Regulation of Stochastic System with Information Cost

Thesis submitted to

Indian Institute of Technology, Kharagpur

in partial fulfilment of the requirements

for the award of the degree of

Master of Technology

In

Visual Information and Embedded Systems Engineering

Submitted by –

Arnab Biswas

21EC65R01

Under the guidance of

Prof. Ritwik Kumar Layek

Department of Electronics and Electrical Communication Engineering

Indian Institute of Technology Kharagpur

Kharagpur, West Bengal, India – 721 302

Academic Year 2022-23

Certificate

This is to certify that the Thesis entitled "Regulation of Stochastic System with Information Cost" submitted by Arnab Biswas (21EC65R01) to Indian Institute of Technology, Kharagpur, India, is a record of bonafide Thesis work that has been done by him under my supervision and guidance in order to get considered for the award of degree of Master of Technology in the Department of Electronics and Electrical Communication Engineering with Specialization in Visual Information and Embedded Systems Engineering.

Date: **Prof. Ritwik Kumar Layek**

(Project Guide)

Department of E & ECE, IIT Kharagpur

Acknowledgement

I would like to thank my project guide and mentor, Prof. Ritwik Kumar Layek for his constant guidance and valuable feedbacks during the course of the project. Your suggestions and ideas were extremely helpful to carry out the thesis work properly. Your constant support and motivation kept me going throughout the semester and reach at this point.

I would like to also thank my friends, my colleagues and the respected seniors for their support and helping hands whenever I needed the most.

Arnab Biswas

21EC65R01

M.Tech (VIPES)

Indian Institute of Technology, Kharagpur

Abstract

The study of information transfer and processing in complex systems has been a topic of interest in various fields. The ability to understand how information flows within these systems has implications for many applications, including communication networks, biological systems, and financial markets. Different approaches have been proposed to measure and analyze information transfer, including Transfer Entropy, Symbolic Transfer Entropy, and Log-Likelihood Ratio. However, there is still debate on the suitability of these methods for complex systems, and there is a need for further exploration and development. Furthermore, the concept of information flow raises important questions about causality and the relationship between the past, present, and future. These questions have led to the development of theories such as Stochastic Optimal Control Theory and Feedback Capacity of Information Theory, which explore the duality between control and information transfer. Overall, a deeper understanding of information transfer and processing in complex systems can have significant implications for various fields, and further research in this area is necessary.

Contents

Chapter 1

Introduction:

In complex systems, information transfer and processing are crucial for understanding how the system functions. Information transfer refers to the movement of information between different components or subsystems of the system, while information processing refers to the transformation of information within a subsystem. Understanding how information is transferred and processed is essential for predicting and controlling the behavior of complex systems.

One approach for analyzing information transfer in complex systems is through information theory. This involves quantifying the amount of information transferred between subsystems using measures such as transfer entropy or mutual information. These measures can reveal the strength and direction of information flow and help identify which subsystems are more influential in the overall system behavior.

Another approach is through control theory, which involves designing control strategies that influence the behavior of the system. Control strategies can be based on optimizing certain objectives or minimizing certain costs, and can be implemented through feedback mechanisms that adjust the system's state or inputs based on its outputs. By understanding the relationship between information transfer and control, one can design more effective control strategies for complex systems.

A third approach is through symbolic dynamics, which involves representing the behavior of the system using discrete symbols or states. This can simplify the analysis of information transfer and processing by reducing the complexity of the system dynamics. Symbolic transfer entropy is a measure that quantifies the information flow between different symbolic states and can reveal which states are more informative for predicting future behavior.

Despite the usefulness of these approaches, there are also challenges and limitations. For example, measuring information transfer and processing can be difficult in practice, especially in systems with high-dimensional or continuous state spaces. Additionally, different measures may be more appropriate for different types of systems or information processing tasks. Finally, it is important to consider the trade-offs between information transfer and control when designing control strategies, as increasing information transfer may also increase system instability or unpredictability.

Overall, understanding how information is transferred and processed in complex systems is a challenging and important problem in many fields, including physics, engineering, biology, and social sciences. This thesis discussed in this context provide different perspectives and techniques for analyzing and controlling information transfer, highlighting the need for interdisciplinary approaches and collaborations to tackle this complex problem.

• Linear Quadratic Gaussian:

Linear Quadratic Gaussian (LQG) control is a mathematical framework used for designing optimal control systems. It combines the Kalman filter and Linear Quadratic Regulator (LQR) control techniques to form a complete control system. The LQR control algorithm minimizes the expected value of a quadratic cost function by adjusting the control input. It uses a state feedback approach to control the system by directly measuring the system's state variables. On the other hand, the Kalman filter is an estimator that uses noisy measurements of the system's output and input to estimate the system's state variables. The LQG control combines the Kalman filter's state estimation capabilities and LQR control's optimal control strategy to achieve a complete control system that is robust to disturbances and noisy measurements.

In an LQG control system, the controller receives noisy measurements of the system's output and input, and the Kalman filter is used to estimate the current state of the system. The LQR control algorithm then uses this estimated state to calculate the optimal control input that minimizes the expected value of the quadratic cost function. The LQG control system is designed to minimize the error between the desired and actual system response by adjusting the control input based on the current state estimate.

LQG control is widely used in various fields, including aerospace, robotics, and finance. It is particularly useful in situations where the system is subject to external disturbances or uncertainties in the measurements. The LQG control can adapt to these changes and provide optimal control inputs to maintain the desired system response. However, designing an LQG control system can be challenging, as it requires knowledge of the system's mathematical model and accurate measurement of the system's state variables. Additionally, the LQG control system's performance heavily relies on the choice of the cost function, and choosing an appropriate cost function can be a challenging task.

In the paper "Information Transfer of Control Strategies: Dualities of Stochastic Optimal Control Theory and Feedback Capacity of Information Theory", the linear quadratic Gaussian (LQG) control theory is utilized to investigate the transfer of information between different elements in a complex system. The LQG framework is used to model the control strategies used by individual agents in the system, as well as the overall behavior of the system as a whole.

The LQG approach is based on the assumption that the system can be described by a set of linear stochastic differential equations. The objective is to find an optimal control policy that minimizes the expected value of a cost function that captures the system's performance. The LQG framework combines a quadratic cost function with Gaussian noise models for both the system dynamics and the observation process.

The paper uses LQG control theory to explore the duality between stochastic optimal control theory and information theory. It shows that the optimal control strategies derived from the LQG framework can be interpreted as solutions to an information theoretic optimization problem. Specifically, the optimal control policy is shown to maximize the mutual information between the observed system state and the desired output.

The paper also investigates the relationship between the information transfer capacity of the system and the control strategies employed by individual agents. It shows that the information transfer capacity can be quantified using the concept of feedback capacity from information theory. The feedback capacity measures the maximum rate at which information can be transmitted from the output of the system to the individual agents controlling the system.

Overall, the paper highlights the role of LQG control theory in understanding the transfer of information in complex systems. It demonstrates the close connection between stochastic optimal control theory and information theory, and shows how the LQG framework can be used to model and analyze the control strategies employed by individual agents in a system, as well as the overall behavior of the system as a whole.

- ## Linear Quadratic Regulator:

The LQR algorithm is a type of optimal control algorithm that seeks to minimize a quadratic cost function subject to certain constraints. The quadratic cost function typically includes the squared error between the system's states and desired setpoints, as well as the squared control effort required to achieve these setpoints. The LQR algorithm finds a feedback control law that minimizes the cost function by computing the optimal control input as a function of the system's current state.

To compute the feedback control law, the LQR algorithm requires a state-space model of the system, which is a set of first-order differential equations that describe the evolution of the system's states over time. The LQR algorithm also requires weighting matrices that specify the relative importance of the states and control inputs in the cost function. The weighting matrices can be adjusted to tradeoff between control effort and state tracking performance.

Once the feedback control law is computed, it is applied to the system as a closed-loop control system. The closed-loop system uses feedback from the system's states to compute the optimal control input at each time step. The closed-loop system is designed to stabilize the system and optimize its performance.

LQR is widely used in control systems engineering because it provides a systematic way to design optimal control laws for linear time-invariant systems. LQR can be combined with other control and estimation techniques to improve control performance and stability. For example, LQR can be used in conjunction with state estimation techniques such as Kalman filtering to estimate the system's states and improve control performance in the presence of noise and uncertainty.

- ## Kalman Filter:

The Kalman filter is an algorithm used in control and estimation theory to estimate the state of a system based on noisy measurements. It is a recursive algorithm that provides an optimal estimate of the state of a system in the presence of noise and uncertainty. The Kalman filter is widely used in applications such as guidance and navigation systems, robotics, and signal processing.

The Kalman filter works by combining a predicted state estimate with a measurement update to generate an optimal estimate of the true state of the system. The predicted state estimate is based on the system's state dynamics and the control inputs, while the measurement update incorporates measurements of the system's states with a certain level of noise and uncertainty. The Kalman filter uses statistical information about the noise and uncertainty in the system to adjust the weighting between the predicted state estimate and the measurement update.

The Kalman filter is a recursive algorithm, meaning that it continuously updates the state estimate as new measurements become available. This allows the algorithm to adapt to changes in the system and improve the accuracy of the state estimate over time. The Kalman filter is particularly useful in applications where the true state of the system is unknown and measurements are noisy or incomplete.

Overall, the Kalman filter is a powerful algorithm that provides an optimal estimate of the state of a system in the presence of noise and uncertainty. It is widely used in applications such as guidance and navigation systems, robotics, and signal processing, and has been a key technology in many advancements in these fields.

• Bellman Filtering for State-space Model:

The Bellman equation is a fundamental principle in dynamic programming and reinforcement learning that expresses the optimal value function of a system as a recursive function of the optimal value function of its successor states. In the context of state-space models, the optimal value function represents the expected cumulative reward that can be obtained by following the optimal policy from any given state.

Bellman filtering combines the Bellman equation with the Kalman filter to estimate the state of a linear dynamical system and compute the optimal control policy. The algorithm estimates the optimal value function by computing a linear combination of the system's states and control inputs using the Kalman filter. The estimated value function is then used to compute the optimal control policy and update the state estimate.

One potential advantage of Bellman filtering is that it can improve the accuracy and efficiency of state estimation, especially in the presence of high process noise or measurement noise. By incorporating the Bellman equation into the filtering process, the algorithm can make use of information about the structure of the system and the optimal control policy, which can improve the accuracy of the state estimate.

Bellman filtering also has potential applications in fields such as robotics, control systems engineering, and finance. For example, it could be used to estimate the state of a robot or autonomous vehicle or to optimize the control of a financial portfolio.

Overall, Bellman filtering is a valuable technique for solving optimal control and decision-making problems in the context of state-space models, and it has the potential to improve the accuracy and efficiency of state estimation in a variety of applications.

- **Bertsekas filter:**

The Bertsekas filter is a recursive algorithm for state estimation that was introduced by Dimitri Bertsekas. The algorithm is based on the principles of stochastic optimal control and Kalman filtering, but it differs from the Kalman filter in several key respects.

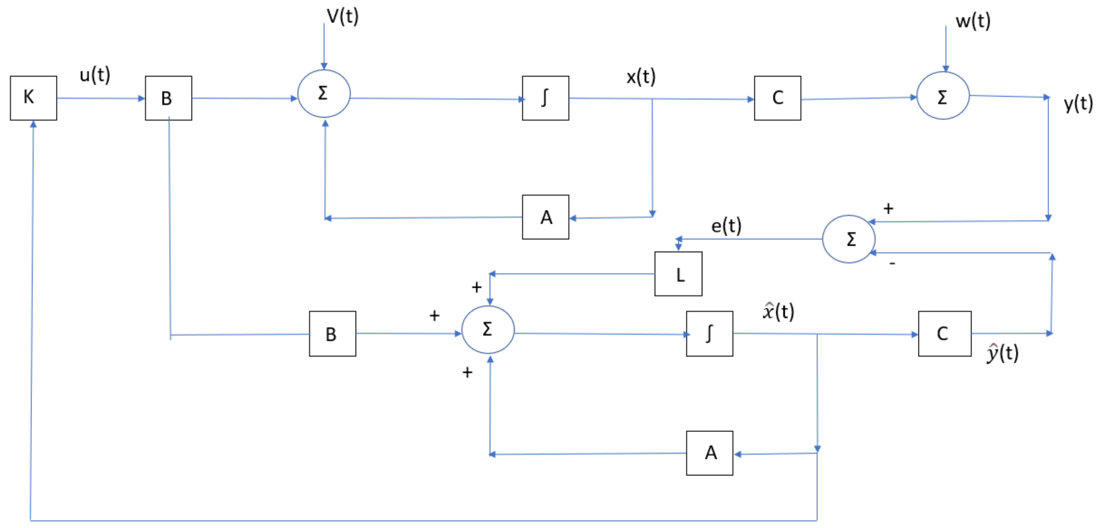
The Bertsekas filter is a nonlinear optimization approach that estimates the state of a dynamic system from noisy measurements. It does not assume that the noise in the system is Gaussian, which makes it suitable for a wide range of applications. Instead, the Bertsekas filter models the noise using a set of unknown parameters that are estimated along with the system state.

The Bertsekas filter uses a cost function that measures the difference between the predicted and measured states of the system. The cost function is optimized using a gradient descent approach, which adjusts the estimates of the state and noise parameters to minimize the cost.

One of the key advantages of the Bertsekas filter is its ability to handle nonlinear and non-Gaussian systems. The algorithm can be applied to both continuous-time and discrete-time systems, and it can be used to estimate the state of systems with constraints on the state or control inputs.

The Bertsekas filter has been used in a variety of applications, including control engineering, robotics, and finance. In control engineering, the algorithm is used to estimate the state of a system based on noisy sensor measurements, and it can be used to design feedback control systems that can adapt to changing environmental conditions. In robotics, the Bertsekas filter is used to estimate the position and orientation of a robot based on sensor data, and it is often used in combination with other algorithms for motion planning and control.

Overall, the Bertsekas filter is a powerful and flexible algorithm for state estimation in dynamic systems, and it is widely used in a variety of applications. Its ability to handle nonlinear and non-Gaussian systems makes it particularly useful in settings where traditional linear filters such as the Kalman filter are not applicable.



Within the scope of this diagram, we have designed a Linear Quadratic Gaussian together with a Linear Quadratic Regulator and a Luenberger observer. Our goal is to reduce the value of e(t) to the greatest extent possible by modifying the value of $\hat{y}(t)$. We are unable to make any adjustments to the parameters that regulate the operation of our plant or system since it functions like a black box. Instead of that, we have the ability to alter the values of the parameters that make up the Luenberger Observer. The Luenberger observer is an algorithm that makes its predictions about the behaviour of a system's internal state over a period of time by employing the mathematical model of the system. These forecasts are evaluated in light of the results of the measurements actually taken of the system's output. The subsequent step is to change the estimated state variables based on the deviation that was found between the predicted and actual output. The process of modifying the variables that are approximated to represent the state is referred to as feedback. To achieve a more accurate estimation of the status of the system over the course of time, the feedback loop is continually iterated. In this step, we determine our expected output y(t) by taking the feedback from the Luenberger Observer and feeding it into the system using a Linear Quadratic Regulator.

In the diagram, A, B, C are the blocks whose identification has been done. V(t) and w(t) are the noise signal, K block is working as Linear Quadratic Regulator and L block is working as gain block.

From the figure, we can write the equations,

$y(t) = w(t) + C * x(t)$

$x(t) = v(t-1) + B * u(t-1) + A * x(t-1)$

$e(t) = y(t) - \hat{y}(t)$

$\hat{x}(t) = A * \hat{x}(t-1) + B * u(t-1) + L * e(t)$

$\hat{y}(t) = C * \hat{x}(t)$

$u(t) = K * \hat{x}(t)$

From these fundamental equations, we can derive the expression for $e(t)$,

$y(t) = w(t) + C * x(t)$

$\Rightarrow y(t) = w(t) + C * [v(t-1) + B * u(t-1) + A * x(t-1)]$

$\Rightarrow y(t) = w(t) + C * [v(t-1) + B * K * \hat{x}(t-2) + A * x(t-1)]$

Again,

$\hat{y}(t) = C * \hat{x}(t)$

$\Rightarrow \hat{y}(t) = C * [A * \hat{x}(t-1) + B * u(t-1) + L * e(t)]$

$\Rightarrow \hat{y}(t) = C * [A * \hat{x}(t-1) + B * K * \hat{x}(t-2) + L * e(t)]$

So,

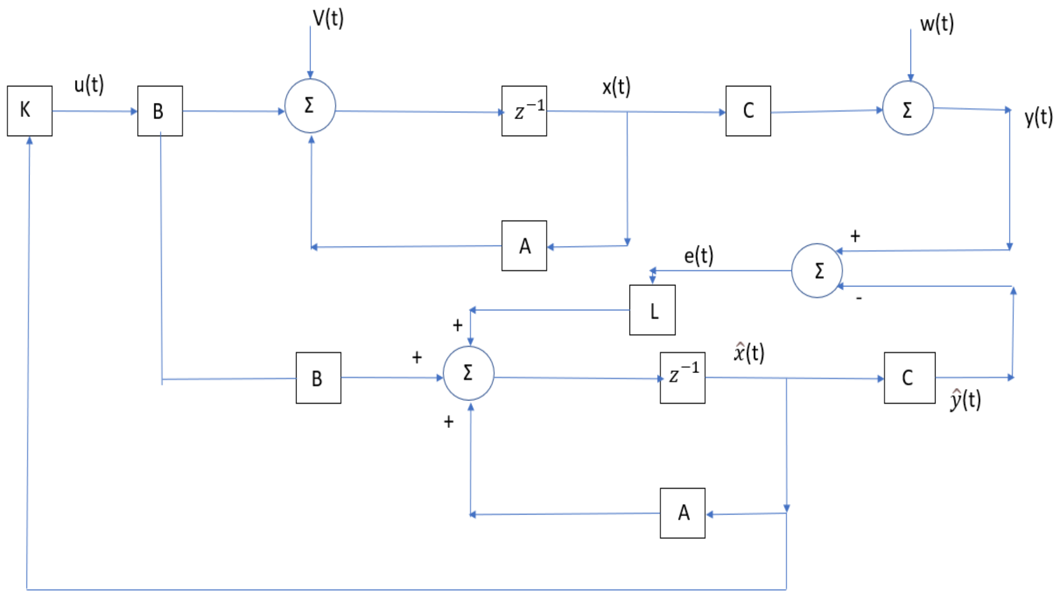
$e(t) = y(t) - \hat{y}(t)$

$\Rightarrow e(t) = w(t) + C * [v(t) + B * K * \hat{x}(t-2) + A * x(t-1)] - C *$
$[A * \hat{x}(t-1) + B * K * \hat{x}(t-2) + L * e(t)]$

$\Rightarrow e(t) = w(t) + C * [v(t-1) + A * x(t-1) - A * \hat{x}(t-1) - L * e(t)]$

$\Rightarrow e(t) * [1 + C * L] = w(t) + C * [v(t-1) + A * x(t-1) - A * \hat{x}(t-1)]$

$$\Rightarrow e(t) = \frac{w(t) + C * [v(t-1) + A * x(t-1) - A * \hat{x}(t-1)]}{1 + C * L}$$



In this figure, we have designed the same block diagram for distinct time series. Here, instead of the integrator block, we have used the delay block.

From the figure, we can write the equations,

$$y_k = w_k + C * x_k$$

$$x_k = v_{k-1} + B * u_{k-1} + A * x_{k-1}$$

$$e_k = y_k - \hat{y}_k$$

$$\hat{x}_k = A * \hat{x}_{k-1} + B * u_{k-1} + L * e_k$$

$$\hat{y}_k = C * \hat{x}_k$$

$$u_k = K * \hat{x}_k$$

From these fundamental equations, we can derive the expression for $e(t)$,

$$y_k = w_k + C * x_k$$

$$\Rightarrow y_k = w_k + C * [v_{k-1} + B * u_{k-1} + A * x_{k-1}]$$

$$\Rightarrow y_k = w_k + C * [v_{k-1} + B * K * \hat{x}_{k-2} + A * x_{k-1}]$$

Again,

$$\hat{y}_k = C * \hat{x}_k$$

$$\Rightarrow \hat{y}_k = C * [A * \hat{x}_{k-1} + B * u_{k-1} + L * e_k]$$

$\Rightarrow \hat{y}_k = C * [A * \hat{x}_{k-1} + B * K * \hat{x}_{k-2} + L * e_k]$

So,

$e_k = y_k - \hat{y}_k$

$\Rightarrow e_k = w_k + C * [v_{k-1} + B * K * \hat{x}_{k-2} + A * x_{k-1}] - C * [A * \hat{x}_{k-1} + B * K * \hat{x}_{k-2} + L * e_k]$

$\Rightarrow e_k = w_k + C * [v_{k-1} + A * x_{k-1} - A * \hat{x}_{k-1} - L * e_k]$

$\Rightarrow e_k * [1 + C * L] = w_k + C * [v_{k-1} + A * x_{k-1} - A * \hat{x}_{k-1}]$

$\Rightarrow e_k = \dfrac{w_k + C * [v_{k-1} + A * x_{k-1} - A * \hat{x}_{k-1}]}{1 + C * L}$

- **Luenberger Observer:**

A Luenberger observer is a mathematical algorithm used to estimate the internal state of a dynamic system based on its input and output signals. The state of a system is a set of internal variables that describe the behavior of the system over time. In many practical applications, the state of a system is not directly measurable or observable. Instead, only measurements of the system's input and output are available. The Luenberger observer is designed to estimate the system state based on the available measurements and a mathematical model of the system.

The Luenberger observer works by using the mathematical model of the system to predict the behavior of the system's internal state over time. These predictions are compared to the actual measurements of the system's output. The difference between the predicted and actual output is then used to adjust the estimated state variables. The process of adjusting the estimated state variables is known as feedback. The feedback loop is repeated continuously to improve the estimate of the system's state over time.

The Luenberger observer is named after Rudolf Luenberger, who introduced the concept in the 1960s. It is widely used in control systems engineering, particularly in applications where the state of the system is not directly observable. The Luenberger observer is known for its simplicity, ease of implementation, and ability to estimate the states of systems with complex dynamics.

One of the key advantages of the Luenberger observer is its ability to provide real-time estimates of the system state. This is particularly useful in applications where accurate and timely state information is critical for making decisions or controlling the system. For example, in the aerospace industry, Luenberger observers are used to estimate the position and velocity of aircraft based on sensor measurements. In the automotive industry, Luenberger observers are used to estimate the speed and position of vehicles based on sensor measurements.

The Luenberger observer is also valuable in cases where the system being observed is subject to noise or disturbances. The observer can filter out the noise and disturbances to provide a more accurate estimate of the system state. The observer can also be designed to provide robustness to uncertainties in the system model or parameter values.

Overall, the Luenberger observer is a powerful tool for estimating the state of a dynamic system in real-time, even in cases where the state variables are not directly observable. It has found numerous applications in a variety of fields, including aerospace, robotics, and automotive control systems.

Chapter 2

• Literature Review:

This section is dedicated to a literature survey on Regulation of Stochastic System. It explains the previous work in the Regulation of Stochastic System, which has helped and inspired me during the project.

The paper "Information Transfer of Control Strategies: Dualities of Stochastic Optimal Control Theory and Feedback Capacity of Information Theory" by Charalambos D. Charalambous explores the relationship between stochastic optimal control theory and information theory. Specifically, the paper shows a duality relationship between the optimal control policy and the feedback capacity of a communication channel.

Stochastic optimal control theory deals with the design of feedback control policies that minimize a certain cost function over a finite or infinite time horizon. The optimal control policy is typically obtained by solving the Bellman equation, which is a recursive equation that expresses the optimal cost-to-go function in terms of the optimal control law.

Information theory, on the other hand, deals with the fundamental limits of communication and data processing. It involves studying the amount of information that can be reliably transmitted over a noisy channel, and the rate at which information can be reliably compressed or transmitted over a communication channel.

The duality relationship between these two fields is based on the fact that the optimal control policy and the feedback capacity of a communication channel are related through a set of dual equations. Specifically, the paper shows that the optimal control policy can be expressed in terms of the feedback capacity of a communication channel, and vice versa.

To understand the duality relationship, it is helpful to consider the following scenario: Suppose we have a control system that receives an input signal u(t) and produces an output signal y(t), and suppose there is a noisy channel that transmits the output signal y(t) to a receiver. The receiver then feeds back a control signal u(t) to the control system. The goal is to design a feedback control policy that minimizes a certain cost function while taking into account the limitations of the communication channel.

The duality relationship between the optimal control policy and the feedback capacity of the communication channel arises because both the control system and the communication channel involve the transmission of information from one point to another. The optimal control policy involves transmitting information from the input signal to the output signal, while the communication channel involves transmitting information from the output signal to the receiver.

The paper shows that the feedback capacity of the communication channel can be expressed in terms of the optimal control policy, and vice versa. Specifically, the paper shows that the feedback capacity of the communication channel is equal to the mutual information between the input signal u(t) and the output signal y(t), while the optimal control policy is equal to the solution of the Bellman equation, which depends on the mutual information between the output signal y(t) and the state of the system x(t).

In summary, the paper shows that there is a duality relationship between stochastic optimal control theory and information theory, and provides a deeper understanding of the fundamental limits of control and communication systems.

Next Transfer Entropy is introduced to measure the amount of information transfer from one time series to another for a system consisting more than one component by Thomas Schreiber.

Entropy is a measure of the uncertainty or randomness of a system, and is commonly used in information theory to quantify the amount of information contained in a message or data set. The paper then introduces the concept of transfer entropy, which is a measure of the directed flow of information between two time series.

Transfer entropy is defined as the amount of information that is transferred from one time series to another, given the past history of both time series. Specifically, transfer entropy measures the reduction in uncertainty of the future of the target time series, given the past values of both the target and source time series, as compared to the uncertainty of the future of the target time series based on its own past values alone. The paper then presents the algorithm for estimating transfer entropy, which is based on a non-parametric approach that does not require any assumptions about the underlying distribution of the data. The algorithm works by dividing the data into time bins, and estimating the probability distributions for each bin using a nearest-neighbour approach. The transfer entropy is then calculated by comparing the joint probability distribution of the past and future values of the two-time series, with the probability distribution based on the past values of the target time series alone.

Transfer entropy is a measure of information flow between two time series data sets. It is based on the idea that if we know the past of one time series, we can predict the future of the other series more accurately than if we didn't know the past of the first series. Transfer entropy is typically formulated as a non-negative quantity that measures the amount of information that is transferred from one series to another.

The paper proposes an algorithm for measuring the transfer of information between two time series signals. Transfer Entropy (TE) algorithm is the name of the method. It is founded on the concept of conditional mutual information and employs a non-parametric estimator to calculate the information transfer between two signals. The TE algorithm is a potent tool for analysing causal interactions between different systems, and it has been implemented in numerous disciplines, such as neuroscience, economics, and physics. The transfer entropy algorithm can be summarized as follows:

1. Choose two time series signals X and Y with length N.
2. Divide the time series signals into M segments of length l, where M = N - l + 1.

3. For each segment i, calculate the probability distribution of X(t) and Y(t) for t = i+l-1.
4. For each segment i, calculate the joint probability distribution of X(t) and Y(t-1) for t = i+l-1.
5. For each segment i, calculate the joint probability distribution of X(t) and Y(t-1) conditioned on the past values of Y, Y(t-2), Y(t-3),..., Y(t-l).
6. For each segment i, calculate the transfer entropy from Y to X using the formula:

   TE(Y->X) = sum[P(X(t), Y(t-1), Y(t-2),..., Y(t-l)) * log(P(X(t) | Y(t-1), Y(t-2),..., Y(t-l)) / P(X(t) | Y(t-1), Y(t-2),..., Y(t-l), X(t-1)))]

   where the sum is taken over all segments i, and P(X|Y) denotes the probability distribution of X conditioned on Y.
7. Repeat steps 3-6 for the reversed direction of information flow, i.e., from X to Y.
8. Calculate the total information transfer from Y to X as:

   IT(Y->X) = TE(Y->X) - TE(X->Y) and the total information transfer from X to Y as:

   IT(X->Y) = TE(X->Y) - TE(Y->X)

In the paper Transfer Entropy as a Log-Likelihood Ratio, the author argues that transfer entropy can also be interpreted as a log-likelihood ratio between two statistical models that describe the joint distribution of the two-time series. Specifically, the author shows that transfer entropy can be written as the difference between the log-likelihood of a model that includes both time series and the log-likelihood of a model that includes only the past of the receiving time series.

This new interpretation of transfer entropy has several benefits. First, it provides a more intuitive understanding of the measure and its relationship to statistical models. Second, it allows for the use of established statistical techniques for hypothesis testing and model selection. Finally, it opens up the possibility of extending transfer entropy to more complex models, such as those involving nonlinear or non-Markovian dependencies.

The paper includes a number of theoretical results and examples to illustrate the use of transfer entropy as a log-likelihood ratio. The author also discusses the implications of this interpretation for the interpretation and use of transfer entropy in various scientific fields.

In this paper a new formulation of transfer entropy is proposed that uses the log-likelihood ratio to measure the dependence between two time series. This method is more computationally efficient and more accurate than traditional methods of transfer entropy calculation. The paper presents the theoretical framework of this method and shows its effectiveness in simulations and real-world applications.

The algorithm involves the following steps:

1. Define the source and target time series as X and Y, respectively.
2. Create a delayed embedding of the source and target time series.
3. Divide the embedded data into two sets: training and test sets.
4. Calculate the conditional entropy of the target variable given its past and the past of the source variable for both the training and test sets.
5. Calculate the conditional entropy of the target variable given only its past for both the training and test sets.
6. Calculate the transfer entropy as the log-likelihood ratio of the conditional entropies from steps 4 and 5.

The final equation for the transfer entropy is:

$$TE = H(Y(t)|Y(t-1), X(t-d)) - H(Y(t)|Y(t-1))$$

where TE is the transfer entropy, H is the conditional entropy, Y(t) is the current value of the target variable, Y(t-1) is the previous value of the target variable, and X(t-d) is the delayed value of the source variable.

In the paper Symbolic Transfer Entropy, the authors propose a new method for estimating transfer entropy that is based on symbolic dynamics. Symbolic dynamics is a mathematical technique for converting a continuous time series into a discrete sequence of symbols based on the behaviour of the system. The authors apply this technique to the time series data and then use the resulting symbolic sequences to estimate the transfer entropy.

The proposed method has several advantages over existing methods for estimating transfer entropy. First, it can handle high-dimensional and noisy data, as well as nonlinear dependencies between the time series. Second, it is computationally efficient, allowing for the analysis of large data sets.

The authors demonstrate the effectiveness of their method using simulations and real-world applications, such as analysing the relationship between the electrical activity of the brain and the heartbeat. They compare their method with existing methods and show that their method outperforms them in terms of accuracy and computational efficiency.

The paper presents a new method for estimating transfer entropy that is based on symbolic dynamics, which offers several advantages over existing methods.

There is an algorithm presented in the paper. The authors introduce a new approach to compute transfer entropy, called Symbolic Transfer Entropy (STE), that uses symbolic dynamics to discretize continuous data and computes the transfer entropy between discrete symbols.

Here is a brief summary of the STE algorithm:

1. Partition the time series data into m-dimensional phase space vectors
2. Use a symbolization scheme to convert each phase space vector into a discrete symbol
3. Compute the probability distribution of the current symbol and the next symbol for each pair of time series using a sliding window approach
4. Compute the transfer entropy between the time series by comparing the joint probability distribution of the current symbol and the next symbol for the pair of time series with the conditional probability distribution of the next symbol for the target time series given the current symbol for the source time series

5. Repeat steps 3-4 for multiple time lags and combine the results to obtain a time-lagged transfer entropy matrix.

But the transfer entropy does not give correct measurement every time of information flow between two time series data sets as discussed by Ryan G. James, Nix Barnett, and James P. Crutchfield.

The paper starts by introducing the concept of transfer entropy, which is a measure of the amount of information that is transferred from one time series to another. Transfer entropy is based on the idea that if we know the past of one time series, we can predict the future of the other series more accurately than if we didn't know the past of the first series. The authors explain the mathematical formulation of transfer entropy and how it can be used to measure information flow.

The authors then discuss the limitations of transfer entropy as a measure of information flow. One of the main issues is that transfer entropy assumes a linear causal relationship between the two-time series. However, in many real-world scenarios, the causal relationship may be nonlinear or more complex. The authors argue that transfer entropy may not be able to capture these complex causal relationships and may lead to incorrect conclusions about information flow.

Another limitation of transfer entropy is that it can be sensitive to the choice of time bin size used in the analysis. This means that different bin sizes can result in different conclusions about information flow, which can be problematic for data analysis.

The authors propose an alternative approach to measuring information flow based on the concept of causal states. Causal states are a way of characterizing the past history of a time series and can be used to identify the causal relationships between two time series. The authors argue that causal states can provide a more robust and accurate measure of information flow compared to transfer entropy.

- **Duality between Control Theory and Information theory:**

The concept of "dualities" in control theory refers to the observation that seemingly different control problems can be transformed into equivalent forms. This is the case with the duality between stochastic optimal control theory and information theory, which are two different fields of study, but share many mathematical similarities and can be used to solve similar problems.

In stochastic optimal control theory, the goal is to design a control strategy that minimizes a cost function while accounting for the stochastic (random) nature of the system's dynamics. This involves finding the optimal control policy that minimizes the expected value of the cost function over a given time horizon.

In information theory, the focus is on quantifying the amount of information that can be transmitted over a communication channel with limited bandwidth and noisy transmission. The feedback capacity of a communication channel is the maximum rate at which information can be transmitted with arbitrarily small error probability using feedback.

The duality between these two fields arises because the same mathematical framework can be used to solve both problems. Specifically, the Bellman equation, which is a central tool in stochastic optimal control theory, has a dual form known as the Hamilton-Jacobi equation, which arises in information theory.

Moreover, the optimal control policy in stochastic optimal control theory can be expressed in terms of the solution to the Hamilton-Jacobi equation, and this provides a way to solve the information-theoretic problem of maximizing the feedback capacity of a communication channel.

In summary, the duality between stochastic optimal control theory and information theory arises due to the mathematical similarities between the two fields and can be used to transfer control strategies between the two domains. This allows researchers to use tools from one field to solve problems in the other and can lead to new insights and approaches for solving complex control problems.

Chapter 3

- **Model:**

The dynamics of a system refer to the behaviour of the system over time, i.e., how it evolves and changes in response to internal and external influences. In particular, the dynamics of a system describe the way in which its variables and components interact and affect each other, leading to changes in the system's state or behaviour. This can involve complex patterns of feedback, nonlinearity, and stochasticity that can give rise to emergent phenomena and behaviour that cannot be predicted from the properties of the individual components alone. The study of system dynamics is an interdisciplinary field that draws on concepts and methods from physics, engineering, mathematics, computer science, biology, and other fields. It has applications in diverse areas such as control theory, robotics, ecology, economics, and social systems, among others. Overall, understanding the dynamics of a system is essential for predicting its behaviour and designing interventions or control strategies to achieve specific goals or outcomes.

Dynamics of Markov Chain:

The dynamics of a Markov chain refer to how the chain evolves over time. Specifically, the dynamics of a Markov chain are described by its transition matrix, which is a square matrix where each entry represents the probability of transitioning from one state to another in a single time step. The transition matrix captures the Markov property, which is the idea that the probability of moving from one state to another only depends on the current state and not on the past history of the chain.

The dynamics of a Markov chain are important because they determine the long-term behaviour of the chain. In particular, the dynamics of a Markov chain can exhibit a range of behaviours, including convergence to the stationary distribution, periodic or transient behaviour, and multiple stationary distributions in the case of reducible chains.

Convergence to the stationary distribution is the most common behaviour of a Markov chain. The stationary distribution represents the long-term behaviour of the chain as it approaches equilibrium, and it describes the probabilities of the chain being in each state after a long time has elapsed. For an irreducible and aperiodic Markov chain, there exists a unique stationary distribution that the chain approaches as time goes to infinity.

Periodic or transient behaviour occurs when the chain visits a subset of states repeatedly without ever visiting some other states. This type of behaviour is typically observed in Markov chains with a small number of states.

Multiple stationary distributions can occur in the case of reducible chains, where the chain can be decomposed into multiple disjoint subsets of states. In this case, there can be multiple stationary distributions, each corresponding to a different subset of states.

In summary, the dynamics of a Markov chain are important for understanding the behaviour of the chain over time. By analysing the transition matrix, we can determine the long-term behaviour of the chain, including whether it converges to a unique stationary distribution, exhibits periodic or transient behaviour, or has multiple stationary distributions.

Dynamics of the system when control is added to the system

The addition of control to a system can be a powerful tool for manipulating and shaping the system's behaviour in a desired way. Control can be used to achieve a variety of objectives, such as regulating the output of the system to a desired value, tracking a reference trajectory, or suppressing unwanted disturbances.

The dynamics of a controlled system are characterized by the relationship between its inputs, outputs, and internal states. The inputs to the system represent the control signals that we can manipulate, and the outputs represent the observable behaviour of the system. The internal states are the intermediate variables or parameters that govern the evolution of the system over time.

To design a control system, we first need to develop a mathematical model of the system, which describes the relationship between its inputs, outputs, and internal states. This model can be based on physical laws, empirical data, or a combination of both.

Once we have a model of the system, we can design a controller that generates the control signals based on the system's current state and the desired control objectives. There are many different types of controllers, including proportional-integral-derivative (PID) controllers, model-based controllers, and adaptive controllers, among others.

After designing the controller, we need to analyse the stability and performance of the closed-loop system, which includes both the system and the controller. Stability analysis involves determining whether the closed-loop system will remain stable over time, while performance analysis involves evaluating how well the closed-loop system achieves its control objectives.

Control theory provides a rich set of mathematical tools and techniques for designing and analysing control systems. These tools include stability analysis, frequency response analysis, optimization, and robust control, among others.

Within the scope of this project, we are tasked with designing a Markov Chain and attempting to comprehend its dynamics in the context of a controlled system. In this case, we have designed the system in such a way that if there are any undesirable states in the state space, then by utilising the control system, we are able to transform these undesirable states to the desired state at a cost that is as low as possible. A stochastic process is said to be a Markov chain if it satisfies the Markov property. This property asserts that the future state of the system depends only on the current state and not on any of the states that have come before it. To put it another way, the Markov chain does not retain any memory of its previous states, and the current state of the chain entirely governs how it will develop in the future. A Markov chain is characterised by a collection of states and a transition probability matrix that describes the likelihood of transitioning from one state to another in a single time step. The set of states and the transition probability matrix are what define a Markov chain. The probabilities of transition are simply determined by the condition the system is in right now; they do not take into account any of its previous states. For the purposes of this project, I have presumed that there are a total of 16 states. The dimensions of the transition matrix are therefore 16 × 64. I have used the semi tensor product of the matrix here, and we have divided the transition matrix into four pieces, with the dimension of each matrix being 16 × 16 squares. Here, we make an attempt to modify the probabilities that are already included in the transition matrix in such a way that anytime any state transitions to an undesirable state, the current state transitions promptly to any preferred state at the lowest possible cost.

When a Markov chain is under control, the dynamics of the system are modified by the addition of external influences that can affect the transition probabilities between the states. Control can be exerted on a Markov chain in a variety of ways, such as by applying feedback, by changing the environment, or by manipulating the system parameters.

In the context of control theory, the dynamics of a Markov chain under control can be described using a controlled Markov chain, which is a stochastic process that evolves over time according to a set of transition probabilities that depend on both the current state and the control action.

The objective of control in a Markov chain is typically to steer the system towards a desired state or to optimize some performance criterion. This can be achieved by feedback control.

Analysing the dynamics of a Markov chain under control typically involves solving a set of differential or difference equations that describe the evolution of the system over time. This can be challenging, especially for complex systems with many states and control inputs.

Chapter 4

• Simulation:

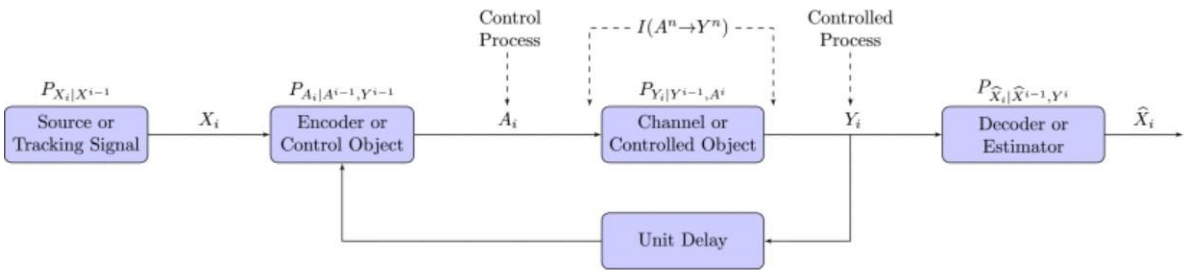
In this section, we try to find the outputs and training details of the existing method from Shannon's communication block model. We try to understand the training details of the control object, controlled object and estimator. We run the code on Google Colab to familiarize ourselves with the outputs. We try to find the outputs with deterministic and randomized input values of the transition matrix.

• Outputs of Controlled Object:

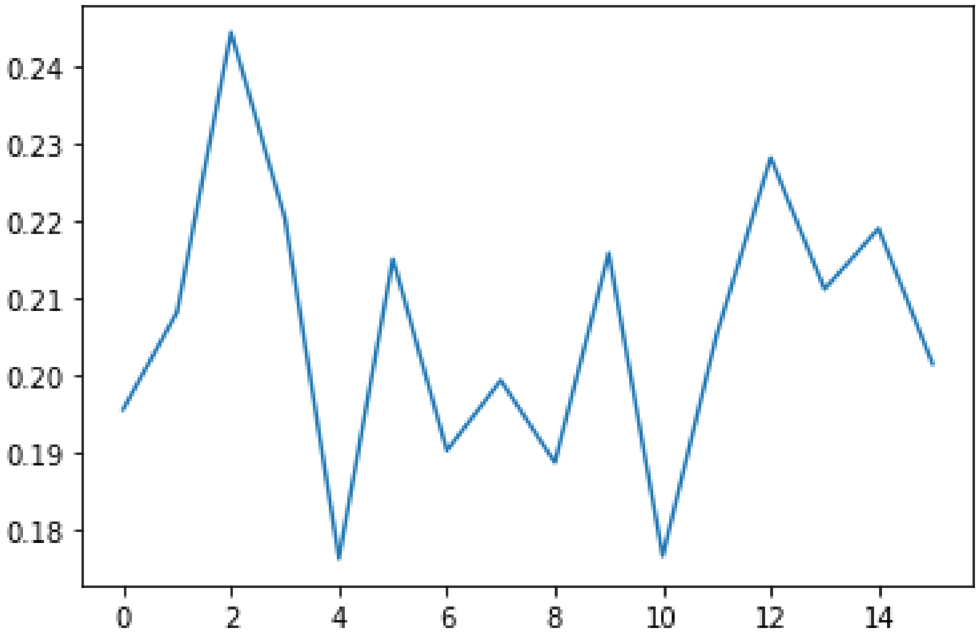
Here, we have designed the channel or Controlled object of Shannon's Communication block. To design the block, we have assumed that the output signal Y_i is of two bits and the control signal A_i is of four bits. The probability of the controlled object can be found using the formula $P_Y i|Y^{i-1}, A^i$, which can be written as

$$\frac{P(Y_i \cap (Y^{i-1}, A^i))}{P(Y^{i-1}, A^i)}$$

The information-transfer-directed information payoff functional from the control process to the controlled process is maximized when stochastic optimum control problems are taken into account. We draw a clear comparison between the feedback capacity of information theory and stochastic optimal control theory, as depicted in Figure. By using the operational notion of control-coding (CC) capacity of the control system, this analogy suggests that every control or decision problem is capable of information transmission from control processes to controlled processes, or other processes.



Shannon's Communication Block Diagram



Probability of output after giving control input

In the first experiment, we try to find how the probability of the output is changed when a control signal is applied to the input. We have used the semi-tensor product of matrix, wherewe divide the transition matrix of dimension 16x64 to four same type of matrices of dimension 16x16.

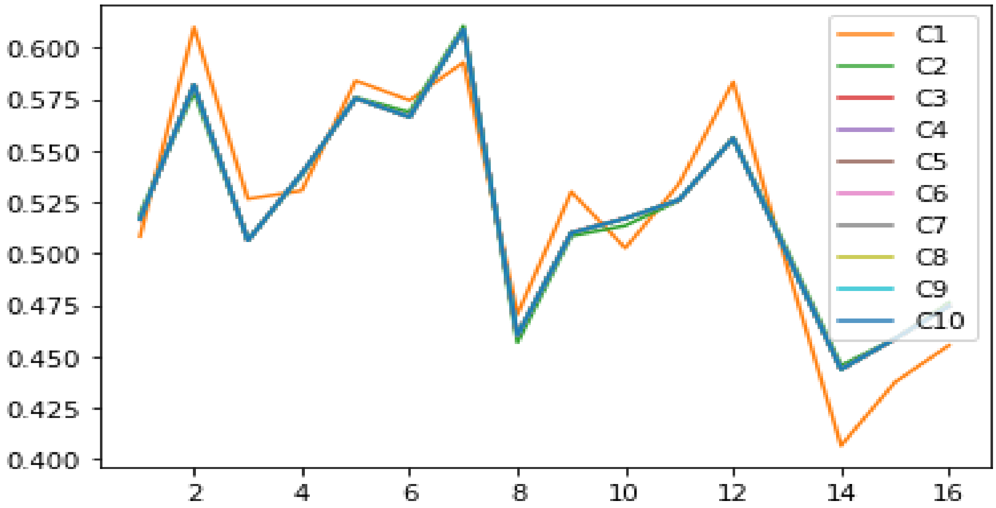
Assume, the transition matrix is H of dimension 16x64, A_i is the control signal and Y_i is the controlled output. Then,

$Y_i = HA_iY_{i-1}$

$= (a_0H_0 + a_1H_1 + a_3H_3 + a_4H_4)Y_{i-1}$,

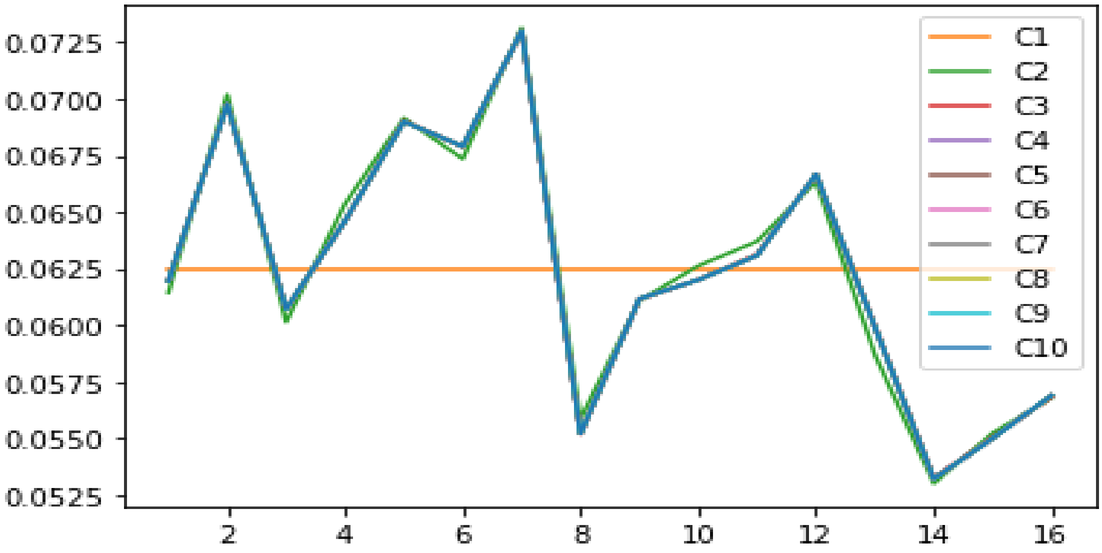
Where, H_0, H_1, H_2, H_3 are the component of the transition matrix H, each of dimension 16x16. The constraints here is the sum of all the elements of the row of each of the matrix should be 1. As we take the control signal of four bits, a_0, a_1, a_2, a_3 represent the control signal where $a_0 + a_1 + a_2 + a_3 = 1$.

In the next experiment, we use the same process to get the stabilized output. At first, we take random values to the input signal and find the values of the transition matrix using the semi-tensor product. We have generated the outputs of ten successive values of Y. Thus, in the graph, we can see at first, the output values fluctuated and at last, the outputsare overlapped. So, we can conclude that the outputs become stabilized.



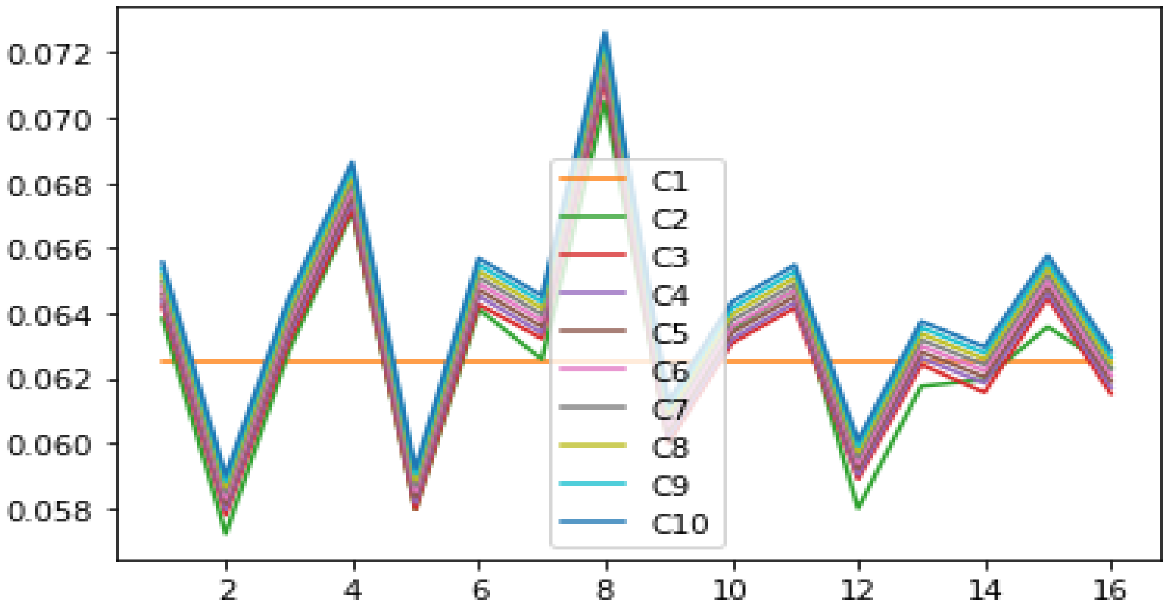
Probability of stabilized outputs after ten iterations

In the next experiment, we use a constant input, where the value of each element is 0.0625. We take the value 0.0625 so that the sum of all sixteen elements of this matrix is equal to 1. Here, we try to find the value of the output-controlled signal after ten iterations. So, in thegraph, we can see that at first, we get a straight line of value 0.0625 and from the next output, we get the fluctuated output. At last, we get overlapped stabilized output.



Probability of stabilized output with constant input

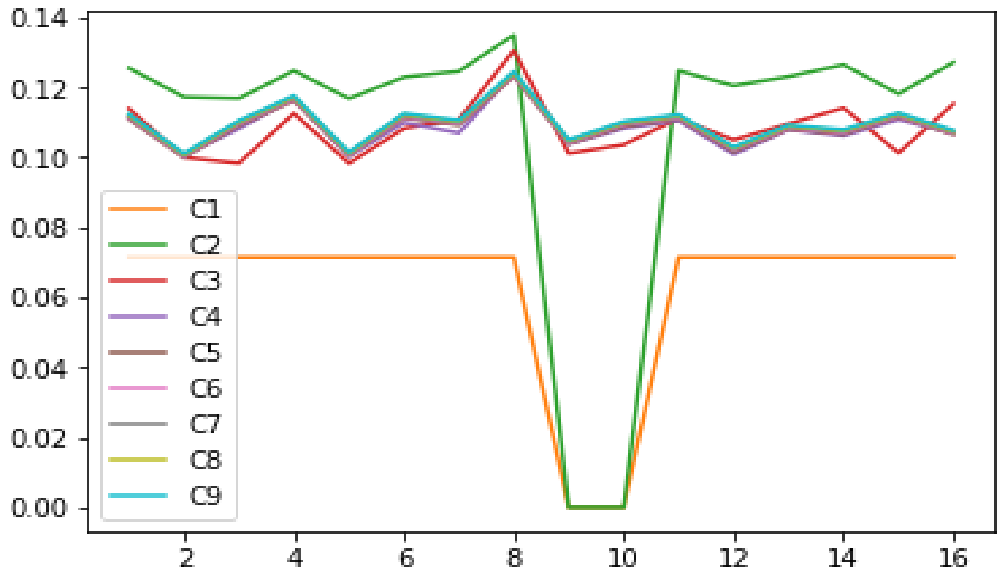
In the next experiment, we use a deterministic transform function instead of randomly generated transform function. The significance of this step is in the output, we canunderstand how the outputs are changed in every step.



Probability of stabilized output with deterministic transform function

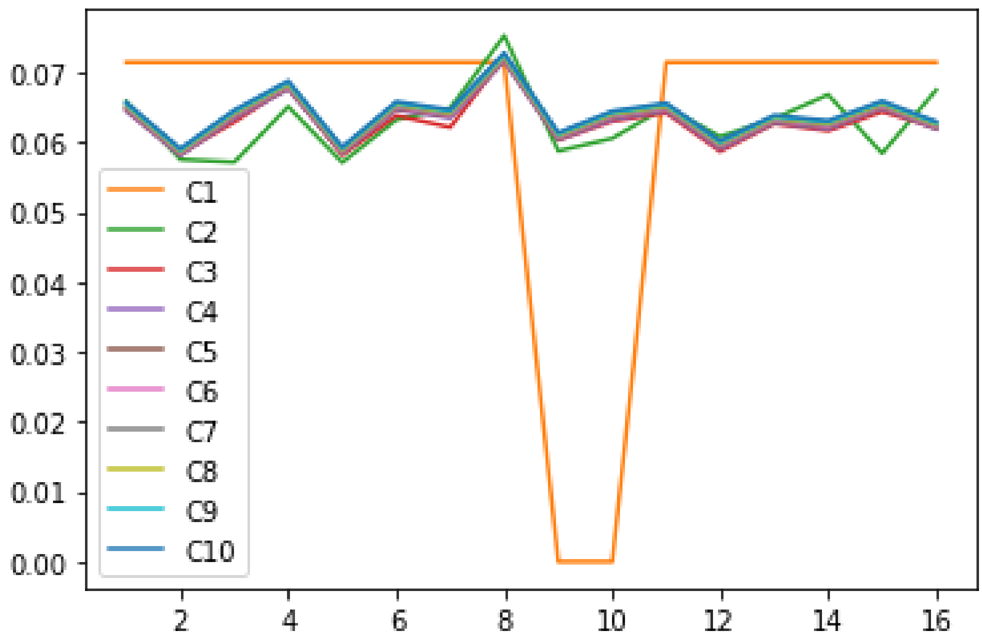
In the next experiment, we change the value of the input matrix. Here, we change the 9^{th} and 10^{th} elements of the matrix to zero and change the other values to

0.07142857 so that the sum of all values is 1. Our intention is to check when the value of the 9^{th} and 10^{th} elementsof the matrix becomes greater than 0.1 so that we can reduce the values of the elements to zero again and the sum of these two elements is evenly distributed among all other elements of the matrix. From the output, we can see that the values of the 9^{th} and 10^{th} elements of the matrix become zero at the 3^{rd} iteration. So, after the 3^{rd} iteration, we distributed the sum of the values of 9^{th} and 10^{th} elements to all the elements of the matrix.



Probability of stabilized output with controlled inputs

In the next experiment, we try to find out the output where, we change the values of the 9^{th} and 10^{th} element to zero if the values of these elements are greater than 0.1 for five times continuously. From the graph plot, we can see that this condition never met.



Probability of stabilized output with periodic controlled input

Chapter 5

- ## Conclusion:

The thesis has provided a comprehensive review of the concept of information transfer in complex systems. Through the analysis of existing literature, we have gained an understanding of the theoretical foundations of information transfer and the different methods used to measure and analyse it.

In the first part, we have seen that the dynamics of a system play a crucial role in information transfer, and that the use of appropriate methods for analysing information transfer is necessary for accurate results. Despite the challenges and limitations of existing methods, information transfer analysis has significant potential for various applications, including in neuroscience, biology, engineering, and communication systems.

Through this thesis, we have also identified gaps in the current understanding of information transfer in complex systems, highlighting the need for further research in this area. By addressing these gaps, we can deepen our understanding of information transfer in complex systems and develop more effective strategies for controlling and optimizing these systems.

In the second part of the thesis, we make an effort to discover the results of these methodologies that have been produced by researchers. In addition to this, we make an effort to build systems in such a way that, in the event that an undesirable state is reached, the transition matrix can be modified with a minimum amount of effort using control algorithms.

Chapter 6

- **Future Scope:**

As we have seen throughout this thesis, the concept of information transfer in complex systems is a complex and multifaceted topic. Our review has highlighted the challenges and limitations of existing methods for measuring and analysing information transfer, as well as the potential of these methods for understanding and controlling complex systems.

We have seen that the dynamics of a system play a critical role in information transfer, and that understanding these dynamics is essential for accurate measurement and analysis of information transfer. We have also seen that there are many different methods for measuring information transfer, each with its own strengths and limitations.

Despite these challenges, our review has highlighted the potential of information transfer analysis for a wide range of applications, including in neuroscience, biology, engineering, and communication systems. By gaining a better understanding of information transfer in these domains, we can develop more effective strategies for controlling and optimizing complex systems.

Looking forward, there is a need for further research in this area, both to develop new methods for measuring and analysing information transfer and to apply these methods to a wider range of domains. With continued research and development, we can gain a deeper understanding of the fundamental principles underlying information transfer in complex systems and develop more effective strategies for controlling and optimizing these systems.

Appendix:

In this section, we have provided the code which we have used throughout the experiment. The codes are written in Google Colab using python language. Also, we have used the Numpy and Matplotlib library. Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python and NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.

1. Implementation of Controlled Object:

```
import numpy as np
from matplotlib import pyplot as plt
mat = np.random.rand(4, 16)
mat = mat / mat.sum(axis = 1)[:, None]
mat=np.array(mat)
print(mat)
result=mat.reshape(1,64)
print(result)
result.shape
column_mat=result.transpose()
print(column_mat)
column_mat.shape
arr= np.random.rand(16,64)
arr=arr/10
print(arr)
final=np.matmul(arr, column_mat)
print(final)
from itertools import chain
final_list = list(chain.from_iterable(final))
print(final_list)
size = len(final_list)
plt.plot(final_list)

from itertools import chain
final_list = list(chain.from_iterable(final))
print(final_list)
```

2. Implementation of Controlled Object to get Stabilized Output:

```
import numpy as np
from matplotlib import pyplot as plt
A_0 = np.random.rand(4,1)
A_0 = A_0 / A_0.sum(axis = 0)[:, None]
H_0 = np.random.rand(16,16)
H_0 = H_0 / H_0.sum(axis = 1)[:, None]
H_1 = np.random.rand(16,16)
H_1 = H_1 / H_1.sum(axis = 1)[:, None]
H_2 = np.random.rand(16,16)
H_2 = H_2 / H_2.sum(axis = 1)[:, None]
H_3 = np.random.rand(16,16)
H_3 = H_3 / H_3.sum(axis = 1)[:, None]
mat=A_0[0][0]*H_0+A_0[1][0]*H_1+A_0[2][0]*H_2+A_0[3][0]*H_3
A=mat
A_n = A
Y=np.random.rand(16,1)
x = np.linspace(1, 16, num=16)
Y = Y.T
for i in range(10):
Y = np.matmul( Y, A_n)
y_temp = np.squeeze(Y)
plt.plot(x, y_temp, ( 'C' + str(i + 1)), label = ( 'C' + str(i +
1)))
plt.legend()
if i == 9:
plt.show()

     Y = np.matmul( Y, A_n)
```

3. Implementation of Controlled Object with Constant Input:

```
import numpy as np
from matplotlib import pyplot as plt
A_0 = np.random.rand(4,1)
A_0 = A_0 / A_0.sum(axis = 0)[:, None]
H_0 = np.random.rand(16,16)
H_0 = H_0 / H_0.sum(axis = 1)[:, None]
H_1 = np.random.rand(16,16)
H_1 = H_1 / H_1.sum(axis = 1)[:, None]
H_2 = np.random.rand(16,16)
H_2 = H_2 / H_2.sum(axis = 1)[:, None]
H_3 = np.random.rand(16,16)
H_3 = H_3 / H_3.sum(axis = 1)[:, None]
```

```
mat=A_0[0][0]*H_0+A_0[1][0]*H_1+A_0[2][0]*H_2+A_0[3][0]*H_3
A=mat
A_n = A
Y =
np.array([[0.0625],[0.0625],[0.0625],[0.0625],[0.0625],[0.0625],[0.0625],[0
.0625],[0.0625],[0.0625],[0.0625],[0.0625],[0.0625],[0.0625],[0.0625],[0.06
25]])
x = np.linspace(1, 16, num=16)
Y = Y.T
for i in range(10):
y_temp = np.squeeze(Y)
plt.plot(x, y_temp, ( 'C' + str(i + 1)), label = ( 'C' + str(i + 1)))
plt.legend()
if i == 9:
plt.show()
Y = np.matmul( Y, A_n)
```

4. Implementation of Controlled Object with defined Transition Matrix:

```
import numpy as np
from matplotlib import pyplot as plt
H_0 = np.random.rand(16,16)
H_0 = H_0 / H_0.sum(axis = 1)[:, None]
H_0[:,10]
H=np.array([[0.07527784, 0.02007027, 0.09163684, 0.06528682, 0.01063487,
0.09609753, 0.08839618, 0.03445689, 0.02279468, 0.07714319,
0.05092684, 0.07485238, 0.10503808, 0.12089839, 0.02480924,
0.06313317),(0.07634427, 0.02209472, 0.06155843, 0.01500389,
0.05707611,
0.03810861, 0.06667732, 0.09437364, 0.07070958, 0.03900604,
0.10449036, 0.0204581 , 0.09672827, 0.06740963, 0.04690289,
0.0384146),(0.03235141, 0.01512289, 0.05954445, 0.09539182,
0.06026821,
0.07703276, 0.07873357, 0.05375404, 0.10693678, 0.09842434,
0.05144003, 0.08517181, 0.05357468, 0.06412403, 0.06765408,
0.0060867),(0.06013753, 0.07416098, 0.12554014, 0.03448225,
0.01604578,
0.00394654, 0.10260812, 0.0855378 , 0.04544507, 0.11739346,
0.03847869, 0.0908026 , 0.10177899, 0.06574652, 0.01765938,
0.09423444),(0.16713655, 0.02581254, 0.02955681, 0.12589341,
0.09719125,
0.07040171, 0.0244071 , 0.0197867 , 0.10414959, 0.02371842,
0.01478322, 0.0610329 , 0.04698036, 0.05654566, 0.03507481,
0.02485988),(0.0487352 , 0.03067784, 0.09830142, 0.04620615,
0.10795428,
```

```
0.08666492, 0.01133415, 0.09104088, 0.08182612, 0.05917829,
0.08082296, 0.1108307 , 0.01918843, 0.0490753 , 0.10286994,
20
0.0013167),(0.00337123, 0.08780448, 0.12212272, 0.11918823,
0.03819318,
0.09716497, 0.04165674, 0.03166576, 0.03178489, 0.06080871,
0.11857938, 0.00812532, 0.03778478, 0.08367629, 0.11835609,
0.00097407),(0.12058477, 0.08031121, 0.11028541, 0.06716545,
0.00195529,
0.03283065, 0.11198451, 0.09172099, 0.02467429, 0.05050336,
0.05260971, 0.07967134, 0.00633676, 0.08009607, 0.07515299,
0.14245302),(0.08584856, 0.09712806, 0.0240823 , 0.04815418,
0.04506466,
0.05147445, 0.06169249, 0.04753094, 0.06520565, 0.07325289,
0.09936277, 0.06623568, 0.04622193, 0.03815901, 0.06784542,
0.04374992),(0.0689205 , 0.04106151, 0.11434367, 0.0005997 ,
0.08308746,
0.02913057, 0.1075402 , 0.01341225, 0.11165076, 0.05653748,
0.10214598, 0.02450561, 0.06104217, 0.07094784, 0.06744443,
0.06350415),(0.0617847 , 0.06952383, 0.00364139, 0.07337879,
0.14015709,
0.08818753, 0.03756679, 0.01559599, 0.05981211, 0.06486917,
0.08413208, 0.09016048, 0.04290792, 0.00275845, 0.1008866 ,
0.1000916),(0.01183893, 0.0226003 , 0.12284908, 0.08085116,
0.0588664 ,
0.08372591, 0.1116576 , 0.07514589, 0.05859817, 0.01797786,
0.12574085, 0.01144967, 0.00343883, 0.05912257, 0.05083616,
0.03312517),(0.05281292, 0.01039042, 0.08602995, 0.09242212,
0.07624426,
0.0483706 , 0.02982603, 0.10019737, 0.02785634, 0.07280174,
0.02869765, 0.09570464, 0.02171307, 0.09644372, 0.08907599,
0.05945247),(0.11261839, 0.03154921, 0.07638288, 0.03886505,
0.13522254,
0.05581319, 0.09525595, 0.03618694, 0.0546547 , 0.00160691,
0.03031774, 0.08061912, 0.02456201, 0.05823389, 0.11411402,
0.04532202),(0.07183118, 0.0194614 , 0.07181443, 0.01330194,
0.09676488,
0.03577768, 0.02423734, 0.14371819, 0.07245179, 0.12683937,
0.10145491, 0.0478678 , 0.01171488, 0.01024096, 0.10550277,
0.06423823),(0.00387947, 0.1126764 , 0.06697921, 0.10893782,
0.10555313,
0.0785023 , 0.05002096, 0.01810255, 0.00979778, 0.04282297,
0.03013733, 0.10749098, 0.12148221, 0.04416706, 0.00019646,
0.09630957)]])
H=H.T
Y_0 =
np.array([[0.0625],[0.0625],[0.0625],[0.0625],[0.0625],[0.0625],[0.0625],[0
.0625],[0.0625],[0.0625],[0.0625],[0.0625],[0.0625],[0.0625],[0.0625],[0.06
```

```
25]])
x = np.linspace(1, 16, num=16)
Y=Y_0
Y = Y.T
for i in range(10):
y_temp = np.squeeze(Y)
plt.plot(x, y_temp, ( 'C' + str(i + 1)), label = ( 'C' + str(i + 1)))
plt.legend()
if i == 9:
plt.show()
Y = np.matmul( Y, H)
```

5. Implementation of Controlled Object with zero input values:

```
import numpy as np
from matplotlib import pyplot as plt
Y_0 =
np.array([[0.0714285714],[0.0714285714],[0.0714285714],[0.0714285714],[0.07
14285714],[0.0714285714],[0.0714285714],[0.0714285714],[0],[0],[0.071428571
4],[0.0714285714],[0.0714285714],[0.0714285714],[0.0714285714],[0.071428571
4]])
Y=Y_0.T
H=np.array([(0.07527784, 0.02007027, 0.09163684, 0.06528682, 0.01063487,
0.09609753, 0.08839618, 0.03445689, 0.02279468, 0.07714319,
0.05092684, 0.07485238, 0.10503808, 0.12089839, 0.02480924,
0.06313317),(0.07634427, 0.02209472, 0.06155843, 0.01500389,
0.05707611,
0.03810861, 0.06667732, 0.09437364, 0.07070958, 0.03900604,
0.10449036, 0.0204581 , 0.09672827, 0.06740963, 0.04690289,
0.0384146),(0.03235141, 0.01512289, 0.05954445, 0.09539182,
0.06026821,
0.07703276, 0.07873357, 0.05375404, 0.10693678, 0.09842434,
0.05144003, 0.08517181, 0.05357468, 0.06412403, 0.06765408,
0.0060867),(0.06013753, 0.07416098, 0.12554014, 0.03448225,
0.01604578,
0.00394654, 0.10260812, 0.0855378 , 0.04544507, 0.11739346,
0.03847869, 0.0908026 , 0.10177899, 0.06574652, 0.01765938,
0.09423444),(0.16713655, 0.02581254, 0.02955681, 0.12589341,
0.09719125,
0.07040171, 0.0244071 , 0.0197867 , 0.10414959, 0.02371842,
0.01478322, 0.0610329 , 0.04698036, 0.05654566, 0.03507481,
0.02485988),(0.0487352 , 0.03067784, 0.09830142, 0.04620615,
0.10795428,
0.08666492, 0.01133415, 0.09104088, 0.08182612, 0.05917829,
0.08082296, 0.1108307 , 0.01918843, 0.0490753 , 0.10286994,
```

```
0.0013167),(0.00337123, 0.08780448, 0.12212272, 0.11918823,
0.03819318,
0.09716497, 0.04165674, 0.03166576, 0.03178489, 0.06080871,
0.11857938, 0.00812532, 0.03778478, 0.08367629, 0.11835609,
0.00097407),(0.12058477, 0.08031121, 0.11028541, 0.06716545,
0.00195529,
0.03283065, 0.11198451, 0.09172099, 0.02467429, 0.05050336,
0.05260971, 0.07967134, 0.00633676, 0.08009607, 0.07515299,
0.14245302),(0.08584856, 0.09712806, 0.0240823 , 0.04815418,
0.04506466,
0.05147445, 0.06169249, 0.04753094, 0.06520565, 0.07325289,
0.09936277, 0.06623568, 0.04622193, 0.03815901, 0.06784542,
0.04374992),(0.0689205 , 0.04106151, 0.11434367, 0.0005997 ,
0.08308746,
0.02913057, 0.1075402 , 0.01341225, 0.11165076, 0.05653748,
0.10214598, 0.02450561, 0.06104217, 0.07094784, 0.06744443,
0.06350415),(0.0617847 , 0.06952383, 0.00364139, 0.07337879,
0.14015709,
0.08818753, 0.03756679, 0.01559599, 0.05981211, 0.06486917,
0.08413208, 0.09016048, 0.04290792, 0.00275845, 0.1008866 ,
0.1000916),(0.01183893, 0.0226003 , 0.12284908, 0.08085116,
0.0588664 ,
0.08372591, 0.1116576 , 0.07514589, 0.05859817, 0.01797786,
0.12574085, 0.01144967, 0.00343883, 0.05912257, 0.05083616,
0.03312517),(0.05281292, 0.01039042, 0.08602995, 0.09242212,
0.07624426,
0.0483706 , 0.02982603, 0.10019737, 0.02785634, 0.07280174,
0.02869765, 0.09570464, 0.02171307, 0.09644372, 0.08907599,
0.05945247),(0.11261839, 0.03154921, 0.07638288, 0.03886505,
0.13522254,
0.05581319, 0.09525595, 0.03618694, 0.0546547 , 0.00160691,
0.03031774, 0.08061912, 0.02456201, 0.05823389, 0.11411402,
0.04532202),(0.07183118, 0.0194614 , 0.07181443, 0.01330194,
0.09676488,
0.03577768, 0.02423734, 0.14371819, 0.07245179, 0.12683937,
0.10145491, 0.0478678 , 0.01171488, 0.01024096, 0.10550277,
0.06423823),(0.00387947, 0.1126764 , 0.06697921, 0.10893782,
0.10555313,
0.0785023 , 0.05002096, 0.01810255, 0.00979778, 0.04282297,
0.03013733, 0.10749098, 0.12148221, 0.04416706, 0.00019646,
0.09630957)]])
H_0=H.T
x = np.linspace(1, 16, num=16)
j=0
for j in range(10):
y_temp = np.squeeze(Y)
plt.legend()
if j == 9:
```

```
plt.show()
if Y[0][8]>0.1 and Y[0][9]>0.1:
a=(Y[0][8]+Y[0][9])/2
for i in range(len(Y[0])):
if i ==8 or i==9:
continue
Y[0][i]+=a
Y[0][8]=0
Y[0][9]=0
plt.plot(x, y_temp, ( 'C' + str(j + 1)), label = ( 'C' + str(j + 1)))
Y=np.matmul(Y,H_0)
else:
Y=np.matmul(Y,H_0)
plt.plot(x, y_temp, ( 'C' + str(j + 1)), label = ( 'C' + str(j + 1)))
```

6. Implementation of Controlled Object with variation of zero input values:

```
import numpy as np
from matplotlib import pyplot as plt
Y_0 =
np.array([[0.0714285714],[0.0714285714],[0.0714285714],[0.0714285714],[0.07
14285714],[0.0714285714],[0.0714285714],[0.0714285714],[0],[0],[0.071428571
4],[0.0714285714],[0.0714285714],[0.0714285714],[0.0714285714],[0.071428571
4]])
Y=Y_0.T
H=np.array([(0.07527784, 0.02007027, 0.09163684, 0.06528682, 0.01063487,
0.09609753, 0.08839618, 0.03445689, 0.02279468, 0.07714319,
0.05092684, 0.07485238, 0.10503808, 0.12089839, 0.02480924,
0.06313317),(0.07634427, 0.02209472, 0.06155843, 0.01500389,
0.05707611,
0.03810861, 0.06667732, 0.09437364, 0.07070958, 0.03900604,
0.10449036, 0.0204581 , 0.09672827, 0.06740963, 0.04690289,
0.0384146),(0.03235141, 0.01512289, 0.05954445, 0.09539182,
0.06026821,
0.07703276, 0.07873357, 0.05375404, 0.10693678, 0.09842434,
0.05144003, 0.08517181, 0.05357468, 0.06412403, 0.06765408,
0.0060867),(0.06013753, 0.07416098, 0.12554014, 0.03448225,
0.01604578,
0.00394654, 0.10260812, 0.0855378 , 0.04544507, 0.11739346,
0.03847869, 0.0908026 , 0.10177899, 0.06574652, 0.01765938,
0.09423444),(0.16713655, 0.02581254, 0.02955681, 0.12589341,
0.09719125,
0.07040171, 0.0244071 , 0.0197867 , 0.10414959, 0.02371842,
```

```
0.01478322, 0.0610329 , 0.04698036, 0.05654566, 0.03507481,
0.02485988),(0.0487352 , 0.03067784, 0.09830142, 0.04620615,
0.10795428,
0.08666492, 0.01133415, 0.09104088, 0.08182612, 0.05917829,
0.08082296, 0.1108307 , 0.01918843, 0.0490753 , 0.10286994,
0.0013167),(0.00337123, 0.08780448, 0.12212272, 0.11918823,
0.03819318,
0.09716497, 0.04165674, 0.03166576, 0.03178489, 0.06080871,
0.11857938, 0.00812532, 0.03778478, 0.08367629, 0.11835609,
0.00097407),(0.12058477, 0.08031121, 0.11028541, 0.06716545,
0.00195529,
0.03283065, 0.11198451, 0.09172099, 0.02467429, 0.05050336,
0.05260971, 0.07967134, 0.00633676, 0.08009607, 0.07515299,
0.14245302),(0.08584856, 0.09712806, 0.0240823 , 0.04815418,
0.04506466,
0.05147445, 0.06169249, 0.04753094, 0.06520565, 0.07325289,
0.09936277, 0.06623568, 0.04622193, 0.03815901, 0.06784542,
0.04374992),(0.0689205 , 0.04106151, 0.11434367, 0.0005997 ,
0.08308746,
0.02913057, 0.1075402 , 0.01341225, 0.11165076, 0.05653748,
0.10214598, 0.02450561, 0.06104217, 0.07094784, 0.06744443,
0.06350415),(0.0617847 , 0.06952383, 0.00364139, 0.07337879,
0.14015709,
0.08818753, 0.03756679, 0.01559599, 0.05981211, 0.06486917,
0.08413208, 0.09016048, 0.04290792, 0.00275845, 0.1008866 ,
0.1000916),(0.01183893, 0.0226003 , 0.12284908, 0.08085116,
0.0588664 ,
0.08372591, 0.1116576 , 0.07514589, 0.05859817, 0.01797786,
0.12574085, 0.01144967, 0.00343883, 0.05912257, 0.05083616,
0.03312517),(0.05281292, 0.01039042, 0.08602995, 0.09242212,
0.07624426,
0.0483706 , 0.02982603, 0.10019737, 0.02785634, 0.07280174,
0.02869765, 0.09570464, 0.02171307, 0.09644372, 0.08907599,
0.05945247),(0.11261839, 0.03154921, 0.07638288, 0.03886505,
0.13522254,
0.05581319, 0.09525595, 0.03618694, 0.0546547 , 0.00160691,
0.03031774, 0.08061912, 0.02456201, 0.05823389, 0.11411402,
0.04532202),(0.07183118, 0.0194614 , 0.07181443, 0.01330194,
0.09676488,
0.03577768, 0.02423734, 0.14371819, 0.07245179, 0.12683937,
0.10145491, 0.0478678 , 0.01171488, 0.01024096, 0.10550277,
0.06423823),(0.00387947, 0.1126764 , 0.06697921, 0.10893782,
0.10555313,
0.0785023 , 0.05002096, 0.01810255, 0.00979778, 0.04282297,
0.03013733, 0.10749098, 0.12148221, 0.04416706, 0.00019646,
0.09630957)]])
H_0=H.T
j=0
```

```
x = np.linspace(1, 16, num=16)
cnt = 0
for j in range(10):
y_temp = np.squeeze(Y)
#plt.plot(x, y_temp, ( 'C' + str(j + 1)), label = ( 'C' + str(j + 1)))
plt.legend()
if j == 9:
plt.show()
if Y[0][8]<0.1 and Y[0][9]<0.1:
cnt += 1
if(cnt == 5) :
a=(Y[0][8]+Y[0][9])/2
print(a)
for i in range(len(Y[0])):
if i ==8 or i==9:
continue
#print(Y[0][i])
Y[0][i]+=a
Y[0][8]=0
Y[0][9]=0
plt.plot(x, y_temp, ( 'C' + str(j + 1)), label = ( 'C' + str(j +
1)))
Y=np.matmul(Y,H_0)
plt.plot(x, y_temp, ( 'C' + str(j + 1)), label = ( 'C' + str(j +
1)))
else:
cnt = 0
Y=np.matmul(Y,H_0)
plt.plot(x, y_temp, ( 'C' + str(j + 1)), label = ( 'C' + str(j + 1)))
```

7. Implementation of Transfer Entropy Algorithm:

```
import numpy as np

def transfer_entropy(x, y, k):
  """
  Calculates transfer entropy from x to y using k nearest neighbors
  x, y: numpy arrays of shape (n,)
  k: number of nearest neighbors to consider
  """
  n = len(x)
  te = 0.0
  for i in range(k, n):
    px = estimate_distribution(x[:i-k], k)
    py = estimate_distribution(y[:i-k], k)
    pxy = estimate_joint_distribution(x[:i-k], y[:i-k], k)
    pxy_prev = estimate_joint_distribution(x[:i-k-1], y[:i-k-1], k)
    te += np.sum(pxy * np.log2(pxy / (px * py)))
    te -= np.sum(pxy_prev * np.log2(pxy_prev / (px * py)))
```

```python
    return te / (n - k)

def estimate_distribution(x, k):
    """
    Estimates probability distribution of x using k nearest neighbors
    x: numpy array of shape (n,)
    k: number of nearest neighbors to consider
    """
    n = len(x)
    d = np.zeros((n, n))
    for i in range(n):
        for j in range(i+1, n):
            d[i,j] = np.abs(x[i] - x[j])
            d[j,i] = d[i,j]
    knn = np.argpartition(d, k+1, axis=1)[:,1:k+1]
    p = np.zeros(n)
    for i in range(n):
        p[i] = (k / (2 * n)) * np.sum(d[i,knn[i]] <= d[i,knn[:,knn[i]]])
    return p

def estimate_joint_distribution(x, y, k):
    """
    Estimates joint probability distribution of x and y using k nearest neighbors
    x, y: numpy arrays of shape (n,)
    k: number of nearest neighbors to consider
    """
    n = len(x)
    d = np.zeros((n, n))
    for i in range(n):
        for j in range(i+1, n):
            d[i,j] = np.sqrt((x[i]-x[j])**2 + (y[i]-y[j])**2)
            d[j,i] = d[i,j]
    knn = np.argpartition(d, k+1, axis=1)[:,1:k+1]
    p = np.zeros((n, n))
    for i in range(n):
        for j in range(i+1, n):
            p[i,j] = (k / (n**2)) * np.sum((d[i,knn[i]] <= d[i,knn[:,knn[i]]]) & (d[j,knn[j]] <= d[j,knn[:,knn[j]]]))
            p[j,i] = p[i,j]
    return p
```

8. Implementation of Transfer Entropy as log-likelihood Ratio:

```python
import numpy as np
from scipy.stats import norm

def transfer_entropy(x, y, k=1, l=1, m=1):
    """
    Calculate the transfer entropy from x to y using the log-likelihood ratio method.

    Args:
        x (numpy.ndarray): The source time series.
        y (numpy.ndarray): The target time series.
        k (int): The history length of x (default 1).
        l (int): The history length of y (default 1).
        m (int): The number of nearest neighbors to use for estimating probability densities (default 1).
```

```
    Returns:
      float: The transfer entropy from x to y.
    """
    n = len(x)
    x_past = np.array([x[i-k:i] for i in range(k, n)])
    y_past = np.array([y[i-l:i] for i in range(l, n)])
    xy_past = np.hstack((x_past, y_past))
    x_future = x[k:]
    y_future = y[l:]
    log_ratio = 0
    for i in range(n-k-l):
      xy_distances = np.sum(np.abs(xy_past - xy_past[i]), axis=1)
      xy_knn = np.argsort(xy_distances)[:m+1]
      x_knn = np.argsort(np.abs(x_past - x_past[i]))[:m+1]
      y_knn = np.argsort(np.abs(y_past - y_past[i]))[:m+1]
      p_xy = np.mean(np.isin(xy_knn, i))
      p_x = np.mean(np.isin(x_knn, i))
      p_y = np.mean(np.isin(y_knn, i))
      if p_x == 0 or p_y == 0 or p_xy == 0 or np.isnan(p_xy):
        continue
      log_ratio += np.log2(p_xy / (p_x * p_y))
    return log_ratio / (n - k - l)
```

9. Implementation of Symbolic Transfer Entropy:

```
import numpy as np
from itertools import product

def SymbolicTransferEntropy(data, source, destination, order):
  """
  Compute Symbolic Transfer Entropy from source to destination
  of specified order

  Parameters:
  data (np.ndarray): The data series
  source (int): The source index
  destination (int): The destination index
  order (int): The order of the transfer entropy

  Returns:
  float: The transfer entropy value
  """
  alphabet = np.unique(data)
  K = alphabet.size
  N = data.shape[0]
  r = np.zeros((K, order + 1))
  c = np.zeros((K, K, order))

  for i in range(order + 1):
    for j in range(K):
      subseq = data[i:N-order+i+1]
      mask = (subseq[i-1] == alphabet[j])
      r[j, i] = mask.sum()
```

```
for i in range(order):
    for j in range(K):
        for k in range(K):
            subseq = data[i:N-order+i+1]
            mask = ((subseq[i-1] == alphabet[j]) & (subseq[order-1] == alphabet[k]))
            c[j, k, i] = mask.sum()

index_tuples = product(range(K), repeat=order+1)
sum_num = 0
sum_denom = 0

for indices in index_tuples:
    cond_count = r[indices[0], 0]
    for i in range(1, order+1):
        cond_count *= c[indices[i-1], indices[i], i-1] / r[indices[i-1], i-1]
    sum_num += (indices[0] == source) & (indices[1:] == (destination,)*(order))
    sum_denom += (indices[0] == source) & (cond_count > 0)

return np.log(K) * (sum_num - sum_denom) / N
```

References:

1. C. D. Charalambous, C. K. Kourtellaris and I. Tzortzis, "Information Transfer of Control Strategies: Dualities of Stochastic Optimal Control Theory and Feedback Capacity of Information Theory," in IEEE Transactions on Automatic Control, vol. 62, no. 10, pp. 5010-5025, Oct. 2017, doi: 10.1109/TAC.2017.2690147.
2. O. Hernandez-Lerma and J. Lasserre, Discrete-Time Markov Control Processes: Basic Optimality Criteria, New York, NY, USA:Springer-Verlag, 1996.
3. P. R. Kumar and J. H. van Schuppen, "On the optimal control of stochastic systems with an exponential-of-integral performance index", J. Math. Anal. Appl., vol. 80, no. 2, pp. 312-332, Apr 1981.
4. I. R. Petersen, M. R. James and P. Dupuis, "Minimax optimal control of stochastic uncertain systems with relative entropy constraints", IEEE Trans. Autom. Control, vol. 45, no. 3, pp. 398-412, Mar 2000.
5. I. I. Gihman and A. V. Skorohod, Controlled Stochastic Processes, New York, NY, USA:Springer-Verlag, 1979
6. S. I. Bross and M. A. Wigger , On the relay channel with receiver transmitter feedback , IEEE Trans. Inform. Theory , 55 (2009), pp. 275 -- 291 .
7. C. D. Charalambous, C. Kourtellaris and S. Loyka , Capacity achieving distributions and separation principle for feedback Gaussian channels with memory: The LQG theory of directed information , IEEE Trans. Inform. Theory , 64 (2018), pp. 6384 – 6418
8. R. G. James, N. Barnett, and J. P. Crutchfield, ‘Information Flows? A Critique of Transfer Entropies’, Phys. Rev. Lett., vol. 116, p. 238701, Jun. 2016.
9. T. Schreiber, ‘Measuring Information Transfer’, Phys. Rev. Lett., vol. 85, pp. 461–464, Jul. 2000.
10. L. Barnett and T. Bossomaier, ‘Transfer Entropy as a Log-Likelihood Ratio’, Phys. Rev. Lett., vol. 109, p. 138105, Sep. 2012.
11. M. Staniek and K. Lehnertz, ‘Symbolic Transfer Entropy’, Phys. Rev. Lett., vol. 100, p. 158101, Apr. 2008.
12. Gencaga, D., Knuth, K. H., & Rossow, W. B. (2015). A Recipe for the Estimation of Information Flow in a Dynamical System. *Entropy*, *17*(1), 438-470.

https://doi.org/10.3390/e17010438

13. D. Gencaga, 'Transfer Entropy', Entropy, vol. 20, p. 288, 04 2018.

14. Minyue Fu, "Linear quadratic Gaussian control with quantized feedback," 2009 American Control Conference, St. Louis, MO, USA, 2009, pp. 2172-2177, doi: 10.1109/ACC.2009.5160058.

15. M. Fu, "Lack of Separation Principle for Quantized Linear Quadratic Gaussian Control," in IEEE Transactions on Automatic Control, vol. 57, no. 9, pp. 2385-2390, Sept. 2012, doi: 10.1109/TAC.2012.2187010.

16. P. R. Kumar and P. Varaiya, Stochastic Systems: Estimation Identification and Adaptive Control, Englewood Cliffs, NJ, USA:Prentice-Hall, 1986.

17. P. R. Kumar and J. H. van Schuppen, "On the optimal control of stochastic systems with an exponential-of-integral performance index", *J. Math. Anal. Appl.*, vol. 80, no. 2, pp. 312-332, Apr 1981.

18. T. M. Cover and J. A. Thomas, Elements of Information Theory, Hoboken, NJ, USA:Wiley, 2006.

19. C. E. Shannon, "A mathematical theory of communication", *Bell Syst. Tech.*, vol. 27, pp. 379-423, Jul. 1948.

20. C. E. Shannon, "Coding theorems for a discrete source with a fidelity criterion", *IRE Nat. Conv. Rec.*, vol. 27, no. Pt4, pp. 325-350, Jul. 1959.

21. O. Hernandez-Lerma and J. Lasserre, Discrete-Time Markov Control Processes: Basic Optimality Criteria, New York, NY, USA:Springer-Verlag, 1996.

22. M. R. James, J. Baras and R. J. Elliot, "Risk-sensitive control and dynamic games for partially observed discrete-time nonlinear systems", *IEEE Trans. Autom. Control*, vol. 39, no. 4, pp. 780-792, Apr 1994.

23. N. U. Ahmed and C. D. Charalambous, "Stochastic minimum principle for partially observed systems subject to continuous and jump diffusion processes and driven by relaxed controls", *SIAM J. Control Optim.*, vol. 51, no. 4, pp. 3235-3257, 2013.

24. H. Marko, "The bidirectional communication theory—A generalization of information theory", *IEEE Trans. Commun.*, vol. 21, no. 12, pp. 1345-1351, Dec. 1973.

25. H. Permuter, T. Weissman and A. Goldsmith, "Finite state channels with time-invariant deterministic feedback", *IEEE Trans. Inf. Theory*, vol. 55, no. 2, pp. 644-

662, Feb. 2009.

26. T. Cover and S. Pombra, "Gaussian feedback capacity", *IEEE Trans. Inf. Theory*, vol. 35, no. 1, pp. 37-43, Jan. 1989.

27. J. Arnhold, P. Grassberger, K. Lehnertz, and C. E. Elger, Physica (Amsterdam) **134D**, 419 (1999); M. G. Rosenblum, A. S. Pikovsky, and J. Kurths, Phys. Rev. Lett. **76**, 1804 (1996); M. Le Van Quyen, C. Adam, M. Baulac, J. Martinerie, and F. J. Varela, Brain Res. **792**, 24 (1998).

28. L. Shi, Y. Yuan and H. Zhang, "Sensor data scheduling for linear quadratic Gaussian control with full state feedback," 2012 American Control Conference (ACC), Montreal, QC, Canada, 2012, pp. 2030-2035, doi: 10.1109/ACC.2012.6314650.

29. G. Nair, R. Evans, I. Mareels and W. Moran, "Topological feedback entropy and nonlinear stabilization", *IEEE Trans. Autom. Control*, vol. 49, no. 9, pp. 1585-1597, Sep. 2004.

Final

ORIGINALITY REPORT

10%	5%	7%	2%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Charalambos D. Charalambous, Christos K. Kourtellaris, Ioannis Tzortzis. "Information Transfer of Control Strategies: Dualities of Stochastic Optimal Control Theory and Feedback Capacity of Information Theory", IEEE Transactions on Automatic Control, 2017 Publication	1%
2	Submitted to Indian Institute of Technology, Kharagpure Student Paper	1%
3	docshare.tips Internet Source	<1%
4	Submitted to University College London Student Paper	<1%
5	ebin.pub Internet Source	<1%
6	epdf.tips Internet Source	<1%
7	Jan H. van Schuppen. "Control and System Theory of Discrete-Time Stochastic Systems",	<1%

Springer Science and Business Media LLC, 2021
Publication

---

8 Sabag, Oron, Haim Permuter, and Navin Kashyap. "The Feedback Capacity of the $(1,\infty)$-RLL Input-Constrained Erasure Channel", IEEE Transactions on Information Theory, 2015.
Publication — <1%

---

9 "Abstracts", 2012 American Control Conference (ACC), 2012.
Publication — <1%

---

10 silo.pub
Internet Source — <1%

---

11 cpb-us-w2.wpmucdn.com
Internet Source — <1%

---

12 epdf.pub
Internet Source — <1%

---

13 edoc.hu-berlin.de
Internet Source — <1%

---

14 www.collectionscanada.gc.ca
Internet Source — <1%

---

15 Submitted to Southern New Hampshire University - Continuing Education
Student Paper — <1%

---

16 Stochastic Networked Control Systems, 2013.

Publication

<1%

17 Submitted to Universiti Teknologi Malaysia
Student Paper

<1%

18 Submitted to Coventry University
Student Paper

<1%

19 Terry Bossomaier, Lionel Barnett, Michael Harré, Joseph T. Lizier. "An Introduction to Transfer Entropy", Springer Science and Business Media LLC, 2016
Publication

<1%

20 Understanding Complex Systems, 2014.
Publication

<1%

21 Randa Herzallah. "A Fully Probabilistic Design for Tracking Control for Stochastic Systems with Input Delay", IEEE Transactions on Automatic Control, 2020
Publication

<1%

22 Andreas A. Malikopoulos. "A Multiobjective Optimization Framework for Online Stochastic Optimal Control in Hybrid Electric Vehicles", IEEE Transactions on Control Systems Technology, 2016
Publication

<1%

23 Submitted to Liverpool John Moores University
Student Paper

<1%

24	lastminuteengineers.com Internet Source	<1 %
25	www.wseas.us Internet Source	<1 %
26	Wenbin Shi, Pengjian Shang, Aijing Lin. "The coupling analysis of stock market indices based on cross-permutation entropy", Nonlinear Dynamics, 2014 Publication	<1 %
27	multidollar.ru Internet Source	<1 %
28	nozdr.ru Internet Source	<1 %
29	Baoyao Yang, Mang Ye, Qingxiong Tan, Pong C. Yuen. "Cross-Domain Missingness-Aware Time-Series Adaptation With Similarity Distillation in Medical Applications", IEEE Transactions on Cybernetics, 2020 Publication	<1 %
30	P. F. Verdes. "Assessing causality from multivariate time series", Physical Review E, 2005 Publication	<1 %
31	deepai.org Internet Source	<1 %
	repository.kaust.edu.sa	

32	Internet Source	<1%
33	www.frontiersin.org Internet Source	<1%
34	"Seizure Prediction in Epilepsy", Wiley, 2008 Publication	<1%
35	Jamie Evans, Vikram Krishnamurthy. "Optimal sensor scheduling for hidden Markov model state estimation", International Journal of Control, 12/15/2001 Publication	<1%
36	Jun Yoneyama. "Risk-sensitive optimal control for jump systems with application to sampled-data systems", International Journal of Systems Science, 2010 Publication	<1%
37	M. Jayakumar, R.N. Banavar. "Risk-sensitive filters for recursive estimation of motion from images", IEEE Transactions on Pattern Analysis and Machine Intelligence, 1998 Publication	<1%
38	M.R. James, J.S. Baras, R.J. Elliott. "Risk-sensitive control and dynamic games for partially observed discrete-time nonlinear systems", IEEE Transactions on Automatic Control, 1994 Publication	<1%

39 Subhrajit Sinha, Umesh Vaidya. "Causality preserving information transfer measure for control dynamical system", 2016 IEEE 55th Conference on Decision and Control (CDC), 2016
Publication

<1 %

40 Ugrinovskii, V.A.. "Guaranteed cost LQG control for uncertain systems with a normalized coprime factor uncertainty structure", Systems & Control Letters, 200910/11
Publication

<1 %

41 mast.queensu.ca
Internet Source

<1 %

42 pure.tue.nl
Internet Source

<1 %

43 qspace.library.queensu.ca
Internet Source

<1 %

44 Abdelmalek Ouannou, Adil Brouri, Laila Kadi, Hafid Oubouaddi. "Identification of switched reluctance machine using fuzzy model", International Journal of System Assurance Engineering and Management, 2022
Publication

<1 %

45 Ameet Kumar Banerjee, Md Akhtaruzzaman, Andreia Dionisio, Dora Almeida, Ahmet Sensoy. "Nonlinear nexus between

<1 %

cryptocurrency returns and COVID–19 COVID-19 news sentiment", Journal of Behavioral and Experimental Finance, 2022
Publication

46 E. Soljanin. "Compressing quantum mixed-state sources by sending classical information", IEEE Transactions on Information Theory, 8/2002 <1 %
Publication

47 Edoardo Pinzuti, Patricia Wollstadt, Aaron Gutknecht, Oliver Tüscher, Michael Wibral. "Measuring spectrally-resolved information transfer", PLOS Computational Biology, 2020 <1 %
Publication

48 Jan A. Freund. "Frequency and phase synchronization in stochastic systems", Chaos An Interdisciplinary Journal of Nonlinear Science, 2003 <1 %
Publication

49 fr.scribd.com <1 %
Internet Source

50 hal.archives-ouvertes.fr <1 %
Internet Source

51 helda.helsinki.fi <1 %
Internet Source

52 www.doc.ic.ac.uk <1 %
Internet Source

53	www.scilit.net Internet Source	<1 %
54	Charalambos D. Charalambous, Christos K. Kourtellaris, Sergey Loyka. "Capacity Achieving Distributions and Separation Principle for Feedback Gaussian Channels With Memory: the LQG Theory of Directed Information", IEEE Transactions on Information Theory, 2018 Publication	<1 %
55	Yaohui Huang, Peisong Zhang, Ziyang Wang, Zhenkun Lu, Zhijin Wang. "HFMD Cases Prediction Using Transfer One-Step-Ahead Learning", Neural Processing Letters, 2022 Publication	<1 %
56	Yixin Diao, J.L. Hellerstein, S. Parekh, R. Griffith, G.E. Kaiser, D. Phung. "A control theory foundation for self-managing computing systems", IEEE Journal on Selected Areas in Communications, 2005 Publication	<1 %
57	Advances in Statistical Control Algebraic Systems Theory and Dynamic Systems Characteristics, 2008. Publication	<1 %
58	D. Sundararajan. "Chapter 4 Time-Domain Analysis of Continuous Systems", Springer	<1 %

Science and Business Media LLC, 2023
Publication

59	Victor Privalsky. "Time Series Analysis in Climatology and Related Sciences", Springer Science and Business Media LLC, 2021 Publication	<1%
60	Wu, Wei, and Ari Arapostathis. "Optimal Sensor Querying: General Markovian and LQG Models With Controlled Observations", IEEE Transactions on Automatic Control, 2008. Publication	<1%

Exclude quotes	On	Exclude matches	Off
Exclude bibliography	On		