To verify the controllability and observability of the given system, we need to check the properties of the system's controllability and observability matrices.

The controllability matrix is given by: C = [B AB]

The observability matrix is given by: O = [C; CA]

Controllability: To check the controllability of the system, we need to determine if the controllability matrix C is full rank (i.e., its rank is equal to the number of states).

Given: A = [-1 -1; 1 0] B = [1; 0]

The controllability matrix C becomes: C = [B AB] = [1 -1; 0 1]

Calculating the rank of C, we find: rank(C) = 2

Since the rank of C is equal to the number of states (2), the system is controllable.

Observability: To check the observability of the system, we need to determine if the observability matrix O is full rank (i.e., its rank is equal to the number of states).

Given: C = [0 1]

The observability matrix O becomes: O = [C; CA] = [0 1; -1 -1]

Calculating the rank of O, we find: rank(O) = 2

Since the rank of O is equal to the number of states (2), the system is observable.

Therefore, based on the calculations, the given system is both controllable and observable.

Q1) Explain sensitivity? what is effects of feedback on sensitivity.

Sensitivity refers to the ability of a system or an individual to detect or perceive changes in a given parameter or stimulus. It is a measure of how responsive or reactive something is to the presence or magnitude of a particular input. Sensitivity can be observed in various domains, including physical systems, biological organisms, and even human perception.

In general, sensitivity is quantified as the ratio of the change in the system's response to the change in the input. It indicates how much the output or response of a system will vary for a given change in the input. Higher sensitivity means even small changes in the input will result in noticeable changes in the output, while lower sensitivity implies that larger changes in the input are required to produce a noticeable effect on the output.

Feedback plays a crucial role in influencing sensitivity in different contexts. Feedback is a process where a portion of the output of a system is fed back and used as an input to modify the behavior of the system. The effects of feedback on sensitivity can be understood in the following ways:

1. Amplification or attenuation: Feedback can amplify or attenuate the sensitivity of a system. Positive feedback tends to amplify the sensitivity by reinforcing the output and promoting further change in the same direction. This can lead to exponential growth or instability in some cases. On the other hand, negative feedback tends to attenuate the sensitivity by reducing the discrepancy between the desired output and the actual output, thereby stabilizing the system.
2. Control and regulation: Feedback mechanisms are often employed in control systems to regulate sensitivity. By continuously monitoring the output and comparing it to a desired value, feedback can be used to adjust the system's behavior and maintain stability. This allows for more precise control over the sensitivity, ensuring it remains within acceptable ranges.
3. Sensitivity to feedback itself: Sensitivity can also refer to how responsive a system or individual is to feedback. Some systems may be highly sensitive to feedback, meaning that even small changes in the feedback signal can significantly influence their behavior. Others may be less sensitive, requiring more substantial feedback inputs to induce a noticeable effect. Sensitivity to feedback can impact the ability to learn, adapt, and self-regulate.

Q2) Explain signal flow graph in detail with the help of Masons gain formula.

A signal flow graph is a graphical representation of a system that consists of interconnected nodes and directed edges. It is commonly used in control systems engineering to analyze and understand the flow of signals through a system. The graph visually represents the paths that signals take from inputs to outputs, allowing for the application of various analysis techniques.

In a signal flow graph, nodes represent variables or signals, and directed edges represent the relationships between these variables. The direction of the edges indicates the flow of signals, usually from left to right. The edges are labeled with gain values, which represent the amplification or attenuation that occurs as the signal passes through a particular edge.

Mason's gain formula, also known as Mason's rule or Mason's theorem, is a technique used to calculate the overall transfer function of a signal flow graph. It provides a systematic method to

determine the ratio of the output signal to the input signal, taking into account all possible paths and the gains associated with each path.
Here's a step-by-step explanation of how to apply Mason's gain formula:
Step 1: Assign a variable (let's call it Δ) to represent the determinant of the graph. The determinant represents the overall transfer function of the system.
Step 2: Calculate the Δ value. To calculate Δ, you need to sum up the products of individual path gains and their associated cofactors.

- Start by identifying all the individual paths in the graph. A path is a sequence of edges that connects an input node to an output node without forming a loop.
- For each path, calculate the product of the gains along that path.
- Determine the cofactor of each path. To calculate the cofactor of a path, remove all nodes and edges that are part of the path from the original graph and find the determinant of the resulting subgraph.
- Multiply the path gain by its cofactor and sum up all these products. This will give you the value of Δ.

Step 3: Calculate the Δ_i values. To calculate Δ_i, you need to modify the original graph by removing the edges that are associated with the i-th forward path (i.e., the path you are interested in calculating the gain for).

- For each modified graph, calculate the Δ value using the same process as in Step 2.

Step 4: Calculate the overall gain. The overall gain (G) is calculated by summing up the Δ_i values and dividing it by Δ:

$G = \Delta_1/\Delta + \Delta_2/\Delta + \ldots + \Delta_n/\Delta$

where n represents the total number of forward paths in the graph.

The resulting G value represents the overall transfer function of the system.

Mason's gain formula is a powerful tool for analyzing signal flow graphs and determining the transfer function of complex systems. By considering all possible paths and their gains, it provides a systematic approach to understanding the flow of signals and their overall impact on the system's behavior.

Q3) Determine transfer function of given system using block diagram reduction technique?

To determine the transfer function of a system using block diagram reduction techniques, you need to simplify the block diagram by systematically reducing it into a single transfer function. Here's a step-by-step process to follow:

Step 1: Identify the individual blocks in the block diagram. These blocks represent the transfer functions of the system's components.

Step 2: Apply the rules of block diagram reduction to simplify the diagram. Some commonly used rules include:

- Series rule: When two or more blocks are connected in series, their transfer functions can be multiplied together.
- Parallel rule: When two or more blocks are connected in parallel, their transfer functions can be added together.
- Feedback rule: When a block has feedback connections, you can use the feedback rules to simplify the diagram.

Step 3: Repeat the reduction process until you obtain a single transfer function representing the overall system.

Let's illustrate this process with an example. Suppose we have the following block diagram:

luaCopy code

```
+--------+ +--------+ ---->| Block1 |------>| Block2 |----> +--------+ +--------+ | ^ | | +--------|-----+
```

Step 1: Identify the blocks. In this example, we have two blocks: Block1 and Block2.

Step 2: Apply the rules of block diagram reduction.

- Since the blocks are connected in series, we can multiply their transfer functions: Block1 * Block2.
- If there were any feedback connections, we would need to apply the feedback rules as well.

Step 3: Obtain the overall transfer function.

The transfer function of the overall system can be represented as:

Transfer function = Block1 * Block2

In this simplified form, you have the transfer function of the given system, which can be further analyzed and manipulated as needed.

Q4) Derive expression for delay time, rise time, peak time and settling time?

To derive expressions for delay time, rise time, peak time, and settling time, we first need to understand their definitions and the characteristics of a system's step response.

1. Delay Time (t_d): The delay time represents the time it takes for the system's response to reach a specified fraction of its final value after the application of a step input. It is typically measured as the time at which the response first crosses a specific threshold, such as 10% or 50% of the final value.
2. Rise Time (t_r): The rise time is the time it takes for the system's response to transition from a specified lower threshold to a specified upper threshold during the rising portion of the step

response. It is often defined as the time taken for the response to go from, say, 10% to 90% or 20% to 80% of the final value.

3. Peak Time (t_p): The peak time is the time at which the system's response reaches its maximum or peak value during the step response.
4. Settling Time (t_s): The settling time is the time it takes for the system's response to reach and stay within a certain tolerance band around the final value. It is commonly defined as the time required for the response to stay within a small percentage (e.g., 2% or 5%) of the final value and remain there.

The specific expressions for these time parameters can vary depending on the system's transfer function and characteristics. However, for a second-order underdamped system, which is a common system type, approximate formulas can be used:

1. Delay Time: The delay time is usually negligible for second-order underdamped systems.
2. Rise Time: For a second-order underdamped system, the rise time can be estimated using the following expression: $t_r = \pi / (\omega_n * \zeta * \sqrt{1 - \zeta^2})$

where ω_n is the natural frequency of the system and ζ is the damping ratio.

3. Peak Time: The peak time for a second-order underdamped system is given by: $t_p = \pi / (\omega_d * \sqrt{1 - \zeta^2})$

where ω_d is the damped natural frequency of the system.

4. Settling Time: The settling time for a second-order underdamped system can be approximated as: $t_s \approx 4 / (\zeta * \omega_n)$

These expressions provide approximations for the time parameters based on the characteristics of a second-order underdamped system. It's important to note that these formulas may not hold true for all system types, and for more accurate results, a thorough analysis of the system's transfer function and characteristics may be necessary.

Q5) Find the steady state error for unit step, unit ramp and unit acceleration inputs for the following systems.?

To find the steady-state error for unit step, unit ramp, and unit acceleration inputs in a control system, we can use the final value theorem. The final value theorem states that the steady-state value of the output is equal to the limit of the transfer function multiplied by the input as time approaches infinity.

Given the transfer function $G(s) = 10 / (s * (0.1s + 1) * (0.5s + 1))$, we can find the steady-state error for different input types:

1. Unit Step Input: The Laplace transform of the unit step input is 1/s. To find the steady-state error, we substitute s = 0 into the transfer function and apply the final value theorem:

E(s) = 1/s * G(s) E_ss = lim(s->0) [s * E(s)] E_ss = lim(s->0) [s * (1/s * G(s))] E_ss = lim(s->0) [G(s)] E_ss = lim(s->0) [10 / (s * (0.1s + 1) * (0.5s + 1))] E_ss = 10 / (0.1 * 0.5) E_ss = 200

Therefore, the steady-state error for a unit step input is 200.

2. Unit Ramp Input: The Laplace transform of the unit ramp input is 1/s^2. Using the final value theorem again:

E(s) = 1/s^2 * G(s) E_ss = lim(s->0) [s * E(s)] E_ss = lim(s->0) [s * (1/s^2 * G(s))] E_ss = lim(s->0) [G(s) / s] E_ss = lim(s->0) [10 / ((s^2) * (0.1s + 1) * (0.5s + 1))] E_ss = ∞

For a unit ramp input, the steady-state error is infinite (∞), indicating that the system cannot track a ramp input perfectly.

3. Unit Acceleration Input: The Laplace transform of the unit acceleration input is 1/s^3. Using the final value theorem:

E(s) = 1/s^3 * G(s) E_ss = lim(s->0) [s * E(s)] E_ss = lim(s->0) [s * (1/s^3 * G(s))] E_ss = lim(s->0) [G(s) / s^2] E_ss = lim(s->0) [10 / ((s^3) * (0.1s + 1) * (0.5s + 1))] E_ss = 0

For a unit acceleration input, the steady-state error is 0, indicating that the system can accurately track and follow an acceleration input.

Therefore, the steady-state errors for the given system are:

- Unit step input: 200
- Unit ramp input: ∞ (infinite)
- Unit acceleration input: 0

Q6) Determine stability of the system with the following characteristic equation.
D(s)=S^6+S^5+7S^4+6S^3+31S^2+25S+25

To determine the stability of the system with the given characteristic equation D(s), we need to analyze the roots of the equation. A system is stable if all the roots of the characteristic equation have negative real parts.

The characteristic equation is: D(s) = s^6 + s^5 + 7s^4 + 6s^3 + 31s^2 + 25s + 25

To determine the stability, we can analyze the roots of the equation. There are different ways to perform this analysis, such as using the Routh-Hurwitz criterion or checking the location of the roots on the complex plane. Let's use the Routh-Hurwitz criterion in this case.

Step 1: Create the Routh array using the coefficients of the characteristic equation:

Row 1: [1, 7, 31] Row 2: [1, 6, 25] Row 3: [2.333, 18.333] Row 4: [2.083, 25] Row 5: [4.819]

Step 2: Check the signs of the first elements in each row. If any of the first elements are negative, it indicates that there are roots with positive real parts, meaning the system is unstable. In this case, the first element of Row 5 is positive (4.819), so we conclude that the system is stable.

Therefore, based on the Routh-Hurwitz criterion, the given system with the characteristic equation D(s) = s^6 + s^5 + 7s^4 + 6s^3 + 31s^2 + 25s + 25 is stable.

Q7) Write a short note on PI Controller and PID Controller?

A PI controller (Proportional-Integral controller) and a PID controller (Proportional-Integral-Derivative controller) are commonly used control algorithms in engineering and automation systems. They are feedback control mechanisms that adjust the system's output based on the error between the desired setpoint and the actual value.

1. PI Controller: A PI controller combines proportional and integral control actions to improve the performance of a control system. It provides proportional control to respond to the current error and integral control to eliminate steady-state errors.

The proportional term of a PI controller produces an output that is directly proportional to the error signal. It helps in reducing the system's settling time and improving the response speed. However, it may not eliminate steady-state errors entirely.

The integral term of a PI controller sums up the error over time and adjusts the output accordingly. This integration action helps in eliminating steady-state errors by continuously reducing the accumulated error. The integral term is particularly useful when there are constant disturbances or biases in the system.

2. PID Controller: A PID controller builds upon the PI controller by adding a derivative control action. It includes proportional, integral, and derivative control terms to achieve better dynamic response and robustness.

The derivative term of a PID controller measures the rate of change of the error signal and adjusts the output proportionally. It helps in predicting the future behavior of the system based on the current rate of change. By dampening the response and reducing overshoot, the derivative term improves system stability and transient response.

The PID controller combines all three control actions to adjust the system output. The proportional term provides immediate response to the error, the integral term eliminates steady-state errors, and the derivative term improves the system's response to changing conditions.

Both PI and PID controllers are widely used in various applications, including industrial control systems, robotics, process control, and more. The choice between a PI and PID controller depends on the specific requirements of the system, such as desired response time, stability, and sensitivity to disturbances. Tuning the controller gains is essential to optimize the control performance and achieve the desired system behavior.

Q8) A Unity feedback system has an open loop transfer function $G(S)=K(S+4)/S^2+2s+2$?

To analyze the unity feedback system with the given open-loop transfer function G(s), which is $G(s) = K(S+4)/(S^2+2s+2)$, we can determine the stability, poles, and the effect of the gain K.

1. Stability: To determine the stability of the system, we need to analyze the poles of the transfer function. The system is stable if all the poles have negative real parts.

   The denominator of G(s) is S^2+2s+2. We can find the poles by solving the characteristic equation:

   S^2+2s+2 = 0

   Using the quadratic formula, we can solve for the poles:

   S = (-2 ± √(2^2 - 4*1*2)) / 2 S = (-2 ± √(-4)) / 2 S = (-2 ± 2i) / 2 S = -1 ± i

   Since the real part of the poles is -1, which is negative, we can conclude that the system is stable.

2. Poles: From the characteristic equation, we found that the system has two complex conjugate poles at -1 ± i. These poles determine the system's dynamics and response.
3. Effect of Gain K: The gain K in the transfer function G(s) = K(S+4)/(S^2+2s+2) affects the overall system behavior. Increasing K amplifies the input signal, making the system response more pronounced. However, high values of K can lead to instability or oscillations in the system.

   To optimize the system's performance, the gain K should be carefully selected or tuned based on the desired response specifications such as settling time, overshoot, and stability requirements.

   In summary, the given unity feedback system with the open-loop transfer function G(s) = K(S+4)/(S^2+2s+2) is stable, with complex conjugate poles at -1 ± i. The gain K influences the system's response and should be chosen appropriately to meet the desired performance criteria.

Q9) A Unity feedback system has an open loop transfer function G(S)=10/S(1+0.4s)(1+0.1s) Determine GM, PM and comment on system stability using bode plot?

To analyze the system's stability and determine the gain margin (GM) and phase margin (PM) using the Bode plot, let's work through the steps:

1. Convert the transfer function $G(s) = 10 / (s(1 + 0.4s)(1 + 0.1s))$ to Bode form: $G(s) = 10 / (s(1 + 0.4s)(1 + 0.1s)) = 10 / (s^2 + 0.5s + 0.04)$
2. Identify the key parameters from the Bode plot:
    - Gain crossover frequency (ωc): The frequency at which the magnitude plot intersects 0 dB.
    - Phase crossover frequency (ωp): The frequency at which the phase plot intersects -180 degrees.
    - GM: The magnitude at the gain crossover frequency (ωc) in dB.
    - PM: The phase margin, which is the amount of phase lag at the gain crossover frequency (ωc) before reaching -180 degrees.
3. Plot the Bode magnitude and phase plots for G(s): The magnitude plot can be calculated as: $|G(j\omega)| = 10 / \sqrt{((\omega^2 + 0.5\omega + 0.04)^2)}$

   The phase plot can be calculated as: $\varphi(j\omega) = -\arctan(\omega / (0.5 - \omega^2))$
4. Determine the gain crossover frequency (ωc): The gain crossover frequency is the frequency at which the magnitude plot intersects 0 dB. It can be found by solving: $20\log10(|G(j\omega c)|) = 0$ dB
5. Determine the phase crossover frequency (ωp): The phase crossover frequency is the frequency at which the phase plot intersects -180 degrees. It can be found by solving: $\varphi(j\omega p) = -180$ degrees
6. Calculate the gain margin (GM): $GM = 1 / |G(j\omega c)|$ in dB
7. Calculate the phase margin (PM): $PM = -180$ degrees $- \varphi(j\omega c)$
8. Analyze the stability and comment on the system:
    - If $GM > 1$ and $PM > 0$, the system is stable and has a margin of stability.
    - If $GM < 1$ or $PM < 0$, the system is unstable, and the margin of stability is insufficient.

By following the above steps and plotting the Bode magnitude and phase plots, you can determine the gain margin (GM) and phase margin (PM) for the given unity feedback system. The stability of the system can be assessed based on the values of GM and PM, as explained in step 8.

Q10) Derive the expression for the transfer function from the state model X=AX+BU Y=CX+DU

To derive the transfer function expression from the state model, let's start with the given state-space representation:

1. State equation: X = AX + BU
2. Output equation: Y = CX + DU

Here, X represents the state vector, U represents the input vector, and Y represents the output vector. A, B, C, and D are matrices that define the system's dynamics and relationships between the state, input, and output.

To derive the transfer function, we need to take the Laplace transform of both the state and output equations.

1. Taking the Laplace transform of the state equation: sX(s) - X(0) = AX(s) + BU(s)

Rearranging the equation, we have: (sI - A)X(s) = BU(s) + X(0)

Solving for X(s), we get: X(s) = (sI - A)^(-1)BU(s) + (sI - A)^(-1)X(0)

2. Taking the Laplace transform of the output equation: Y(s) = CX(s) + DU(s)

Substituting the expression for X(s) obtained above, we get: Y(s) = C(sI - A)^(-1)BU(s) + C(sI - A)^(-1)X(0) + DU(s)

The transfer function, H(s), relates the Laplace transforms of the output, Y(s), to the input, U(s). It can be obtained by rearranging the equation and factoring out U(s):

Y(s) = (C(sI - A)^(-1)B + D)U(s) + C(sI - A)^(-1)X(0)

Comparing this equation with the transfer function expression: Y(s) = H(s)U(s)

We can conclude that the transfer function, H(s), is given by: H(s) = C(sI - A)^(-1)B + D

Therefore, the derived expression for the transfer function from the given state model is H(s) = C(sI - A)^(-1)B + D. This transfer function relates the Laplace transform of the output to the Laplace transform of the input, taking into account the system's matrices A, B, C, and D.

Q11) Find the state space model for the system having transfer function Y(S)/U(S)=1/(S^2+S+1)

To find the state space model for the given transfer function, which is $Y(s)/U(s) = 1/(s^2 + s + 1)$, we can follow these steps:

Step 1: Define the state variables. Let's assume that the state variables are $x1$ and $x2$.

Step 2: Express the state equations. The state equations describe the dynamics of the system. We can write them in the form of $dx/dt = Ax + Bu$.

Assuming $x = [x1\ x2]^T$, the state equations become: $dx1/dt = x2$ $dx2/dt = -x1 - x2 + u$

Step 3: Express the output equation. The output equation relates the state variables to the output. In this case, the output equation is $Y(s) = Cx + Du$.

Here, Y(s) represents the Laplace transform of the output, and U(s) represents the Laplace transform of the input.

The transfer function $Y(s)/U(s) = 1/(s^2 + s + 1)$ can be rewritten as: $Y(s) = (1/(s^2 + s + 1))U(s)$

Comparing this with the output equation, we have: $C = [1\ 0]$ and $D = 0$

Step 4: Construct the state space model. Now, we can assemble the state space model using the state equations and the output equation obtained above.

The state equations: $dx1/dt = x2$ $dx2/dt = -x1 - x2 + u$

The output equation: $Y(s) = [1\ 0]\ [x1\ x2]^T$

Therefore, the state space model for the given transfer function is: $dx/dt = [0\ 1;\ -1\ -1]\ x + [0;\ 1]\ u$ $Y(s) = [1\ 0]\ [x1\ x2]^T$

In matrix form: $\dot{x} = Ax + Bu$ $y = Cx + Du$

Where: $A = [0\ 1;\ -1\ -1]$ $B = [0;\ 1]$ $C = [1\ 0]$ $D = 0$

This state space model represents the dynamic behavior of the system with the given transfer function. The state variables x1 and x2 describe the internal state of the system, while the input u and output y are related through the equations mentioned above.

Q12) Verify the controllability and observability given system represented by X=[x1 x2]=[-1 -1 1 0] [x1 x2]+[1 0] [u] Y=[0 1] [x1 x2]?

To verify the controllability and observability of the given system, we need to check the properties of the system's controllability and observability matrices.

The controllability matrix is given by: C = [B AB]

The observability matrix is given by: O = [C; CA]

Controllability: To check the controllability of the system, we need to determine if the controllability matrix C is full rank (i.e., its rank is equal to the number of states).

Given: A = [-1 -1; 1 0] B = [1; 0]

The controllability matrix C becomes: C = [B AB] = [1 -1; 0 1]

Calculating the rank of C, we find: rank(C) = 2

Since the rank of C is equal to the number of states (2), the system is controllable.

Observability: To check the observability of the system, we need to determine if the observability matrix O is full rank (i.e., its rank is equal to the number of states).

Given: C = [0 1]

The observability matrix O becomes: O = [C; CA] = [0 1; -1 -1]

Calculating the rank of O, we find: rank(O) = 2

Since the rank of O is equal to the number of states (2), the system is observable.

Therefore, based on the calculations, the given system is both controllable and observable.