Reliability Engineering

Module 4

Discrete Time Markov Processes

Gyan Ranjan Biswal
PhD (IITR), FIE, SMIEEE, LMISTE
Associate Professor
Department of Electrical & Electronics Engineering (EEE)
Veer Surendra Sai University of Technology (formerly UCE), Burla
PIN – 768018, Sambalpur (Odisha), India

E-mail: gyanbiswal@vssut.ac.in
URL: http://www.vssut.ac.in
Google Scholar: Gyan Biswal
Home Page: http://in.linkedin.com/pub/gyan-biswal/14/458/a8
ORCID id: https://orcid.org/0000-0001-7730-1985

Contact Hours: Tuesday, 04:30 PM to 05:30 PM at E-106

Gyan Ranjan Biswal received his B.E. in Electronics Engineering from the Pt. Ravishankar Shukla University, India in 1999 and M. Tech. (Honors) in Instrumentation & Control Engineering from the Chhattisgarh Swami Vivekananda Technical University, India in 2009 followed by Ph.D. in Electrical Engineering, specialized in the area of Power System Instrumentation (Power Generation Automation) from the Indian Institute of Technology Roorkee, India in 2013.

He is expertise in Design and Development of cooling systems for large size electrical generators, and the C&I of process industries. He has been in academia for about twelve years. Presently, he is with VSS University of Technology, Burla, India at the capacity of Associate Professor, EEE from Dec. 2016, and HOD, EEE from Jan. 2020 to Feb. 2023, and conferred with the Best faculty Award for the AY 2021-22 under Professor/ Associate Professor category . He has more than 75 publications in various Journals and Conferences of Internationally repute to his credit. He also holds a patent as well, and filed one more. He also adapted one international edition book published by Pearson India. He received research grants of US $80,000 (INR 64 lakhs). He has been supervised 02 Ph.D. theses and 09 Masters' theses, and ongoing 03 PhD theses. He has also been recognized with many national and international awards by elite bodies. He has been awarded with CICS award under the head of Indian National Science Academy for travel support to USA, MHRD Fellowship by Govt. of India, and Gopabandhu Das Scholarship in his career. His major areas of interests are Power System Instrumentation, Industrial Automation, Robust and Intelligent Control, the Smart Sensors, IoT enabled Smart Sensors, the Smart Grid, Hydrogen Cooling System, Hydrogen Storage and Its Processing, Fuel Cell lead Sustainable Sources of Energy, and System Reliability.

Dr. Biswal is a Fellow IE (India), Senior Member of IEEE, USA, and Life Member of ISTE, India. He is actively involved in review panels of different societies of international repute viz. IEEE, IFAC, and the ISA. Currently, he is also actively involved as a Member of IEEE–SA (Standards Association) working groups; IEEE P1876 WG, IEEE P21451-001 WG, and IEEE P1415. He has also been invited for delivering guest lectures at World Congress on Sustainable Technologies (WCST) Conf. 2012, London, UK, INDICON 2015, New Delhi, India, National Power Training Institute (NPTI), Nangal, India, and G.B. Pant Engineering College, Pauri, Gharwal, India, Surendra Sai University of Technology, Burla, as a guest expert in 2016 IEEE PES General Meeting Boston, MA, USA, 1st Annual Webinar of Complex Engineering System, Politecnico di Milano, Italy in 2022, and Keynote lecture in 12th EAI International Conference on Sensor Systems and Software, Portugal in 2021.

Syllabus

Reliability Engineering

MODULE-I (6 HOURS)

Types of System, Qualitative and Quantitative assessment, Use of quantitative assessment, Reliability Definition and Concepts, Reliability Indices and Criteria, Reliability and Availability, Absolute and Relative Reliability, Reliability Evaluation Technique, Reliability Improvement, Reliability Activities in System Design & its Economics, Basic Probability Theory, Binomial Distribution and its engineering applications.

MODULE-II (10 HOURS)

Network modeling concepts, Series & Parallel Systems, Series-Parallel System, Partially Redundant & Standby redundant System. Modeling and Evaluation Concept, Conditional Probability Approach, Cut Set Method, Application and Comparison of Previous Technique, Tie Set Method, Connection Matrix Technique, Event Trees, Fault Tree, Multi-Failure Mode.

MODULE-III (8 HOURS)

Distribution Concept & terminologies, General Reliability Function& their evaluation techniques, Shape of Reliability Function. The Poison Distribution & the Normal Concept, Exponential, Weibull, Gamma, Rayleigh, Lognormal and rectangular distributions, Data Analysis, System Reliability Evaluation of different kinds of Using Probability Distributions, Mean Time to Failure, Wear out And Component Reliability, Maintenance And Component Reliability.

MODULE-IV (8 HOURS)

Discrete Markov Chains: General Modeling Concept, Stochastic Transitional Probability Matrix, Time Dependent Probability Evaluation, Limiting State Probability Evaluation, Absorbing States, Application of Discrete Markov Technique.

Continuous Markov Process: General Modeling Concept, State Space Diagrams, Stochastic Transitional Probability Matrix, Evaluating Limiting State Probabilities, Evaluating Time Dependent State Probabilities, Reliability Evaluation in Repairable System, Mean Time to Failure, Application of Technique To Complex System.

MODULE-V (7 HOURS)

Frequency andDuration Technique: Application to Multistate Problems, Frequency Balance Approach, Two Stage Repair and Installation Process. Approximate System Reliability Evaluation. System with Non-Exponential Distribution. Monte Carlo Simulation.

Text and Reference Books

Recommended Text Books:

1. Roy Billinton, Ronald N. Allan. “Reliability Evaluation of Engineering Systems”Second Edition.

Reference Books:

* Gupta A.K., Reliability, Maintenance and Safety Engineering. University Science Press.

Other Important References

Reference Sites:	
1.	NPTEL, The National Programme on Technology Enhanced Learning (NPTEL): https://nptel.ac.in/
2.	MIT OpenCourseWare : https://ocw.mit.edu/index.htm
3.	https://www.youtube.com/channel/UC0ISZ4dMZcIBeIzjZVRZhJw/videos [Gyan Ranjan Biswal @gyanranjanbiswal5649]

Course Outcomes

Upon successful completion of this course, you (students) will be able to

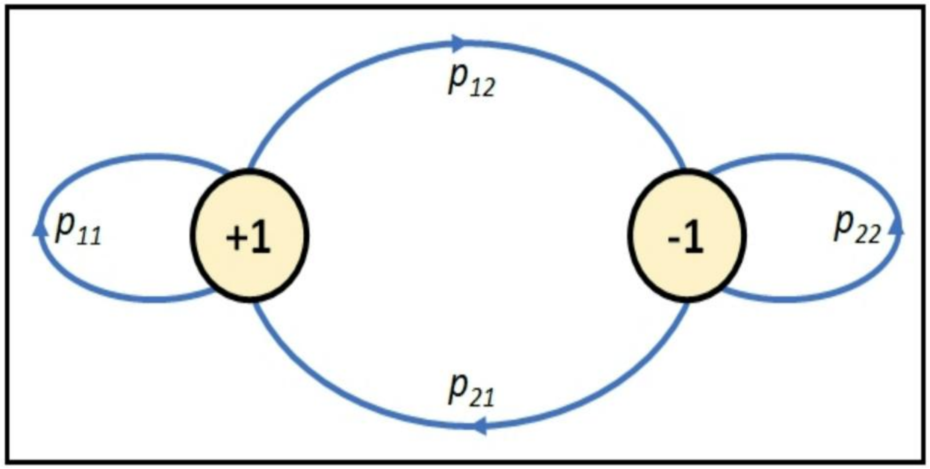
CO1	Define the basic terms in reliability engineering concepts.
CO2	Construct and implement the network modelling of simple and complex systems.
CO3	Evaluate probability distribution for reliability of a system.
CO4	Incorporate discrete and continuous Markov processes for reliability evaluation.
CO5	Express competence on approximate reliability evaluation techniques.

Discrete Time Markov Processes

Introduction

- A Discrete Time Markov Chain can be used to describe the behavior of a system that jumps from one state to another state with a certain probability.
- This probability of transition to the next state depends only on what state the system is in currently, i.e. it does not depend on which states the system was in prior to the current state.
- Even the simplest of Markov Chains can create richly interesting patterns of system behavior.

A simple 2-state Markov chain

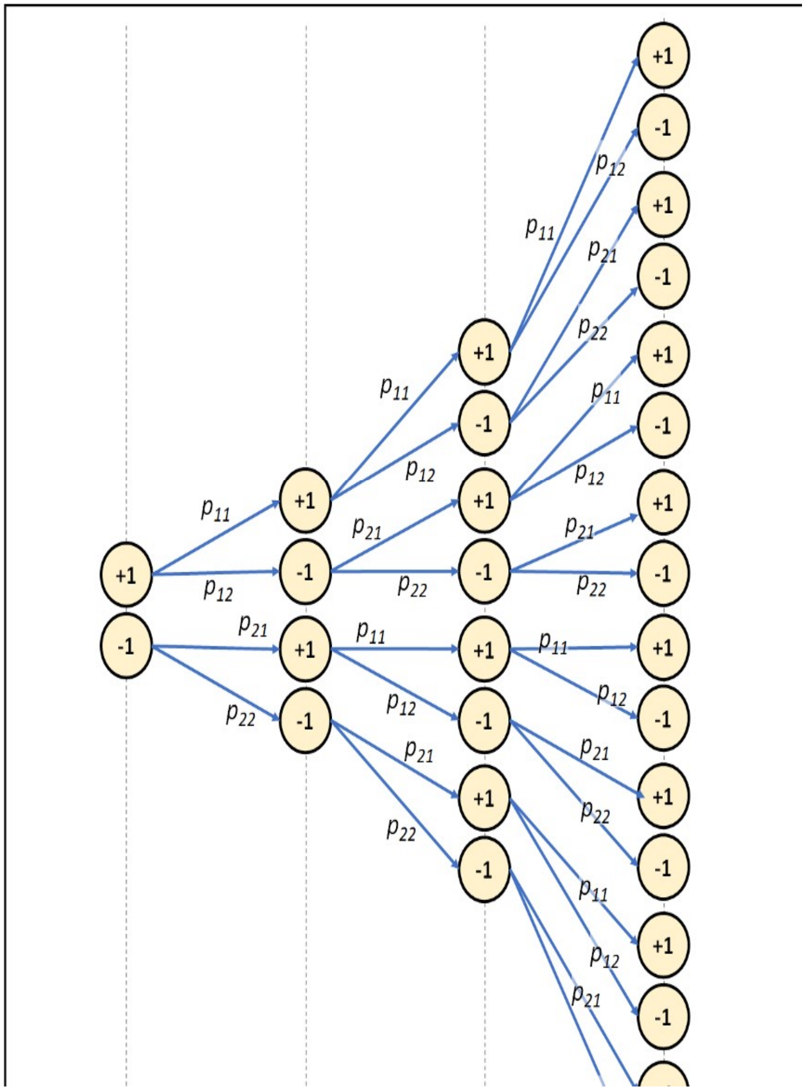


[State transition diagram of the Markov process]

- **The Markov chain shown above has two states, or regimes as they are sometimes called: +1 and -1. There are four types of state transitions possible between the two states:**
  - State +1 to state +1: This transition happens with probability p_11
  - State +1 to State -1 with transition probability p_12
  - State -1 to State +1 with transition probability p_21
  - State -1 to State -1 with transition probability p_22

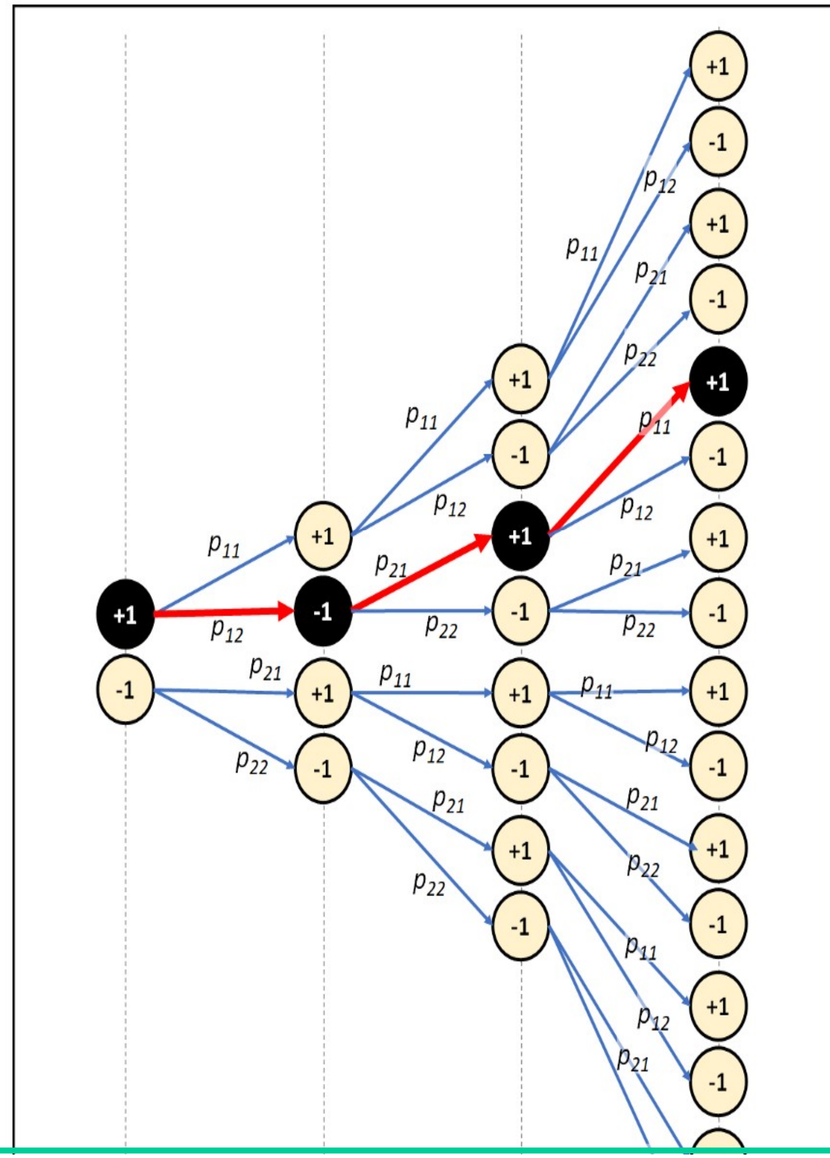
A 2-state Markov process

- This diagram shows another way to represent this Markov process. Here, X-axis: time axis
- The bubbles represent the different possible states that the process could be in at each time step.



Realization of a 2-state Markov chain across 4 consecutive time steps

- In reality, at each time step, the process will be in exactly one of the many possible states.
- The black bubbles depict one such realization of the Markov process across 4 successive time steps.
- There are many such realizations possible. In a 2-state Markov process, there are 2^N possible realizations of the Markov chain over N time steps.
- A realization of a Markov chain along the time dimension is a time series.



State Transition Matrix

- In a 2-state Markov chain, there are four possible state transitions and the corresponding transition probabilities.

- We can represent them in a state transition matrix P as follows:

$$P = \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix}$$

- The state transition matrix P of an n-state Markov process is:

$$P = \begin{bmatrix} p_{11} & \cdots & p_{1n} \\ \vdots & \ddots & \vdots \\ p_{n1} & \cdots & p_{nn} \end{bmatrix}$$

- The state transition matrix has the following two important properties:

  - Since each element p_{ij} is a probability, $0 \leq p_{ij} \leq 1$
  - Each row of P sums to 1.0 i.e. $p_i = \sum_{j=1}^{n} p_{ij} = 1$. This is because the row index represents the source state at time t and the column index represents the destination state at time (t+1). If the process is in source state i at time t, at (t+1), it has to be in one of the allowed set of states (1,2,3,…,n).

- Thus, we can restate the transition matrix of a 2-state Markov process as follows:

$$P = \begin{bmatrix} p_{11} & 1 - p_{11} \\ 1 - p_{22} & p_{22} \end{bmatrix}$$

Markov distributed random variable

- **Introduction:**

  - A random variable X_t is considered. The suffix t in X_t denotes the time step.

  - At each time step t, X_t takes a value from the state space $[1,2,3, \ldots, n]$ as per some probability distribution. One possible sequence of values that X_t takes is $\{X_0 = 1, X_1 = 3, X_2 = 4, X_3 = 0, \ldots, X_t = k\}$.

- **Transition probabilities as conditional probabilities:**

  - At time t, the probability that X_t takes some value j given that X_t had taken a value i at the previous time step $(t-1)$ is given by the following conditional probability: $p_{ij} = P(X_t = j | X_{t-1} = i)$. Here, p_{ij} is the transition probability.

The Markov property

- The Markov property states that p_{ij} is independent of the state in which the system was at times $(t-2), (t-3), \dots, 0$. The Markov property is stated as follows:

$$P(X_t = j | X_{t-1} = i) = P(X_t = j | X_{t-1} = i; X_{t-2} = i_2; X_{t-3} = i_3; \dots; X_0 = i_0)$$

- **n-step transition probabilities:**
  - The state transition matrix P has this nice property by which if you multiply it with itself k times, then the matrix P^k represents the probability that system will be in state j after hopping through k number of transitions starting from state i.
  - For example, consider the following transition probabilities matrix for a 2-step Markov process: $P = \begin{bmatrix} 0.25 & 0.75 \\ 0.75 & 0.25 \end{bmatrix}$
  - The 2-step transition probabilities are calculated as follows:

$$P \times P = P^2 = \begin{bmatrix} 0.625 & 0.375 \\ 0.375 & 0.625 \end{bmatrix}$$

  - In P^2, $p_{11} = 0.625$ is the probability of returning to state 1 after having traversed through two states starting from state 1. $p_{12} = 0.375$ is the probability of reaching state 2 in exactly two time steps starting from state 1. And so on.

State Probability Distribution

- We could continue multiplying P with itself forever to see how the n-step probabilities change over time. What would be more interesting is if we could know what is the unconditional probability distribution of X_t at each time step t.
- For example, in our 2-step Markov process example, at each time step t, what is the probability with which X_t could be +1 or -1? These probabilities constitute what is known as the state probability distribution of the Markov variable X_t at time t and it is denoted by π_t, (or δ_t).
- For example, for the 2-state Markov process consisting of states +1 and -1, the state distribution is given as follows:

$$\pi_t = [P(X_t = +1), P(X_t = -1)]$$

State Probability Distribution

- An n-state Markov process that operates over the states [1,2,3,…,n] could be in any one of those n states at time t, so π_t is a vector of length n as follows:

$$\pi_t = [P(X_t = 1), P(X_t = 2), P(X_t = 3), \dots, P(X_t = n)]$$

- In general, each element of π_t can be referenced using the notation π_{jt}. π_{jt} is indexed using two variables j and t, indicating the unconditional probability π of the process being in state j at time t:

$$\pi_t = [\pi_{1t}, \pi_{2t}, \pi_{3t}, \dots, \pi_{\mathrm{n}t}]$$

where, $\pi_{jt} = P(X_t = \mathrm{j})$

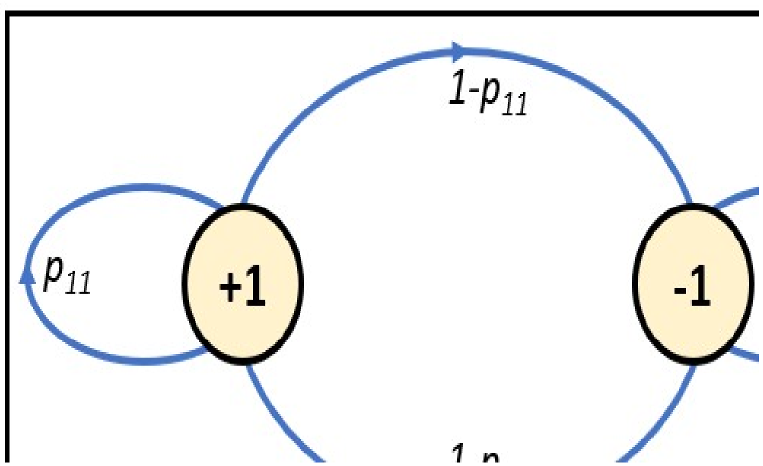
- Since π_t is a probability distribution, its elements always sum up to 1.0: $\sum_{j=1}^{n} \pi_{jt} = 1$

State Probability Distribution

- **How to calculate π_t:**
  - Typically, one assumes a certain value for π_0 which is the probability vector for state 0. And given π_0, it can be shown that π_t can be calculated as: $\pi_t = \pi_0 P^t$
  - The idea behind computing π_t is to matrix-multiply the state transition matrix with itself t number of times to get the t-step transition probabilities, and multiply the t-step transition probabilities with the unconditional probability distribution π_0 at t=0.
- The state probability distribution of a Markov process at time t depends on:
  - The initial probability distribution π_0 at time t=0, and
  - It depends on time step t. Thus, the probability distribution evolves over ti

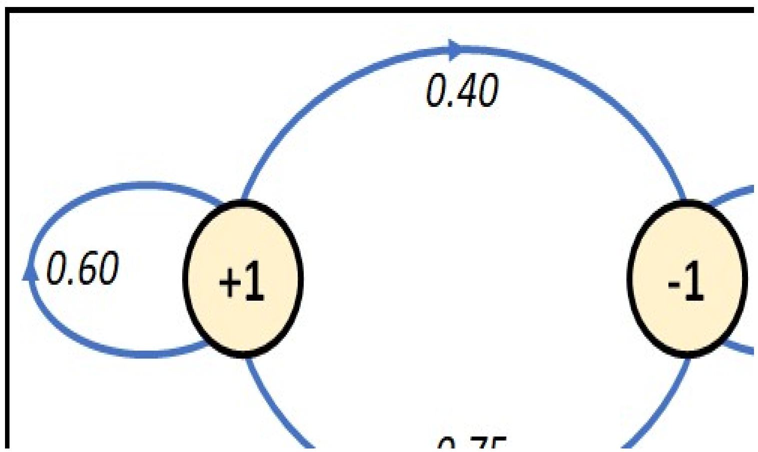
Markov Process Example

- Suppose the stock of Acme Corporation increases or decreases as per the following rules:
  - As compared to the previous day's closing price, the current day's closing price is either higher or lower by some percentage points. For, e.g. on some day, Acme's stock closes 2.3% higher than the previous day. On other days, it might close 0.8% lower than the previous day's closing price.
  - Given any sequence of 3 days denoted by 'day before', 'yesterday' and 'today', if Acme has closed higher yesterday than the day before, it will close higher today than yesterday with a probability p_{11}. Consequently, it would close lower today than yesterday with a probability $(1 - p_{11})$
  - Let's also assume that if Acme has closed lower yesterday than the day before, the probability that it would once again close lower today is p_{22}. And therefore, the probability that it would close higher today than it did yesterday is $(1 - p22\ )$
- Here, Acme's stock price can be modeled using a 2-state Markov process. It's state transition diagram is:



Markov Process Example

- It's transition probabilities matrix P is: $\begin{bmatrix} p_{11} & 1-p_{11} \\ 1-p_{22} & p_{22} \end{bmatrix}$
- Suppose the probability of Acme's price's moving up two days in a row is 0.6 and the chances of the price moving down for two days in a row is 0.25. Thus, $p_{11} = 0.6$ and $p_{22} = 0.25$. And therefore, we have: $p_{12} = 1 - p_{11} = 0.4$, and, $p_{21} = 1 - p_{22} = 0.75$
- The state transition matrix for Acme's stock is: $\begin{bmatrix} 0.60 & 0.40 \\ 0.75 & 0.25 \end{bmatrix}$.
- The state transition diagram of the Markov process model for Acme's stock is as follows:



- The unconditional probability distribution is given as: $\pi_t = [P(X_t = +1), P(X_t = -1)]$
- Let's assume an initial value of $\pi_0 = [0.5,0.5]$, i.e. an equal probability of it's closing higher (+1 state) or lower (-1 state) on listing day, as compared to the IPO price. If we repeatedly apply the formula for $\pi_t = \pi_0 * P^t$, we will get the probability distribution vector π_t for each time step t.

Python code for calculation of π_t

```
import numpy as np
from matplotlib import pyplot as plt

#initialize the transition matrix P
P=np.array([[0.6,0.4],[0.75,0.25]])
#initialize pi_0
pi_0=np.array([0.5, 0.5])

#set up the array to accumulate the state probabilities at times t=1 to 10
pi=[]
pi.append(pi_0)
P_mul=P.copy()

#calculate the state probability for each t and store it away
for i in range(10):
   P_mul=np.matmul(P_mul,P)
   pi_t = np.matmul(pi_0,P_mul)
   pi.append(pi_t)
pi = np.array(pi)
```

- The above code stores all the calculated π_t vectors in the array π.

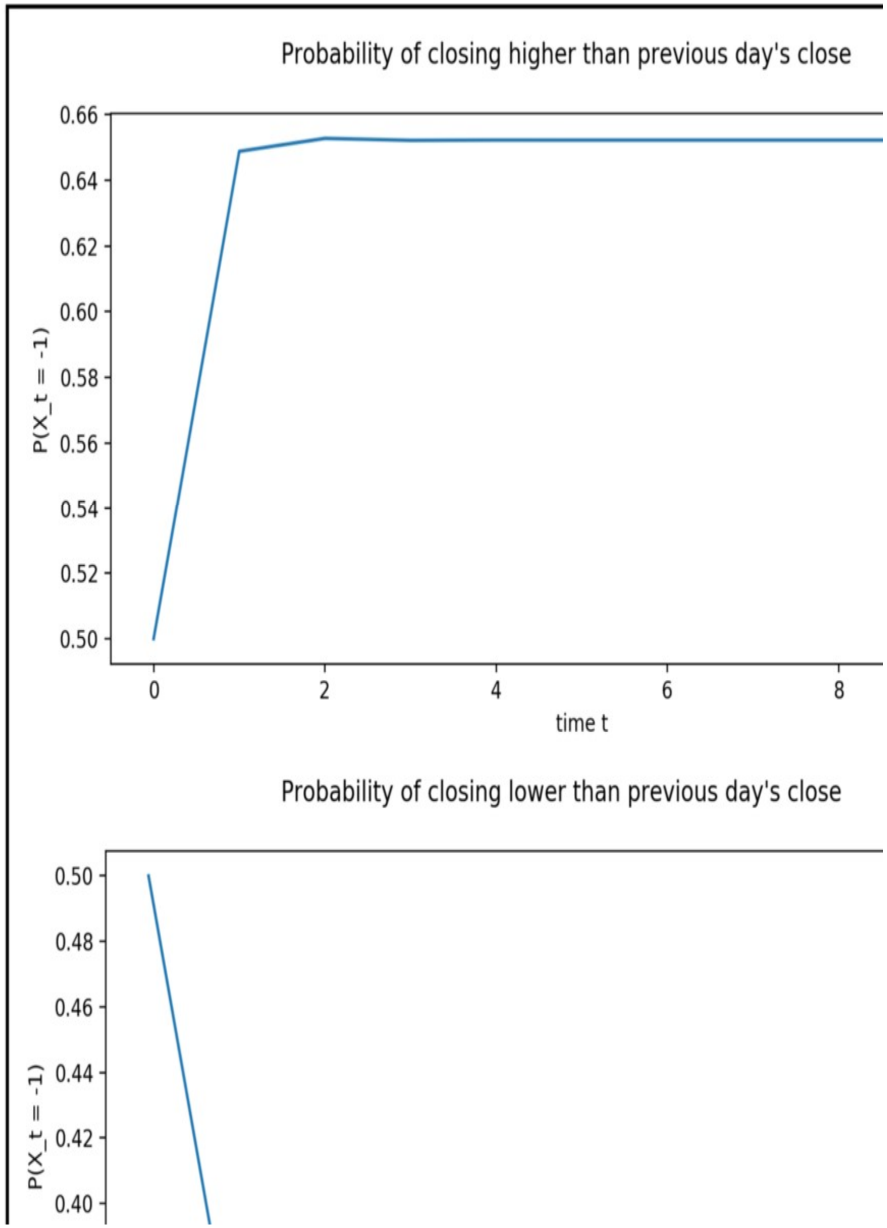
Python code for calculation of π_t (contd….)

- Let's separately plot the components π_1t and π_2t of π_t= [π_1t, π_2t] for t =1 through 10.

```
#plot pi_2t = P(X_t = -1) versus t
fig = plt.figure()
fig.suptitle('Probability of closing lower than previous day\'s close')
plt.plot(range(len(pi)), pi[:,1])
plt.show()

#plot pi_1t = P(X_t = +1) versus t
fig = plt.figure()
fig.suptitle('Probability of closing higher than previous day\'s close')
plt.plot(range(len(pi)), pi[:,0])
plt.show()
```

Python code for calculation of π_t (contd....)



Python code for calculation of π_t (contd....)

- We see that in just a few time steps, the unconditional probabilities vector π_t= [π_1t, π_2t] settles down into the steady state value [0.65217391, 0.34782609].

- In fact, it can be shown that if a two state Markov process with the following transition matrix is run for 'long time':

$$\begin{bmatrix} p_{11} & 1 - p_{11} \\ 1 - p_{22} & p_{22} \end{bmatrix}$$

- Then the state probability distribution π_t steady-states to the following constant (limiting) probability distribution that is independent of both t and the initial probability distribution π_0:

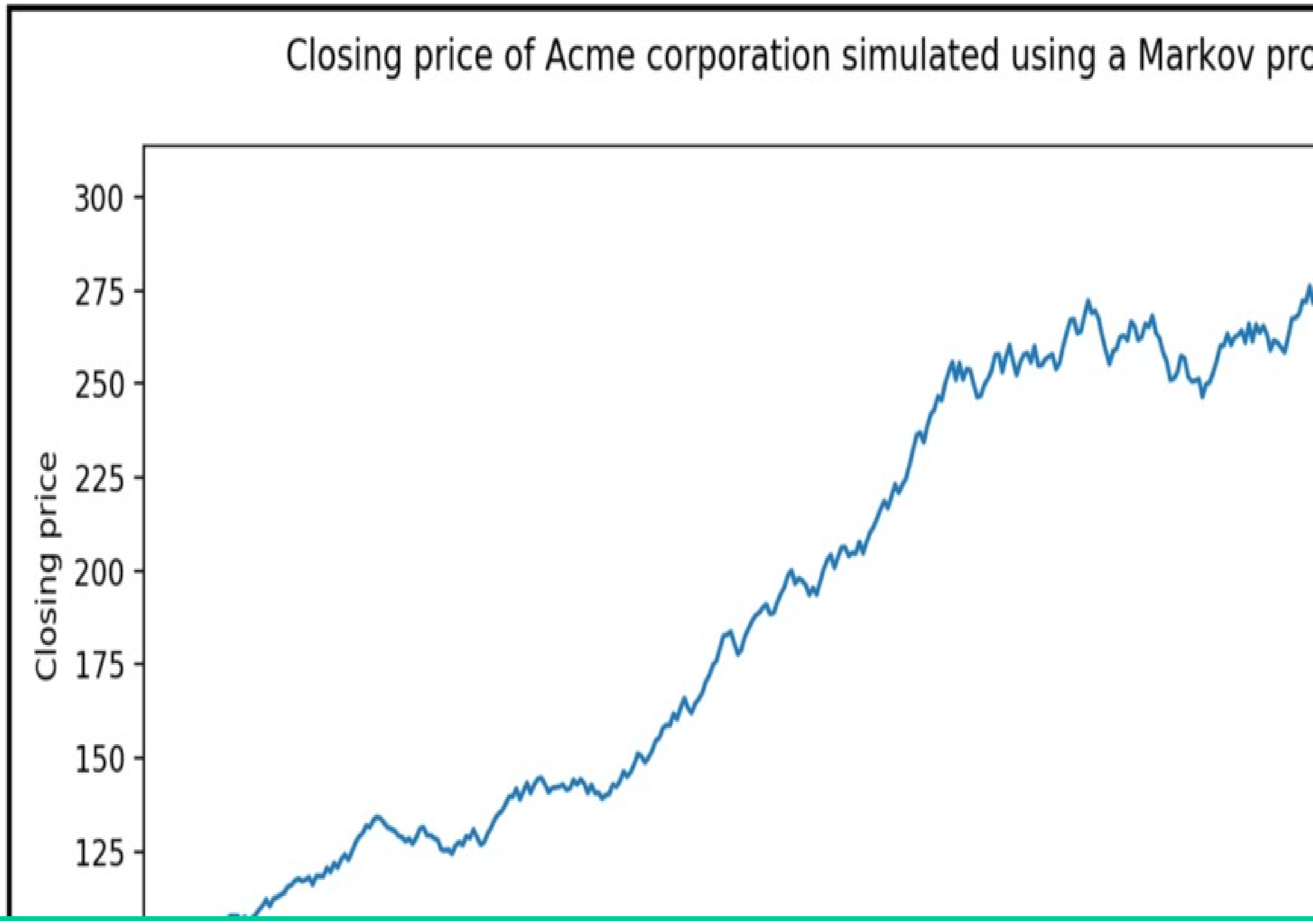
$$\pi_t = \left[\frac{1 - p_{11}}{2 - (p_{11} + p_{22})}, \frac{1 - p_{22}}{2 - (p_{11} + p_{22})}\right]$$

Simulating Acme's stock price movement using a Markov Process

- Let's simulate Acme Corp's stock price movement at each time step t using the probability distribution π_t at time t. The simulation procedure goes like this:
  - Assume Acme's IPO price to be $100.
  - Set the initial probability distribution:
  - $\pi_0 = [P(X_t = +1) = 0.5, \; P(X_t = -1) = 0.5]$
  - Map the +1 Markov state to the action of increasing Acme's previous day's closing price by a random percentage in the interval [0.0%, 2.0%].
  - Map the -1 Markov state to reducing Acme's previous day's closing price by a similar random percentage in the interval [0.0%, 2.0%].
  - At each time step t, calculate the state probability distribution $\pi_t = [P(X_t = +1), P(X_t = -1) = 0.5]$, by using the formula $\pi_t = \pi_0 * P^t$.
  - Generate a uniformly distributed random number in the interval [0, 1.0]. If this number is less than or equal to $\pi_{1t} = P(X_t = +1)$, increase the closing price from the previous time step by a random percentage in the interval [0.0%, 2.0%], else decrease the previous closing price by the same random percentage. Repeat this procedure for as many time steps as needed.

Simulating Acme's stock price movement using a Markov Process

- Here is the plot of Acme's stock price changes over 365 trading days:



Simulating Acme's stock price movement using a Markov Process

- Here is the source code for generating the above plot:

```
#Simulate the closing price of a company
closing_price = 100.0
#initialize pi_0
pi_0=np.array([0.5, 0.5])
#create a random delta in the range [0, 2.0]
delta = random.random() * 2
#generate a random number in the range [0.0, 1.0]
r = random.random()
#if r <= P(X_t = +1), increase the closing price by delta,
#else decrease the closing price by delta
if r <= pi_0[0]:
   closing_price = closing_price*(100+delta)/100
else:
   closing_price = math.max(closing_price*(100-delta)/100,1.0)
#accumulate the new closing price
closing_prices = [closing_price]
P_mul=P.copy()
T=365
```

Simulating Acme's stock price movement using a Markov Process

```
#now repeat this procedure 365 times
for i in range(T):
   #calculate the i-step transition matrix P^i
   P_mul=np.matmul(P_mul,P)
   #multiply it by pi_0 to get the state probability for time i
   pi_t = np.matmul(pi_0,P_mul)
   # create a random delta in the range [0, 2.0]
   delta = random.random() * 2
   # generate a random number in the range [0.0, 1.0]
   r = random.random()
   # if r <= P(X_t = +1), increase the closing price by delta,
   # else decrease the closing price by delta
   if r <= pi_t[0]:
      closing_price = math.max(closing_price*(100+delta)/100,1.0)
   else:
      closing_price = closing_price*(100-delta)/100
   # accumulate the new closing price
   closing_prices.append(closing_price)

#plot all the accumulated closing prices
fig = plt.figure()
fig.suptitle('Closing price of Acme corporation simulated using a Markov process')
plt.xlabel('time t')
plt.ylabel('Closing price')
plt.plot(range(T+1), closing_prices)
plt.show()
```

REFERENCES

- Sachin Date, “Time Series Analysis, Regression, and Forecasting”, https://timeseriesreasoning.com/contents/introduction-to-discrete-time-markov-processes