technique.

2

* 1. **MODULES**
* Network system formation phase
* Initialize network with actuator nodes
* Optimized network
* Optimization graph
  1. **MODULE DESCRIPTION**

Network system formation phase

In this module, the graph with the nodes and edges is developed and will give unique ID to each node. Each node will sense some data and then using connecting edges will send the data to base station via actuator node.

Initialize network with actuator nodes

In this module, each node will read distributed data from its neighbor and then calculated actuator node which is closer to base station. This actuator node will receive data from another node through source node and send to base station.

Optimized network

In this module, the selected source node will get the efficient path. The node will collect the data and then find the trajectory values distance, load and energy consumption of the network in the system. After the several paths of the network the optimized network path is selected.

Optimization graph

In this module, we plot the graph with x axis as the number of data sending in the network and y axis as the resource consumption like energy resources, load and time. The graph is plotted with resource consumption in network system with and without optimization technique.

3

**2.LITERATURE SURVEY**

The survey about the project is done by taking various review papers and the models created. Several procedures are used to gain the stability in the networks. There are methods such as convex programming, convex relaxations, Reinforcement learning, Behavioral approaches, Model predictive control, Time series data and Linear quadratic regulation. Research papers help us attain all the possibilities for the stability of the network in the system.

1. M. Rotkowitz and S. Lall, “A characterization of convex problems in decentralized control,” IEEE Trans. Autom. Control, vol. 51, no. 2, pp. 274–286, Feb. 2006.

The decentralized control has become increasingly important, where one has multiple controllers each with access to different information. There is no known tractable algorithm for computing the optimal solution. However, there are also several special cases of this problem for which efficient algorithms have been found. And applied this to some specific constraint classes. We showed that for distributed systems with delays, optimal controllers may be synthesized in this manner. These results are developed in a very general framework, and are shown to hold in both continuous and discrete time, for both stable and unstable systems, and for any norm. This notion unifies many previous results identifying specific tractable decentralized control

problems, and delineates the largest known class of convex problems in decentralized control. As an example, we show that optimal stabilizing controllers may be efficiently computed in the case where distributed controllers can communicate faster than their dynamics propagate. For the convex problems in the network with convex relaxations the suboptimal controllers for the network are obtained.

1. F. Bullo, J. Cortés, and S. Martinez, Distributed Control of Robotic Networks (Applied Mathematics Series). Princeton, NJ, USA: Princeton Univ. Press, 2009.

This is to present a coherent introduction to basic distributed algorithms for robotic networks. This emerging discipline sits at the intersection of different areas such as distributed algorithms, parallel processing, control, and estimation. This is to provide a self-contained, broad exposition of the notions and tools from these areas that are relevant in cooperative control problems. These concepts include graph-theoretic notions like connectivity, adjacency and Laplacian matrices, distributed algorithms

4

from computer science like leader election, basic tree computations and from parallel

processing, and geometric models and optimization. This is to put forth a model for robotic networks that helps to rigorously formalize coordination algorithms running on them. We illustrate how computational geometry plays an important role in modeling the interconnection topology of robotic networks. We draw on classical notions from distributed algorithms to provide complexity measures that characterize

the execution of coordination algorithms. Such measures allow us to quantify the algorithm performance and implementation costs.

1. J. Kober, J. A. Bagnell, and J. Peters, “Reinforcement learning in robotics: A survey,” Int. J. Robot. Res., vol. 32, no. 11, pp. 1238–1274, 2013.

Reinforcement learning offers to robotics a framework and set of tools for the design of sophisticated and hard-to-engineer behaviors. Conversely, the challenges of robotic problems provide both inspiration, impact, and validation for developments in reinforcement learning. A reinforcement learning problem is to find a policy that optimizes the long term sum of rewards. a reinforcement learning algorithm is one designed to find such an optimal policy. The reward function in this example could be based on the success of the hits as well as secondary criteria like energy consumption. Reinforcement learning may be understood by contrasting the problem with other areas of study in machine learning. To aid in understanding the RL problem and its relation with techniques widely used within robotics, it provides a schematic illustration of two axes of problem variability: the complexity of sequential interaction and the complexity of reward structure. Reinforcement learning research has placed great focus on addressing cases which are analytically intractable using approximations and data-driven techniques. One of the most important approaches to reinforcement learning within robotics centers on the use of classical optimal control techniques to system models learned via repeated interaction with the environment. As the complexity increases the prediction and training for the nodes increased and attaining stability makes difficult.

5

1. L. Bu, R. Babu, and B. D. Schutter, “A comprehensive survey of multiagent reinforcement learning,” IEEE Trans. Syst., Man, Cybern. C, Appl. Rev., vol. 38, no.2.

Multi agent systems are rapidly ﬁnding applications in a variety of domains, including

robotics, distributed control, telecommunications, and economics. The complexity of many tasks arising in these domains makes them difﬁcult to solve with preprogramed agent behaviors. The agents must, instead, discover absolution on their own, using learning. Different view-points on this issue have led to the proposal of many different goals, among which two focal points can be distinguished: stability of the agents’ learning dynamics, and adaptation to the changing behavior of the other agents. Multiagent reinforcement learning research is placed on ensuring that MARL algorithms (eventually) converge to desirable equilibria. As in standard reinforcement learning, convergence generally requires sufficient exploration of strategy space. However, exploration often comes at a price in the form of penalties or foregone opportunities. In multiagent settings, the problem is exacerbated by the need for agents to ``coordinate'' their policies on equilibria. We propose a Bayesian model for optimal exploration in MARL problems that allows these exploration costs to be weighed against their expected benefits using the notion of value of information. Unlike standard RL models, this model requires reasoning about how one's actions will influence the behavior of other. Algorithms for dynamic tasks were analyzed more closely, but techniques for static tasks were investigated as well. A classiﬁcation of MARL algorithms was given, and the different viewpoints on the central issue of the MARL learning goal were presented. Control theory can contribute in addressing issues such as stability of learning dynamics and robustness against uncertainty in other agents.

1. T. M. Maupong and P. Rapisarda, “Data-driven control: A behavioral approach,” Syst. Control Lett., vol. 101, pp. 37–43, Mar. 2017.

In this work, we study the design of a controller using system data. We present three

data-driven approaches based on the notion of control as interconnection. In the first

approach, we use both the data and representations to compute control variable trajectories that impose a prescribed path on the to-be-controlled variables.

6

The second method is completely data-driven and we prove sufficient conditions for determining a controller directly from data. Finally, we show how to determine a controller directly from data in the case where the control and to-be-controlled variables coincide. Network behavior analysis is the practice of analyzing network activity based on behavioral patterns in addition to traffic signatures to detect anomalies in performance and security. Network behavior analysis monitors the inside happenings of an active network by collecting data from many data points and devices to give a detailed offline analysis. It is constantly watching the network, marking known and unknown activities, new and unusual patterns and indicating potential threats by flagging. The program also checks and accounts for change in bandwidth and protocol being used during communication.

1. S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, “Distributed optimization and statistical learning via the alternating direction method of multipliers,” Found. Trends Mach. Learn., vol. 3, no. 1, pp. 1–122, 2011.

Many problems of recent interest in statistics and machine learning can be posed in

the framework of convex optimization. Due to the explosion in size and complexity of

modern datasets, it is increasingly important to be able to solve problems with a very large number of features or training examples. As a result, both the decentralized collection or storage of these datasets as well as accompanying distributed solution methods are either necessary or at least highly desirable. general distributed optimization, extensions to the nonconvex setting, and efficient implementation, including some details on distributed MPI and Hadoop MapReduce implementations. when applying ADMM to a very large model fitting problem, each update reduces to a model fitting problem on a smaller dataset. It can be viewed as a simple way of ‘bootstrapping’ specialized algorithms for small to medium sized problems.

7

**3.METHODOLOGY OF THE PROJECT**

Python is a high-level, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation. Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including structured, object-oriented and functional programming. It is often described as a "batteries included" language due to its comprehensive standard library. Guido van Rossum began working on Python in the late 1980s as a successor to the ABC programming language and first released it in 1991 as Python 0.9.0. Python consistently ranks as one of the most popular programming languages. Python is meant to be an easily readable language. Its formatting is visually uncluttered and often uses English keywords where other languages use punctuation. Unlike many other languages, it does not use [curly brackets](https://en.wikipedia.org/wiki/Curly_bracket_programming_language) to delimit blocks, and semicolons after statements are allowed but rarely used. It has fewer syntactic exceptions and special cases than [C](https://en.wikipedia.org/wiki/C_(programming_language)) or [Pascal](https://en.wikipedia.org/wiki/Pascal_(programming_language)).

Python's large standard library provides tools suited to many tasks and is commonly cited as one of its greatest strengths. It includes modules for creating [graphical user interfaces](https://en.wikipedia.org/wiki/Graphical_user_interface), connecting to [relational databases](https://en.wikipedia.org/wiki/Relational_database), [generating pseudorandom numbers](https://en.wikipedia.org/wiki/Pseudorandom_number_generator), arithmetic with arbitrary-precision decimals, manipulating [regular expressions](https://en.wikipedia.org/wiki/Regular_expression), and [unit testing](https://en.wikipedia.org/wiki/Unit_testing). Most of the standard library is cross-platform Python code, only a few modules need altering or rewriting for variant implementations. Python can serve as a [scripting language](https://en.wikipedia.org/wiki/Scripting_language) for [web applications](https://en.wikipedia.org/wiki/Web_application). With [Web Server Gateway Interface](https://en.wikipedia.org/wiki/Web_Server_Gateway_Interface), a standard API has evolved to facilitate these applications. Libraries such as [NumPy](https://en.wikipedia.org/wiki/NumPy), [SciPy](https://en.wikipedia.org/wiki/SciPy), and [Matplotlib](https://en.wikipedia.org/wiki/Matplotlib) allow the effective use of Python in scientific computing, with specialized libraries such as [Biopython](https://en.wikipedia.org/wiki/Biopython) and [Astropy](https://en.wikipedia.org/wiki/Astropy) providing domain-specific functionality.

Python is commonly used in [artificial intelligence](https://en.wikipedia.org/wiki/Artificial_intelligence) projects and machine

learning projects.  As a scripting language with a [modular architecture](https://en.wikipedia.org/wiki/Modular_programming), simple syntax

and rich text processing tools. While [The Python Language Reference](https://docs.python.org/3/reference/index.html#reference-index) describes the exact syntax and semantics of the Python language, this library reference manual describes the standard library that is distributed with Python. It also describes some of the optional components that are commonly included in Python distributions.

8

NumPy is a Python library used for working with arrays. It also has functions for working in domain of linear algebra, Fourier transform, and matrices. The array object in NumPy is called ndarray, it provides a lot of supporting functions that make working with ndarray very easy. NumPy arrays are stored at one continuous place in memory unlike lists, so processes can access and manipulate them very efficiently. NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays. It is the fundamental package for scientific computing with Python. It is open-source software. It contains various features. NumPy users include everyone from beginning coders to experienced researchers doing state-of-the-art scientific and industrial research and development. The NumPy API is used extensively in Pandas, SciPy, Matplotlib, scikit-learn, scikit-image and most other data science and scientific Python packages.

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. Matplotlib makes easy things easy and hard things possible. Matplotlib is an amazing visualization library in Python for 2D plots of arrays. Matplotlib is a multi-platform data visualization library built on NumPy arrays. Matplotlib is a python library used to create 2D graphs and plots by using python scripts. It has a module named pyplot which makes things easy for plotting by providing feature to control line styles, font properties, formatting axes etc. It supports a very wide variety of graphs and plots namely - histogram, bar charts, power spectra, error charts etc. It is used along with NumPy to provide an environment that is an effective open-source alternative for MATLAB.

Python bindings provide support for importing ns-3 model libraries as Python modules. Coverage of most of the ns-3 C++ API is provided. The intent has been to allow the programmer to write complete simulation scripts in Python, to allow integration of ns-3 with other Python tools and workflows. The intent is not to provide a different language choice to author new ns-3 models implemented in Python.

9

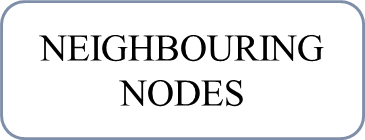
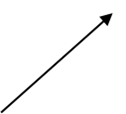
**4.TECHNICAL ARCHITECTURE**

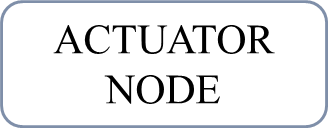
Technical Architecture is a form of IT architecture that is used to design computer systems. It involves the development of a technical blueprint with regard to the arrangement, interaction, and interdependence of all elements so that system-relevant requirements are met. Technical architecture which is also often referred to as application architecture, IT architecture, business architecture and refers to creating a structured software solution that will meet the business needs and expectations while providing a strong technical plan for the growth of the software application. Technical Architecture is the name of the total concept that is applied to the IT Infrastructure of an organization. IT Infrastructure is a coherent set of interconnected hardware and software.

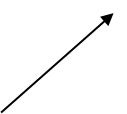
IT Infrastructure can be seen as a logical or physical structure. Architecture can be seen as the conceptual structure. Topology defines the structure of the network of how all the components are interconnected to each other. There are two types of topologies like physical and logical topology. Physical topology is the geometric representation of all the nodes in a network. A Network Topology is the arrangement with which computer systems or network devices are connected to each other. Topologies may define both physical and logical aspect of the network. Both logical and physical topologies could be same or different in a same network.

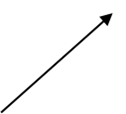
10

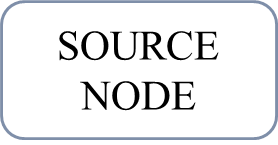
**4.1 ARCHITECTURE**



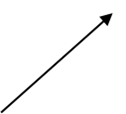


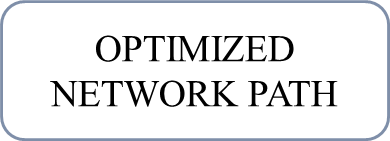


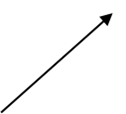
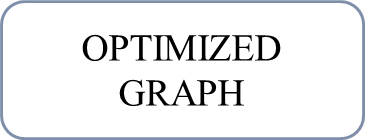






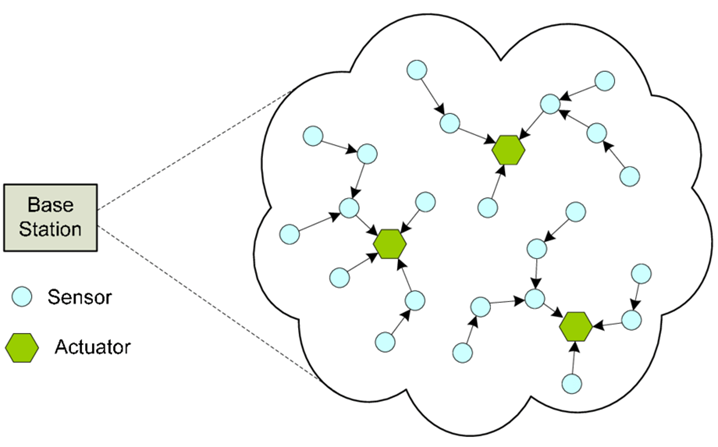






11

**4.2 NETWORK TOPOLOGY**

****

12

**5.RESEARCH REVIEW WRITE UP**

The project is done by taking various review papers and the models created. Several procedures are used to gain the stability in the networks. There are methods such as convex programming, convex relaxations, Reinforcement learning, Behavioral approaches, Model predictive control, Time series data and Linear quadratic regulation. Research papers help us attain all the possibilities for the stability of the network in the system.

However, network system is a NP hard problem but under certain conditions convex programming got the suboptimal controllers in the system. In few methods, the computation of the controller is typically done offline and requires knowledge of the underlying system model. Reinforcement learning is also increasingly popular approach for controlling the robots. But this requires very large number of samples to perform effectively and their complexity becomes difficult to get stability and safety of the network. Another method is the model predictive control, it is used for multi agent systems and have parameterized model for model predictive control formulations. The fundamental approach from behavioral analysis, it characterizes system trajectories from a single sample trajectory.

Data based predictive control algorithm is the approach without a system model, nodes use the sampled data to characterise the network behaviour and control the system. It means that the nodes have information that is measured and must coordinate with one another to predict the path and motivate the focus on distributed databased control of network systems.

13

**6.REFERENCES**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.NO** | **TITLE** | **AUTHOR** | **PUBLISHER** | **YEAR** |
| 1. | Convex relaxation for optimal distributed control | G. Fazelnia, R. Madani, A. Kalbat | IEEE Trans Autom control vol 62 | 2017 |
| 2. | A comprehensive survey of multiagent reinforcement learning | R. Babu, B. D. Schutter | IEEE Trans Syst Man Cybern control | 2008 |
| 3. | Cooperative distributive model predictive control | B. T. Stewart, A. N. Venkat, J.B Rawlings | Sys. Control vol.59 | 2010 |
| 4. | Data driven control: A behavioral approach | T. M Maupong, P. Rapisarda | Sys control Lett vol. 101 | 2017 |
| 5. | Graph theoretic methods in multiagent networks | M. Mesbahi, M. Egerstedt | Princeton NJ Univ Press | 2010 |

14