EEE 4207 Control Eng. II Lecture 2&3, Agnes

**RECAP: State-space representation**

As we know, a control system can be described through G(s), the system’s transfer function is derived via the Laplace transformation of the input (u(t)) and output (y(t)) differential equations of the system to model. Nevertheless, there is another way to define a control system. This is the so-called state-space representation of the system. Transfer function G(s) can be transformed to state-space representation, and vice-versa.

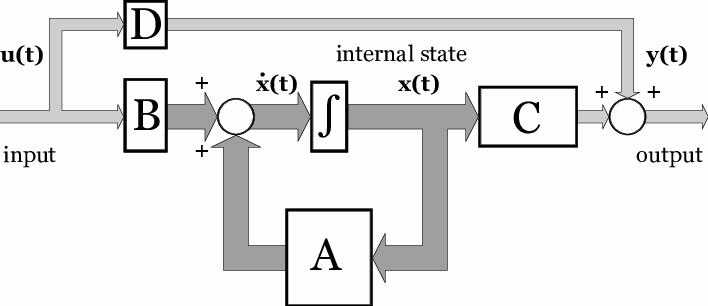
The standard state-space representation is given below:

(state equation);

(output equation);

Where:

* x is the state vector that contains all the state variables, (it can be position, velocity, acceleration, angle, voltage, current, etc.);
* is the time derivative of the state vector;
* u is the input scalar;
* y is the output scalar (it can be position, velocity, acceleration, angle, voltage, current, etc.);
* A is the state matrix;
* B is the input vector;
* C is the output vector;
* D is the feedforward constant.





*The general block diagram of the state-space representation*

Definition of state x(t): state of a system at time t0 is the information by which, together with the control input u(t) (t ≥t0.), the response of the system can be determined for all t ≥t0.

Example 1

A system given as:

Example 2

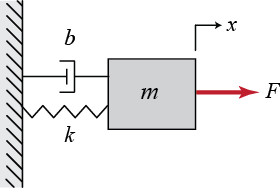


Example 3: Mechanical system:

Newton's laws of motion form the basis for analyzing mechanical systems. Newton’s second law, stating that the sum of the forces acting on a body equals the product of its mass and acceleration. Newton's third law, for our purposes, states that if two bodies are in contact, then they experience the same magnitude contact force, just acting in opposite directions.

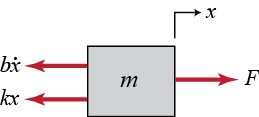
$$
\Sigma \mathbf{F} = m \mathbf{a} = m \frac{d^2 \mathbf{x}}{dt^2}
$$

When applying this equation, it is best to construct a free-body diagram (FBD) of the sysetm showing all of the applied forces.



*Mass-Spring-Damper system*

The free-body diagram for this system is shown below. The spring force is proportional to the displacement of the mass, $x$, and the viscous damping force is proportional to the velocity of the mass, $v=\dot{x}$. Both forces oppose the motion of the mass and are, therefore, shown in the negative $x$-direction. Note also that $x=0$ corresponds to the position of the mass when the spring is unstretched.



Now we proceed by summing the forces and applying Newton’s second law, in each direction. In this case, there are no forces acting in the $y$-direction; however, in the $x$-direction we have:

$$
\Sigma F_x = F(t) - b \dot{x} - k x = m \ddot{x}
$$

This equation, known as the governing equation, completely characterizes the dynamic state of the system. To determine the state-space representation of the mass-spring-damper system, we must reduce the second-order governing equation to a set of two first-order differential equations. To this end, we choose the position and velocity as our state variables.



$$
\mathbf{x} = \left[ \begin{array}{c} x \\ \dot{x} \end{array}\right]
$$



The position variable captures the potential energy stored in the spring, while the velocity variable captures the kinetic energy stored by the mass. The damper only dissipates energy, it doesn't store energy. Often when choosing state variables it is helpful to consider what variables capture the energy stored in the system.



The state equation in this case is:

$$
\mathbf{\dot{x}} = \left[ \begin{array}{c} \dot{x} \\ \ddot{x} \end{array} \right] = \left[ \begin{array}{cc} 0 & 1 \\ -\frac{k}{m}  & -\frac{b}{m} \end{array} \right] \left[ \begin{array}{c} x \\ \dot{x} \end{array} \right] + \left[ \begin{array}{c} 0 \\ \frac{1}{m} \end{array} \right] F(t)
$$

If, for instance, we are interested in controlling the position of the mass, then the output equation is:

$$
y = \left[ \begin{array}{cc} 1 & 0 \end{array} \right] \left[ \begin{array}{c} x \\ \dot{x} \end{array} \right]
$$

Assignment: Direct current machines are the most versatile energy conversion devices. Their outstanding advantage is that the volt-ampere or speed torque characteristic of these machines are very much flexible and easily adaptable for both steady state and dynamic operations. When a wide range of speed control and torque output are required dc motor is an obvious choice.

The state space approach is a generalized time domain method for modelling, analysing and designing a wide range of control systems and is particularly well suited to digital computational technique. In this question we attempt Dc motor modelling using state space analysis. Different equations related to DC motor are given below

Where 𝑒𝑎(t) = armature voltage, 𝑒𝑚 (t) = back emf, 𝑖𝑎(t) = armature current, 𝑇(𝑡) =developed torque, Ɵ(𝑡) = motor shaft angle, , Ɵ= shaft speed, J= moment of inertia of the rotor, B = viscous frictional constant, 𝐿𝑚 = inductance of armature windings, 𝑅𝑚= armature winding resistance, 𝐾𝑡= motor torque constant, 𝐾𝑚= motor constant.

Here the motor speed ω(t) is controlled by varying the armature voltage 𝑒𝑎(t). Hence 𝑒𝑎(t) is the input variable and ω(t) is the output variable.

We chose as the state variables 𝑥1(𝑡) = , and 𝑥2(𝑡) = 𝑖𝑎 (𝑡)

Find the state equations

***Equivalence between state-space representation and system’s transfer function***

State-space description of a system and the system’s transfer function can be transformed to each other. In this lecture we will learn how to realize the conversion from state-space model to transfer function.

|  |  |
| --- | --- |
| We have the general SISO, LTI state-space form as follows: |  |
| (t) |  |
| y(t) = Cx(t) + Du(t); |  |
| After applying the Laplace transformation, the state-space form becomes: |  |
| sX(s) = AX(s) + BU(s); |  |
| Y (s) = CX(s) + DU(s); |  |

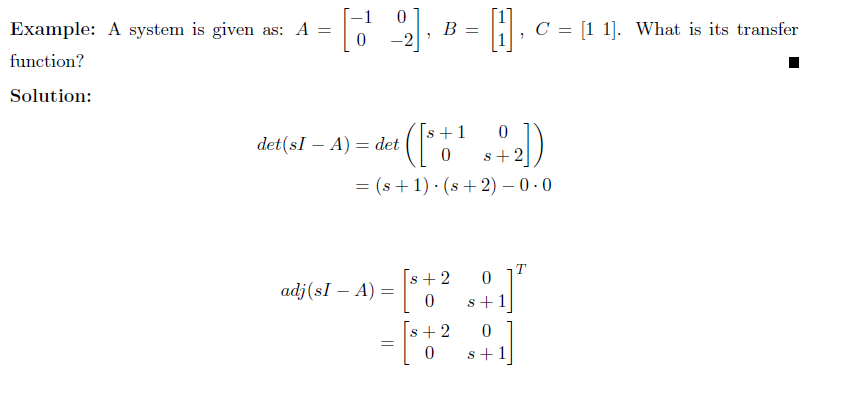
State equation can be reformulated:

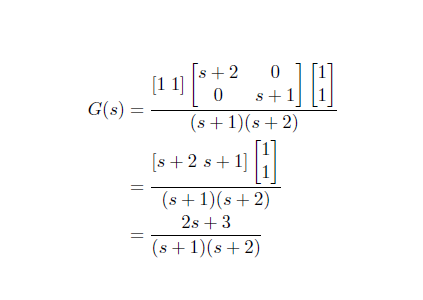
|  |  |  |
| --- | --- | --- |
| sX(s)- | AX(s) = BU(s) |  |
|  | (sI-A) X(s) = BU(s) |  |
|  | X(s) = (sI-A)-1 BU(s) |  |

When we substitute X(s) the output equation becomes:

Y(s) = C(sI-A)-1 BU(s) + DU(s) = [C(sI-A)-1 B+D] U(s)

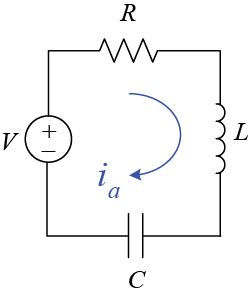
So, the transfer function is





Electrical systems: Like Newton’s laws for mechanical systems, Kirchoff’s circuit laws are fundamental analytical tools for modeling electrical systems. Kirchoff’s current law (KCL) states that the sum of the electrical currents entering a node in a circuit must equal the sum of electrical currents exiting the node. Kirchoff’s voltage law (KVL) states that the sum of voltage differences around any closed loop in a circuit is zero. When applying KVL, the source voltages are typically taken as positive and the load voltages are taken as negative.

Consider a simple series combination of three passive electrical elements: a resistor, an inductor, and a capacitor, known as an RLC Circuit and derive the state space equation and transfer function.



**State diagram**

The significance of the state diagram is that it forms a close relationship among the state equations, computer simulation, and transfer functions. A state diagram is constructed following all the rules of the SFG using the Laplace-transformed state equations.The basic elements of a state diagram are similar to the conventional SFG, except for the **integration** operation. Let the variables *x1(t )* and *x2(t )* be related by the first-order differentiation:

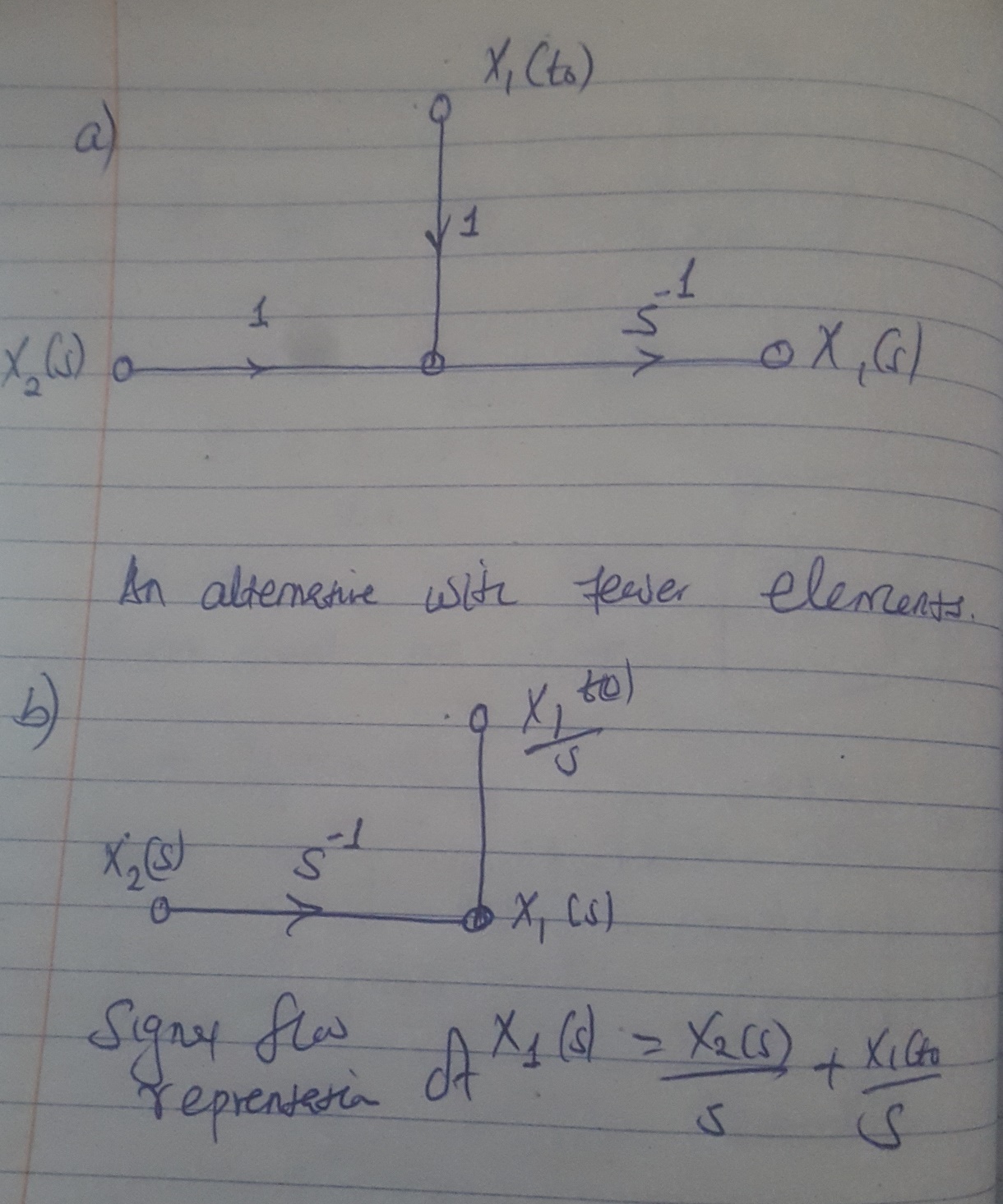
Integrating both sides of the last equation with respect to t from the initial , we get

Because the SFG algebra does not handle integration in the time domain, we must take the Laplace transform on both sides

Because the past history of the integrator is represented by *x*1( *t0 ),* and the state transition is assumed to start at τ = *t0,* x2(τ)=0 for 0<τ< t0

Thus equation becomes:

It is now algebraic and can be represented by an SFG as shown below, the *the output of the integrator is equal to s- 1 times the input, plus the initial condition x*1(t0) / *s*

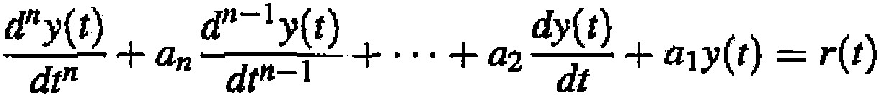


**Importance uses of the state diagram**

* A state diagram can be constructed directly from the system's differential equation. This allows the determination of the state variables and the state equations.
* A state diagram can be constructed from the system's transfer function. This step is defined as the decomposition of transfer functions (we shall see later)
* The state diagram can be used to program the system on an analog computer or for simulation on a digital computer.
* The state-transition equation in the Laplace transform domain may be obtained from the state diagram by using the SFG gain formula.
* The transfer functions of a system can be determined from the state diagram.
* The state equations and the output equations can be determined from the state diagram.

1. **From Differential Equations to State Diagrams**

When a linear system is described by a high order differential equation, a state diagram can be constructed from these equations, although a direct approach is not always the most convenient. Consider the following differential equation:



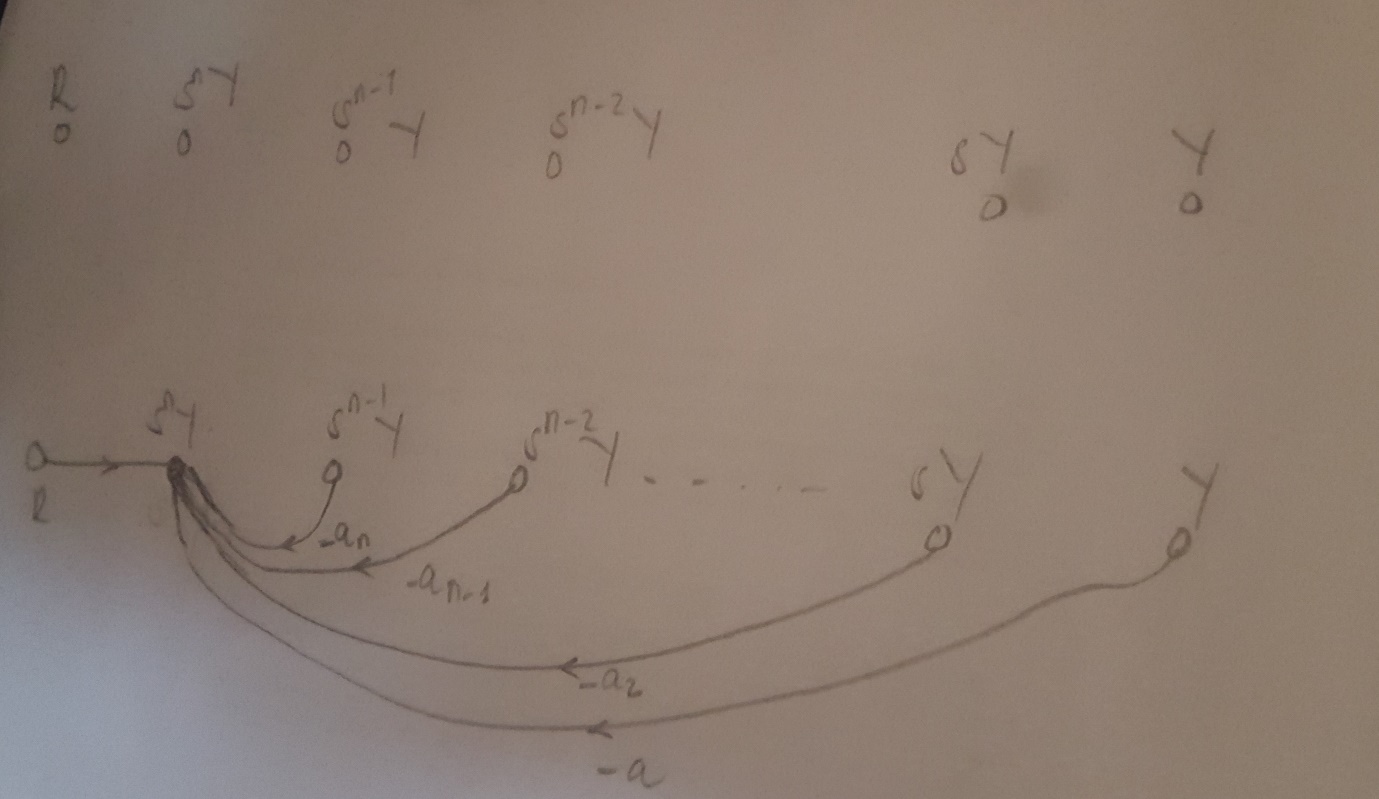
As a first step, the nodes representing *R(s), sny,(s), sn-1 y(s)………. ,sY(s),* and *Y(s)* are arranged from left to right,

*-*

0 0 0 0 0 0

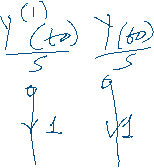
*R Y sy y*

Because *sY(s)* corresponds to  *where i*= 0, 1, 2, ... , *n,* in the Laplace domain, as the the next steps follow as the nodes are connected by branches





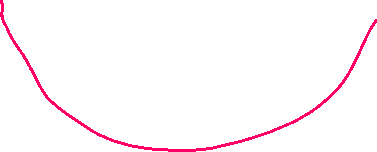
Finally, the integrator branches with gains of s- 1 are inserted, and the initial conditions are added to the outputs of the integrators and this completes state diagram. T*he outputs of the integrators are defined as the state variables, x 1, x 2 ,* ••• , *Xn,* This is usually the natural choice of state variables once the state diagram is drawn.

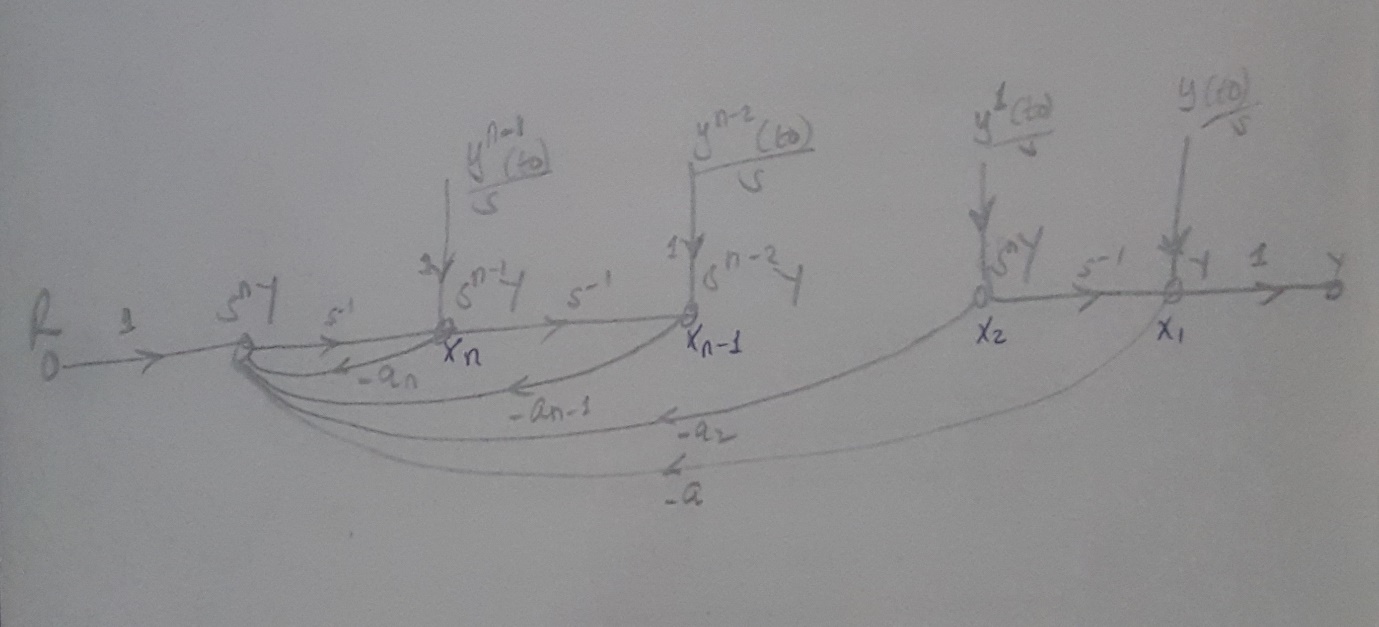






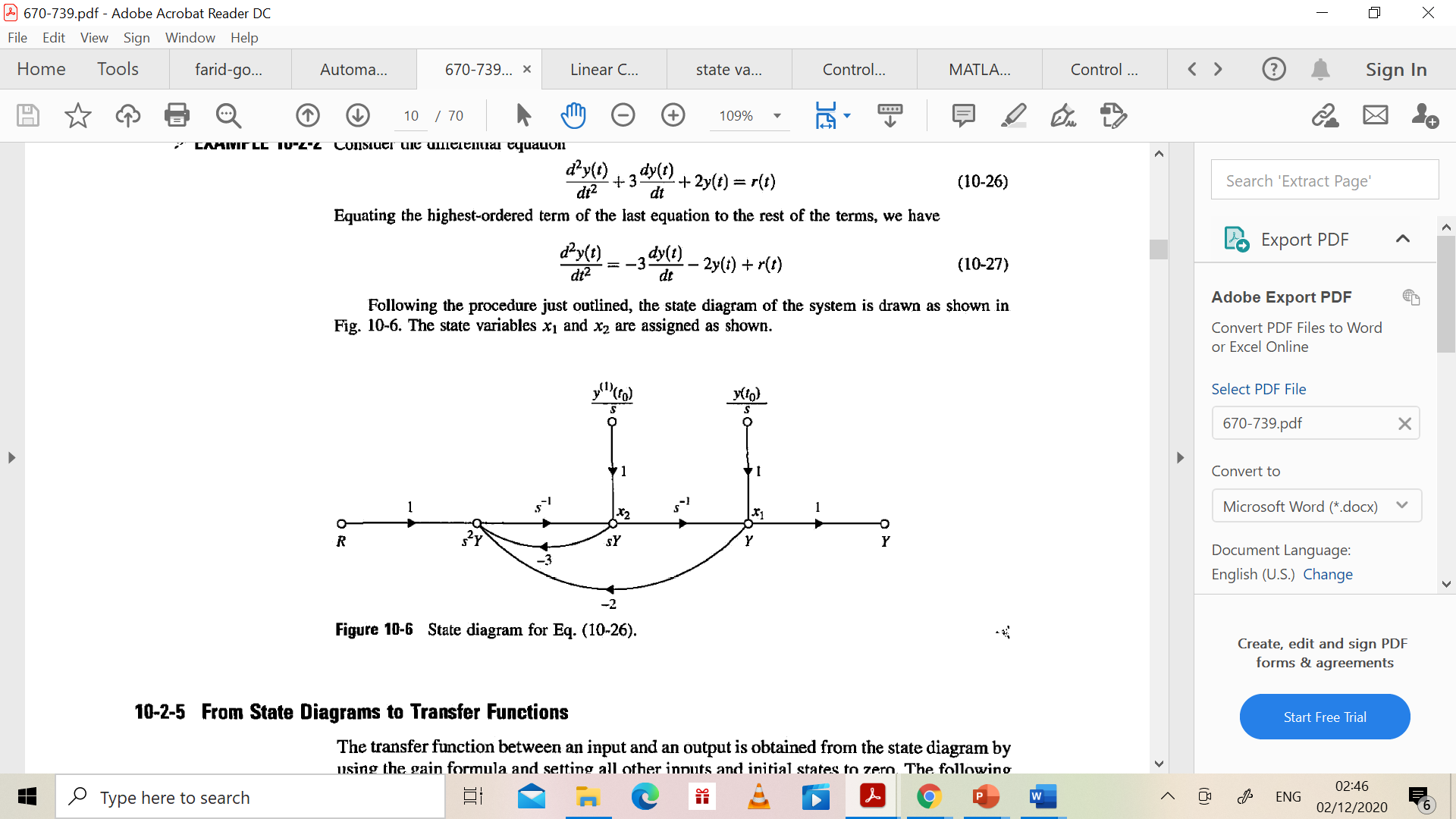






1. **From state diagrams to transfer functions**

The transfer function between an input and an output is obtained from the state diagram by using the gain formula and setting all other inputs and initial states to zero. Consider the state diagram below,



The transfer function between R(s) and Y(s) is obtained by applying the gain formula between these two nodes and setting the initial states to zero. We have:

1. **From state diagrams to state and output equations**

The state equations and the output equations can be obtained directly from the state diagram by using the SFG gain formula.

**State equation:**

**Output equation**

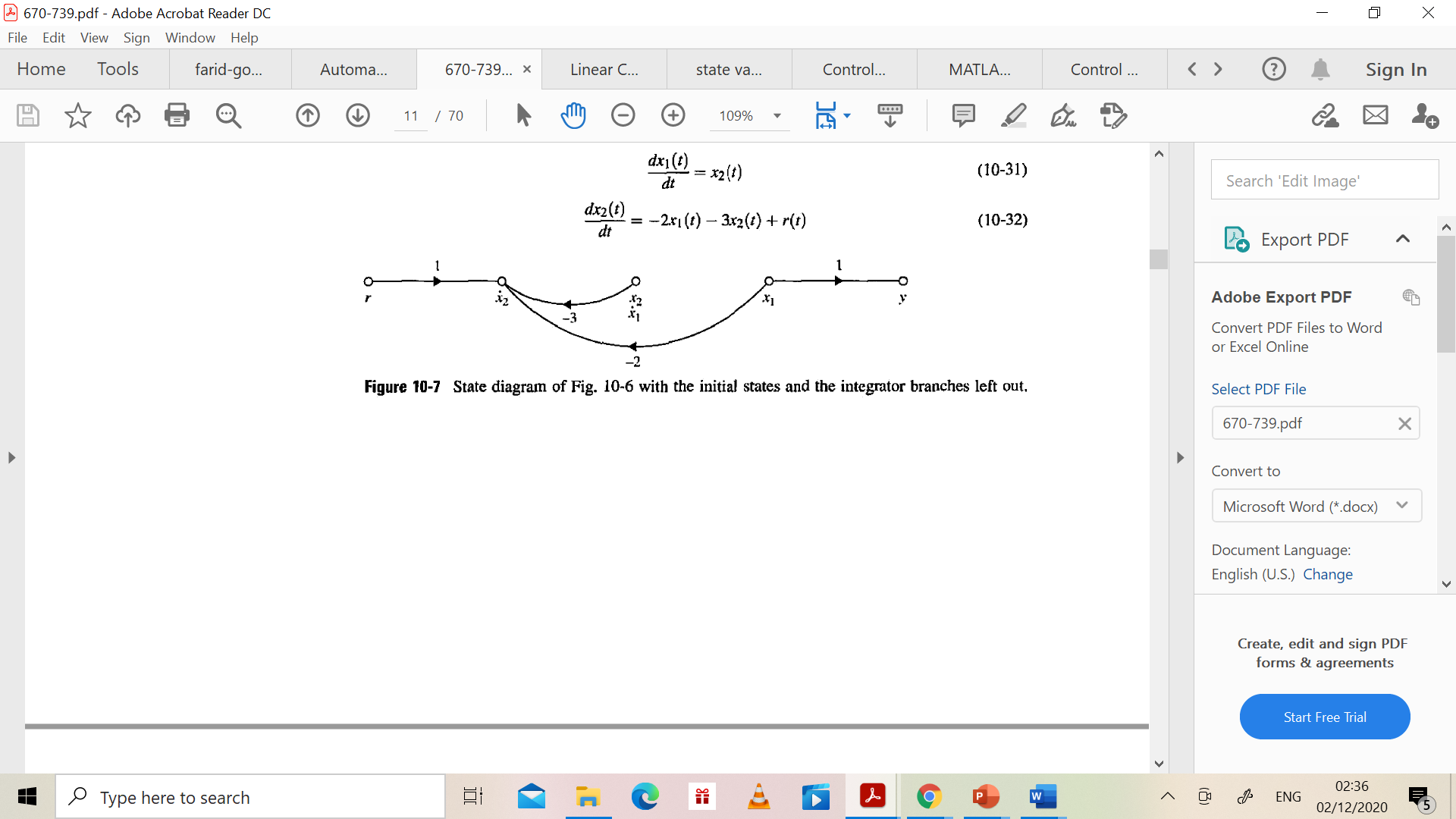
where x(t) is the state variable; r(t) is the input; *y(t)* is the output; and *a, b, c,* and d are constant coefficients. Based on the general form of the state and output equations, the following procedure of deriving the state and output equations from the state diagram are outlined:

*Delete the initial states and the integrator branches with gains s- 1 from the state diagram, since the state and output equations do not contain the Laplace operator s or the initial states.*

*For the state equations, regard the nodes that represent the derivatives of the state variables as output nodes, since these variables appear on the left-hand side of the state equations. The output y(t) in the output equation is naturally an output node variable.*

*Regard the state variables and the inputs as input variables on the state diagram, since these variables are found on the right-hand side of the state and output equations.*

*Apply the SFG gain formula to the state diagram.*

Example: 

state diagram with the integrator branches and the initial states eliminated. Using *dx1(t)/dt* and *dx2(t )/ dt* as the output nodes and X 1(t ), x2(t), and *r(t)* as input nodes, and applying the gain formula between these nodes, the state equations are obtained as

Applying the gain formula with *x 1(t, x 2(t ),* and *r(t)* as input nodes and *y(t)* as the output node, the output equation is written

*y(t)* = x1(t)

**State transition Matrix**

Consider the state equation n of linear time invariant system as,

The matrices A and B are constant matrices. This state equation can be of two types,

1. Homogeneous and

2. Nonhomogeneous

**Homogeneous Equation**

If A is a constant matrix and input control forces are zero then the equation takes the form,

Such an equation is called homogeneous equation. The obvious equation is if input is zero, In such systems, the driving force is provided by the initial conditions of the system to produce the output. For example, consider a series RC circuit in which capacitor is initially charged to V volts. The current is the output. Now there is no input control force i.e. external voltage applied to the system. But the initial voltage on the capacitor drives the current through the system and capacitor starts discharging through the resistance R. Such a system which works on the initial conditions without any input applied to it is called homogeneous system.

**Nonhomogeneous Equation**

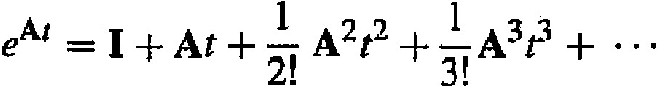
If A is a constant matrix and matrix U(t) is non-zero vector i.e. the input control forces are applied to the system then the equation takes normal form as,

Such an equation is called nonhomogeneous equation. Most of the practical systems require inputs to dive them. Such systems arc nonhomogeneous linear systems.

The **state-transition matrix** is defined as a matrix that satisfies the linear homogeneous state equation:

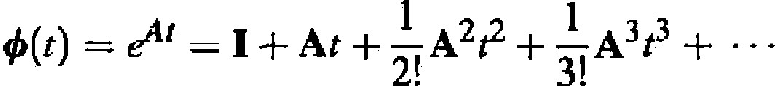
Can be written as: 

For t ≥0 where represents the following power series of the matrix **A**t, and



Since

Thus the state transition matrix:



**Significance of the State-Transition Matrix**

Because the state-transition matrix satisfies the homogeneous state equation, it represents the **free response** of the system. In other words, it governs the response that is excited by the initial conditions only. The state-transition matrix is dependent only upon the matrix **A** and, therefore, is sometimes referred to as the **state­transition matrix of A.** As the name implies, the state-transition matrix Φ**(t)** completely defines the transition of the states from the initial time *t* = 0 to any time *t* when the inputs are zero.

**The matrix exponential representation of the state transition matrix allows some of its properties to be simply stated:**

* Φ(0) = I, which simply states that the state response at time t = 0 is identical to the initial conditions.
* Φ-1(t) = Φ(-t). The response of an unforced system before time t = 0 may be calculated from the initial conditions x(0), x(−t) = Φ(−t)x(0) = Φ-1 (t)x(0) and the inverse always exists.

Proof: post multiplying both sides by e-At , we get



Then, pre multiplying both sides by Φ-1(t), we get



An interesting result from this property of Φ**(t)** is that eqn can be rearranged to read

**X(0)** = Φ ***(-t) x(t)***

which means that the state-transition process can be considered as bilateral in time. That is, the transition in time can take place in either direction.

* Φ(t2-t1)Φ(t2-t0) = Φ(t2 - t0 ) for any t0, t1,t2

Proof

Φ(t2-t1)Φ(t2-t0)= = Φ(t2 - t0 )

This property of the state-transition matrix is important because it implies that a state-transition process can be divided into a number of sequential transitions (the state­ transition process can be divided into any number of parts)

* **for k= positive interger**

**Proof:**