SYSTEM REPRESENTATION

In this topic, we will learn:

- Important definitions of block diagrams and signal flow graphs: signals, system, summing junctions, pickoff points, series, cascade and feedback forms.
- Techniques of simplifying block diagrams.
- Signal flow graphs.
- Changing block diagrams to signal flow graphs and vice versa.
- Mason’s Rule and example questions.

INTRODUCTION

- A control system consists of the inter-connection of subsystems.
- A more complicated system will have many interconnected subsystems.
- For the purpose of analysis, we want to represent the multiple subsystems as a single transfer function.
- A system with multiple subsystems can be represented in two ways:
  - Block diagrams
  - Signal flow graphs


**BLOCK DIAGRAMS**

- The basic components in a block diagram are:

**Signals**

\[ R(s) \rightarrow C(s) \]

The direction of signal flow is shown by the arrow.

**System blocks**

\[ R(s) \rightarrow G(s) \rightarrow C(s) \]

The system block represented by a transfer function.

**Summing junctions**

\[ R_1(s) + R_2(s) - R_3(s) \rightarrow C(s) = \]

The signals are added/subtracted algebraically.
Pickoff points

\[
\begin{align*}
R(s) & \quad R(s) \\
R(s) & \quad R(s) \\
& \quad R(s)
\end{align*}
\]

The same signal is distributed to other subsystems

- The subsystems in a block diagram are normally connected in three forms:

Cascade form

\[
\begin{align*}
X_2(s) &= G_1(s)R(s) \\
X_1(s) &= G_2(s)G_1(s)R(s) \\
C(s) &= G_3(s)G_2(s)G_1(s)R(s)
\end{align*}
\]

The block diagram can be reduced into a single block by multiplying every block to give:
Parallel form

\[ X_1(s) = R(s)G_1(s) \]

\[ X_2(s) = R(s)G_2(s) \]

\[ X_3(s) = R(s)G_3(s) \]

\[ C(s) = \pm [\pm G_1(s) \pm G_2(s) \pm G_3(s)]R(s) \]

The block diagram can be reduced into a single block by summing every block to give:

Feedback form

\[ R(s) + E(s) \]

\[ G(s) \]

\[ H(s) \]

Plant and controller

Actuating signal (error)

Feedback

Output
- closed-loop transfer function:

- open-loop transfer function:

**Moving blocks to create familiar forms**

- It is not always apparent to get block diagrams in the familiar forms.

- We have to move blocks to get the familiar forms in order to be able to reduce the block diagram into single transfer function.

- Moving the summing junction to the front of a block
• Moving the summing junction to the back of a block
- Moving pick-off point to the front of a block

\[
\begin{align*}
\text{G}(s) & \quad \text{R}(s) \quad \text{G}(s) \\
\text{R}(s) & \quad \text{R}(s) \quad \text{R}(s) \\
\text{R}(s) & \quad \text{R}(s)
\end{align*}
\]

- Moving pick-off point to the back of a block

\[
\begin{align*}
\text{G}(s) & \quad \text{R}(s) \quad \text{G}(s) \\
\text{R}(s) & \quad \text{R}(s) \quad \text{R}(s) \\
\text{R}(s) & \quad \text{R}(s)
\end{align*}
\]
Example (5.1): Reduce the following block diagram into a single transfer function
Example (5.2): Reduce the following block diagram into a single transfer function.
SIGNAL FLOW GRAPHS

- An alternative to block diagrams.
- It consists only of branches to represent systems and nodes to represent signals.

Branches

- Represented by a line with an arrow showing the direction of signal flow through the system.

\[ G(s) \]

- The transfer function is written close to the line and arrow.

Nodes

- Represented by a small circle with the signal’s name written adjacent to the node.

\[ V(s) \]
Example:

Converting block diagrams into signal-flow graphs

- The block diagrams in cascade, parallel and feedback forms can be converted into signal-flow diagrams.
- We can start with drawing the signal nodes, and then interconnect the signal nodes with system branches.

Cascade form
Parallel form

\[ X_1(s) = R(s)G_1(s) \]
\[ X_2(s) = R(s)G_2(s) \pm C(s) = [\pm G_1(s) \pm G_2(s) \pm G_3(s)]R(s) \]
\[ X_3(s) = R(s)G_3(s) \]

Feedback form

1. Input: \( R(s) \)
2. Actuating signal (error): \( E(s) \) from \( R(s) \) and \( G(s) \)
3. Feedback: \( H(s) \) to \( C(s) \)
4. Output: \( C(s) \)
5. Plant and controller: \( G(s) \)

\( C(s) = G(s) + H(s) \)
Example (5.6): Convert the following block diagram into signal-flow graph
Mason’s Rule

- Mason’s rule is used to get the single transfer function in the signal-flow graph.
- We need to know some important definitions:

  - **Loop** – a closed path which starts and ends at the same node.
  - **Loop gain** – the product of branch gains found by traversing a loop.

  - **Forward-path** – a path from the input node to the output node of the signal-flow graph in the direction of signal flow.
- Forward-path gain – the product of gains found by traversing a path from the input node to the output node of the signal-flow graph in the direction of signal flow.

- Non-touching loops – loops that do not have any nodes in common.

- Non-touching loops gain – the product of loop gains from non-touching loops taken two, three, four, or more at a time.
Mason’s rule:

\[ G(s) = \frac{C(s)}{R(s)} = \frac{\sum_{k} T_k \Delta_k}{\Delta} \]

\( k \) = number of forward path

\( T_k \) = the \( k \)-th forward-path gain

\( \Delta = 1 - \sum \) loop gains + \( \sum \) non-touching loops gain (taken 2 at a time) - \( \sum \) non-touching loops gain (taken 3 at a time) + \( \sum \) non-touching loops gain (taken 4 at a time) - ... 

\( \Delta_k = \Delta - \sum \) loop gain terms in \( \Delta \) that touch the \( k \)-th forward path. In other words, \( \Delta_k \) is formed by eliminating from \( \Delta \) those loop gains that touch the \( k \)-th forward path.
Example (5.7): Find the transfer function, $C(s)/R(s)$ of the following signal-flow graph,